

Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”  
Завідувач кафедри кібербезпеки  
та програмного забезпечення  
д.т.н., професор  
\_\_\_\_\_ Олексій СМІРНОВ  
“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи IRC-клієнту”**

Виконав здобувач вищої освіти  
II курсу, групи КН-22М-1  
ОПП «Комп’ютерні науки»  
спеціальності 122 «Комп’ютерні науки»  
\_\_\_\_\_ Строміленко А.Д.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Керівник проекту  
кандидат технічних наук  
\_\_\_\_\_ Улічев О.С.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Рівень вищої освіти магістр  
Галузь знань 12 "Інформаційні технології"  
Спеціальність 122 "Комп'ютерні науки"  
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

## ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Строміленку Артему Дмитровичу

(прізвище, ім'я, по батькові)

- |  |  |  |                            |   |   |  |  |  |                     |  |  |
|--|--|--|----------------------------|---|---|--|--|--|---------------------|--|--|
| 1. Тема роботи   | <u>Дослідження та програмна реалізація системи IRC-клієнту</u>   |  |                            |   |   |  |  |  |                     |  |  |
| 2. Керівник роботи   | <u>Улічев Олександр Сергійович, канд. техн. наук</u><br>(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)  |  |                            |   |   |  |  |  |                     |  |  |
| затверджені наказом вищого навчального закладу № 32-13 від 04.08.2023 року           |  |  |                            |   |   |  |  |  |                     |  |  |
| 3. Строк подання студентом роботи до захисту   | <u>10.12.2023 р.</u>   |  |                            |   |   |  |  |  |                     |  |  |
| 4. Мета та завдання випускної кваліфікаційної роботи:                                | <u>Метою розробки є дослідження та програмна реалізація системи IRC-клієнту</u>  |  |                            |   |   |  |  |  |                     |  |  |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | <table border="1"><tr><td><u>1. Призначення та область використання.</u></td><td><u>6. Наукова новизна.</u></td></tr><tr><td><u>2. Перегляд аналогічних існуючих систем.</u></td><td><u>7. Економічна ефективність розробленої програми.</u></td></tr><tr><td><u>3. Опис і обґрунтування проектних рішень.</u></td><td><u>8. Заходи з охорони праці та техніки безпеки.</u></td></tr><tr><td><u>4. Етапи програмування системи.</u></td><td><u>9. Висновки.</u></td></tr><tr><td><u>5. Впровадження системи в промислову експлуатацію</u></td><td></td></tr></table> | <u>1. Призначення та область використання.</u> | <u>6. Наукова новизна.</u> | <u>2. Перегляд аналогічних існуючих систем.</u> | <u>7. Економічна ефективність розробленої програми.</u> | <u>3. Опис і обґрунтування проектних рішень.</u> | <u>8. Заходи з охорони праці та техніки безпеки.</u> | <u>4. Етапи програмування системи.</u> | <u>9. Висновки.</u> | <u>5. Впровадження системи в промислову експлуатацію</u> |  |
| <u>1. Призначення та область використання.</u>                                       | <u>6. Наукова новизна.</u>   |  |                            |   |   |  |  |  |                     |  |  |
| <u>2. Перегляд аналогічних існуючих систем.</u>                                      | <u>7. Економічна ефективність розробленої програми.</u>  |  |                            |   |   |  |  |  |                     |  |  |
| <u>3. Опис і обґрунтування проектних рішень.</u>                                     | <u>8. Заходи з охорони праці та техніки безпеки.</u>   |  |                            |   |   |  |  |  |                     |  |  |
| <u>4. Етапи програмування системи.</u>   | <u>9. Висновки.</u>  |  |                            |   |   |  |  |  |                     |  |  |
| <u>5. Впровадження системи в промислову експлуатацію</u>                             |  |  |                            |   |   |  |  |  |                     |  |  |
| 6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)         |  |  |                            |   |   |  |  |  |                     |  |  |
| <u>Наукова новизна</u>   | <u>1 аркуш</u>   |  |                            |   |   |  |  |  |                     |  |  |
| <u>Структурна схема системи</u>  | <u>1 аркуш</u>   |  |                            |   |   |  |  |  |                     |  |  |
| <u>Функціональна схема системи</u>   | <u>1 аркуш</u>   |  |                            |   |   |  |  |  |                     |  |  |
| <u>Діаграма процесів</u>   | <u>1 аркуш</u>   |  |                            |   |   |  |  |  |                     |  |  |
| <u>Блок-схема алгоритму роботи додатку</u>   | <u>2 аркуша</u>  |  |                            |   |   |  |  |  |                     |  |  |
| <u>Показники економічної ефективності</u>  | <u>1 аркуш</u>   |  |                            |   |   |  |  |  |                     |  |  |

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання  
« 6 » вересня 2023 р.

Підпис керівника

\_\_\_\_\_  
(прізвище та ініціали)Завдання прийнято до виконання  
« 6 » вересня 2023 р.

Підпис здобувача

\_\_\_\_\_  
(прізвище та ініціали)

## АНОТАЦІЯ

**Строміленко А.Д. Дослідження та програмна реалізація системи IRC-клієнту. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи IRC-клієнту.

Метою розробки є дослідження та програмна реалізація системи IRC-клієнту.

Об'єктом дослідження є процес IRC-клієнту.

Предметом дослідження є методи IRC-клієнту.

Методи дослідження базуються на методах побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи IRC-клієнту.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.4.1.

**Ключові слова:** комп'ютерні науки, IRC-клієнт

## ABSTRACT

**Stromilenko A.D. Research and software implementation of the IRC client system. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.**

In this final qualification work for the second (master's) level of higher education, software designed for the IRC client system was developed.

The goal of the development is research and software implementation of the IRC client system.

The object of research is the IRC client process.

The subject of research is the methods of the IRC client.

Research methods are based on methods of building computer networks, methods of mathematical statistics, and methods of software development.

The result of the work is the software implementation of the IRC client system.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Delphi 10.4.1 environment.

**Keywords:** computer science, IRC-client

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	17
2.3 Розгорнута постановка завдання .....	23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	24
3.1 Опис функціонування системи .....	24
3.2 Розробка структурної схеми.....	28
3.3 Розробка функціональної схеми .....	35
3.4 Розробка діаграми процесів.....	40
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	43
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	43
4.2 Захист розробленого програмного забезпечення.....	63
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	64
6 НАУКОВА НОВИЗНА .....	68

						ВКРМ-122.23.0020.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Строміленко А.Д.				Дослідження та програмна реалізація системи IRC-клієнту	Літ.	Аркуш	Аркушів
Перев.	Улічев О.С.					М	1	107
Н.контр.	Коваленко А.С.				ЦНТУ КН-22М-1			
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	69
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	69
7.2 Розрахунок трудомісткості розробки програмної продукції.....	71
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	73
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	78
7.5 Визначення собівартості розробки та ціни програмної продукції.....	82
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	85
7.7 Визначення експлуатаційних витрат.....	86
7.8 Визначення економічної ефективності програмної продукції.....	87
7.9 Висновок.....	89
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	90
8.1 Вступ.....	90
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	90
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	91
8.4 Розробка заходів з умов поліпшення охорони праці.....	94
8.5 Розрахункова частина .....	95
9 ОСНОВНІ ВИСНОВКИ.....	99
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	101

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- DCC – Direct Client-to-Client  
HTTP – HyperText Transfer Protocol  
IPv6 – Internet Protocol version 6  
SSL – Secure Sockets Layer

КБПЗ-2023

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** IRC означає Internet Relay Chat. Це протокол чату. У IRC є 2 компоненти.

– Клієнти – це програми, які дозволяють використовувати IRC. Їх можна встановити на багатьох різних операційних системах або навіть запускати у веб-браузері.

– Сервери – це те, до чого підключаються клієнти. IRCChat Chat має мережу серверів. Не хвилюйтеся, вам потрібно знати лише одну адресу для мережі. Це адреса

Такі мережі, як ми, використовують систему Round Robin для рівномірного розподілу користувачів між серверами. Наявність великої кількості серверів забезпечує резервування, тому несправність одного сервера не призведе до зупинки всієї мережі.

Користування IRC зазвичай безкоштовне, але іноді вам може знадобитися придбати клієнта або сплатити плату за підписку на клієнта, розміщеного кимось іншим.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи IRC-клієнту.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем IRC-клієнту.
- Дослідження системи IRC-клієнту.
- Програмна реалізація системи IRC-клієнту.

*Об'єктом дослідження є процес IRC-клієнту.*

*Предметом дослідження є методи IRC-клієнту.*

*Методи дослідження базуються на методах побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.*

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

**Наукова новизна отриманих результатів.** У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод IRC-клієнту.
- Розроблено вітчизняний продукт IRC-клієнту, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі IRC-клієнту.

**Достовірність наукових результатів** підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи IRC-клієнту, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

### Вибір клієнта

Щоб використовувати IRC, спочатку потрібно вибрати клієнта. Клієнти дуже різноманітні, але більшість із них матиме однакові основні функції, тому не хвилюйтеся, що виберете неправильний.

Ви можете прочитати посібник про клієнтів, щоб допомогти вам прийняти рішення, або ви можете скористатися одним із наших веб-чатів, щоб поки не встановлювати нічого.

Наші веб-чати ведуться нами. Вони безкоштовні для використання, але ми обмежуємо їх, тому вони підключаються лише до irc.IRCChat.chat. Через це вони є хорошим місцем для початку. Ви можете перейти до належного клієнта, коли все зрозумієте, за допомогою людей у мережі.

### Підключення вперше

Більшість клієнтів матимуть якийсь вступ або майстер, який допоможе вам почати роботу. Знайдіть попередньо налаштовану мережу під назвою IRCChat або IRCChat Chat.

Якщо майстра немає або процес заплутаний, вам доведеться або перевірити веб-сайт клієнта, щоб отримати певну документацію про те, як його налаштувати, або спочатку підключитися через веб-чати та знайти когось, хто допоможе. Зазвичай є хтось, #IRCChatxто може вказати вам правильний шлях.

### Розмови

Як і в месенджерах, у програмі буде поле, де ви зможете вводити текст. Тут ви спілкуєтеся.

У цьому полі чату також можна виконувати команди серверу або клієнту. Команди починаються з /.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6



## Послуги

Сервіси – це спеціалізовані боти, які допомагають безперебійній роботі мережі. Серед іншого вони керують користувачами та каналами.

Використання служб – це те, як ви реєструєте псевдонім і ідентифікуєте себе (увійдіть), коли повертаєтеся для наступної сесії IRC у майбутньому.

Це також те, як ви реєструєте канали, якщо вам колись це знадобиться.

## Теми

У більшості IRC-клієнтів опис теми каналу можна знайти у верхній частині екрана. Якщо ви не бачите все це, ви можете відобразити його за допомогою /topіскоманди.

Ви можете знайти нові канали за допомогою служби пошуку за назвою каналу або за описом теми.

У будь-якій мережі IRC є багато різних тем. У IRCChat Chat більшість наших каналів присвячені програмному забезпеченню з вільною ліцензією та спільним проектам. Наша мережева політика визначає, якою ми хочемо бачити мережу.

## Ролі каналу

Коли ви перебуваєте на каналах, ви можете помітити, що деякі люди в списку членів мають символи біля своїх псевдонімів. Більшість людей не гратимуть ролі в каналах.

Люди з символами мають режими «оператор» або «озвучений». Оператори – це люди, які керують конкретним каналом. Вони роблять такі дії, як зміна режимів каналу, куратор теми, керування доступом до каналу. Вони також відповідають за модерування чату, тож іноді вони можуть замовкнути окремих людей або перевести канал у модерований режим. Коли канал знаходиться в модерованому режимі, голосові коментарі користувачів все ще можна побачити. Часто канали надають своїм роботам-помічникам голосовий або операторський режими.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

## 1.2 Область застосування

Це дуже невеликий зразок команд, які можуть бути вам корисними, коли ви починаєте вивчати IRC. У багатьох клієнтах для них можуть бути кнопки.

Частини команд, показані в, [] є необов'язковими.

- /join #IRCChat
  - Увійдіть на канал #IRCChat.
- /part [#IRCChat] [message]
  - Вийти з каналу #IRCChat.
  - Якщо канал не вказано, ви розділите канал, у який введете це.
  - Додаткове повідомлення буде показано лише на цьому каналі під час вашого від'їзду.
- /nick nickname
  - Змінює ваш псевдонім на вказаний вами.
- /msg nickname message
  - Надсилає приватне повідомлення користувачеві «нік», *не* відкриваючи вікно нової розмови.
- /query nickname [message]
  - Відкрийте вікно бесіди з «нікнеймом» користувача та, за бажанням, додайте повідомлення.
  - Інший користувач *не* дізнається, що ви це зробили, доки ви не надішлете повідомлення.
  - Це корисно, щоб переконатися, що ваша бесіда переходить у правильне місце, наприклад, коли ви реєструєтесь у NickServ.
- /me action
  - Використовуйте це для емоцій. З'явиться щось на зразок \* Nickname action.
- /quit [message]
  - Відключає вас від мережі.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

○ Додаткове повідомлення буде показано на кожному каналі, на якому ви перебуваєте.

Реалізація деяких команд буде відрізнятись залежно від вашого клієнта. Якщо наведені нижче дії не працюють належним чином, перегляньте документацію на веб-сайті вашого клієнта або запитайте на його каналі IRC. У багатьох є канали на IRCChat Chat.

- /ignore nickname

○ Ви можете перестати бачити розмову користувача. Корисно, якщо вони є шкідниками.

- /unignore nickname

○ Може дозволити вам знову побачити користувача, якщо ви передумаєте.

- /ignores

○ Може перелічувати встановлені вами поточні ігнорування.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи IRC-клієнту, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		10

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

### 2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

У світі існують найрізноманітніші IRC клієнти. Існують додатки як для сімейства windows, так і для UNIX-подібних систем. Якщо немає доступу до комп'ютеру – не лихो, скористайтеся IRC програмами для КПК, смартфонів або комунікаторів. У випадку простого мобільного телефону можна обійтися java-апплетами J2ME, єдина умова – підтримка вашим апаратом стандарту MIDP.

IRC клієнти для Windows 10/11:

– mIRC – безумовний лідер серед IRC-клієнтів, використовується мільйонами людей в усі світі. Мірк має значного функціонала, що може бути доповнений безліччю скриптів.

– Miranda – даний інтернет-пейджер підтримує IRC і всі інші мислимі протоколи спілкування на рівні плагінів. Сотні аддонів дозволяють гнучко налаштувати подання програми і її зв'язку з операційною системою й мережею.

– xChat – мультиплатформений графічний IRC-клієнт, що володіє всіма необхідними для плідного спілкування функціями. Дозволяє одночасно підключатися відразу до декількох каналів (кімнат), можна вести публічні розмови або приватні бесіди. Також існує можливість обмінюватися файлами.

– KVIrc – графічний кроссплатформений IRC клієнт. Підтримка всіх відомих протоколів, SSL шифрування, у тому числі для DCC з'єднань. Гнучкий інтерфейс, який налаштовується, прекрасна мова скриптингу з підтримкою об'єктно-орієнтованого програмування. Перший IRC клієнт, що оголосив про підтримку аватарів. Реальна альтернатива mIRC або XChat, популярний на заході.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

В Україні розповсюджений недавно, але вже завоював популярність у певних колах.

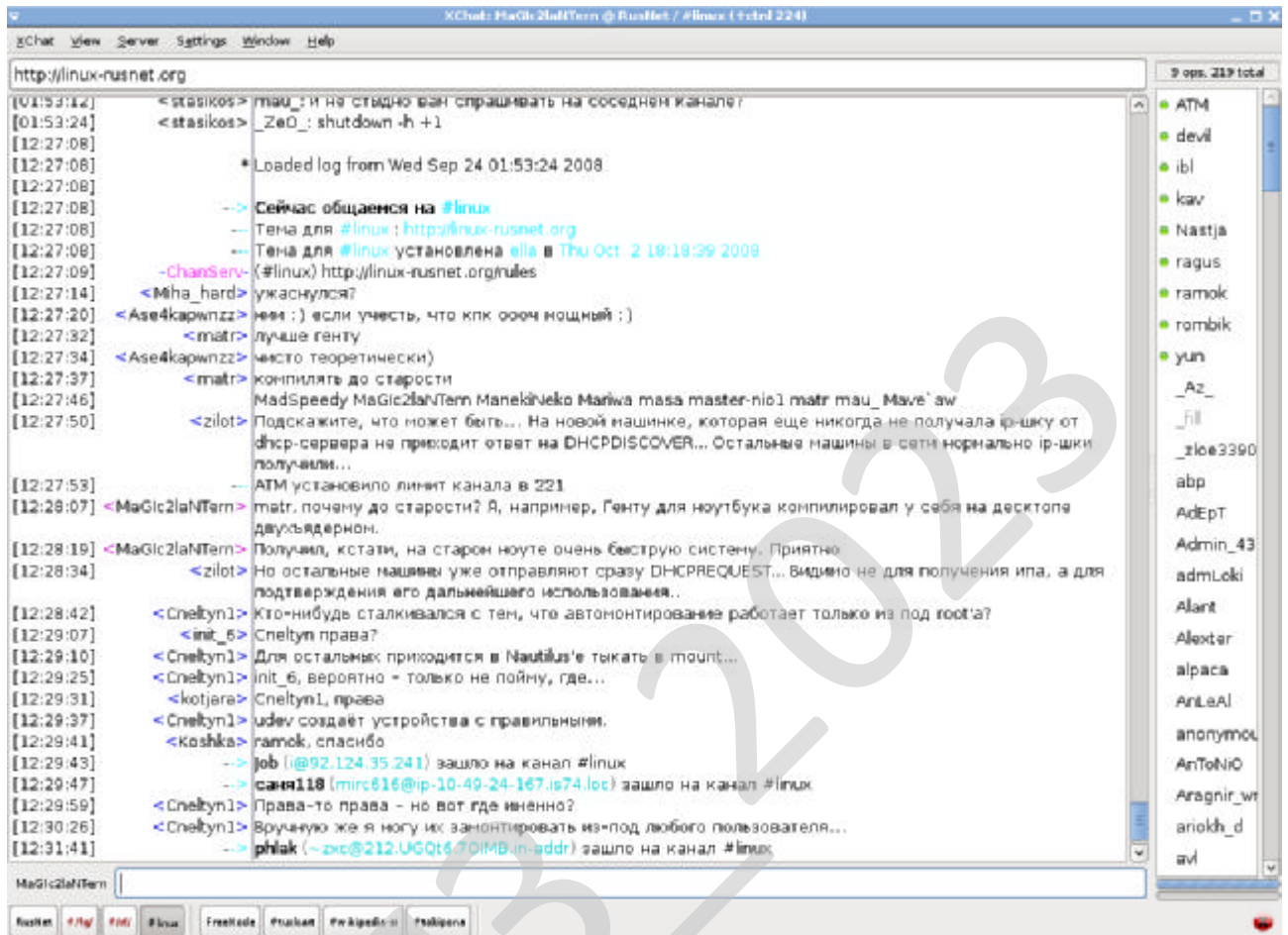


Рисунок 2.1 – Інтерфейс користувача xChat

– Opera – починаючи з версії 7.50 Opera поставляється з IRC клієнтом. Чат-контакти можна зберігати в адресній книзі, і ваші співрозмовники, коли вони перебувають у мережі, відображаються на панелі "Чат"

– Trillian – відрізняється підтримкою багатьох розповсюджених протоколів обміну миттєвими повідомленнями. Підтримує можливість з'єднання під декількома аккаунтами, на різні сервера. Є проблеми з кирилицею.

– Chatzilla – IRC-клієнт, виконаний як розширення для браузерів Mozilla Firefox, Mozilla, Seamonkey і Flock. Підтримує більшість функцій IRC-клієнтів,

такі як одночасне підключення до декількох серверів, IPv6, SSL, UTF-8, DCC. Зовнішній вигляд ChatZilla легко змінюється за допомогою CSS. Chatzilla можна запустити у вкладці браузера, набравши адресний рядок `chrome://chatzilla/content/chatzilla.xul`. Також можна запустити, не відкриваючи браузерне вікно Firefox. Для цього потрібно запустити Firefox із ключем "-chat".

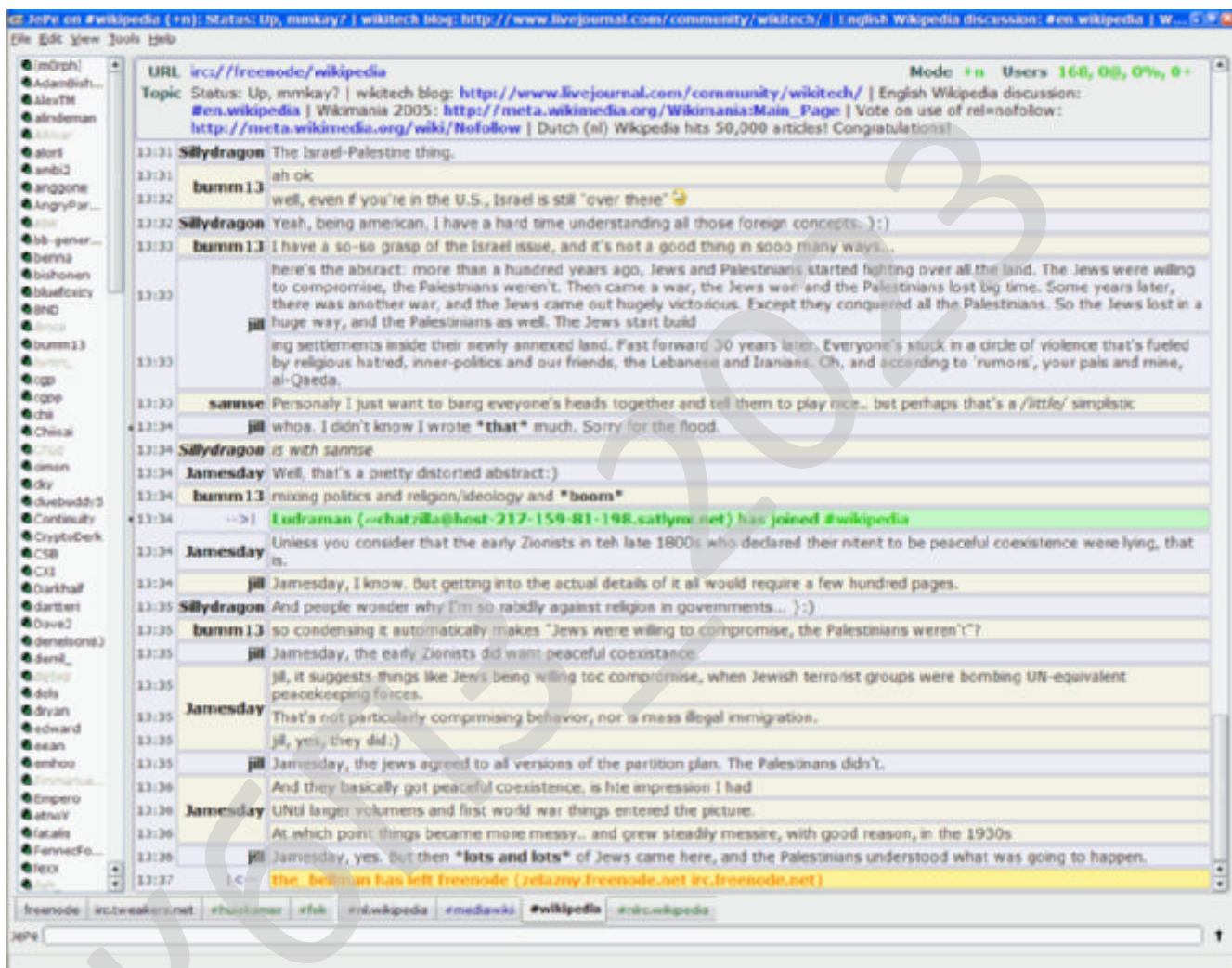


Рисунок 2.2 – Інтерфейс користувача Chatzilla

– Microsoft Chat – інша назва Comic Chat. В інтернеті можна знайти дві версії: 2 і 2.5, остання з яких датується 2002м роком. Чат-клієнт від Microsoft, уже заради цього стоїть на ньому глянути, не чи правда?





цьому клієнт працює на стареньких телефонах з монохромним дисплеєм і буде функціонувати навіть на Siemens C55.

– WLirc – клієнт дозволяє вибрати параметри з'єднання (через інтерфейси socket, HTTP або Bluetooth), підтримує роботу через проксі-сервер, уміє виконувати скрипти, проходити автентифікацію при з'єднанні з віддаленими ресурсами, міняти кодування й розміри шрифтів, плюс налаштовувати на свій лад зовнішній вигляд інтерфейсу програми й вибрати різні типи оповіщення про різні події в чаті. Є підтримка протоколу прямого спілкування між користувачами DCC. Саме із цієї причини автор WLirc оснастив мідлет убудованим файловим менеджером.

– jmlirc – багатовіконий інтерфейс із колірним підсвічуванням тексту, підтримка кодувань KOI 8-R і Windows-1251, можливість з'єднання через HTTP-проксі – все це вдалося вмістити в одному мідлеті, не вилазячи за рамки першої версії специфікації Mobile Information Device Profiles. У процесі роботи можна використовувати "гарячі" клавіші.

– InetTools IRC – вітчизняна розробка, русифікована й цілком функціональна. До послуг користувача різні профілі настроювань, маса операцій з тестом у чаті (цитування, копіювання, збереження, робота із шаблонами, стилями та ін.) і керування командами IRC-мережі (зміна ніку або теми, перегляд каналів, виконання певної дії, режим розмови й ін.). Додаток уміє вести облік мережного трафіку й працювати у фоновому режимі, дозволяючи власникові телефону робити дзвінки, не закриваючи клієнт і не виходячи із чату.

– SnIRC – Ще один український мідлет для роботи з IRC-мережами. Набір стандартний: профілі, керування оформленням тексту, автоідентифікація при підключенні до сервера, підтримка розширених користувальницьких статусів і кириличних кодувань у повному обсязі. По функціоналі програма близька до Virca і jmlirc.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

## 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

### Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

#### Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

– Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

– Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

### **Істотне поліпшення Delphi Code Insight**

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

### **Delphi Custom Managed Records**

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

### **Єдине керування пам'яттю**

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

### **Розширена підтримка бібліотек C++**

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

### **Win 64-відладник і збирач для C++**

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відлагодочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відлагодочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20



## **Поліпшена кроссплатформеність**

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

## **Оновлений менеджер пакетів Getit**

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

## **Універсальний інсталятор для установки Online і Offline**

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи IRC-клієнту.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23



- Псевдонім, aspell, charset, logger тощо.

Система IRC-клієнту пропонує багато інноваційних функцій , що відрізняє його від інших консольних/термінальних клієнтів чату.

Подивіться самі (перелік не вичерпний):

- 256 кольорів.
- Підтримка миші.
- Настроювані смуги.
- 8 скриптових мов.
- Поділ по горизонталі/вертикалі.
- Динамічна фільтрація рядків.
- Поступовий пошук тексту.
- Труба FIFO.
- Перевірка орфографії.
- Менеджер скриптів.
- Віддалені інтерфейси.
- Оновлення без виходу.

Повнофункціональний плагін IRC

Плагін IRC реалізує протокол, як описано в RFC 1459 і 2812.

Він поставляється з багатьма вбудованими функціями , такими як:

- Мультисервери.
- IPv6, SSL, проксі.
- Аутентифікація SASL.
- Ніклист.
- Колір для ніків.
- Колір для виїзних ніків.
- Розумний фільтр.
- Протипаводковий.
- Перенаправлення команд.
- Спеціальні відповіді CTCP.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

- Індикатор відставання.
- DCC (файл/чат).

### **Віддалені інтерфейси**

Зовнішні інтерфейси (графічні або текстові) можуть підключатися до запущеного Система IRC-клієнту.

Вони спілкуються з Система IRC-клієнту за допомогою протоколу ретрансляції.

### **Документація та підтримка**

Система IRC-клієнту перекладено кількома мовами та має вичерпну документацію , також перекладену.

- Ви новачок у Система IRC-клієнту? Прочитайте короткий посібник і посібник користувача.
- Є тривіальне запитання? FAQ чекає на вас!
- Ви хочете написати сценарій для Система IRC-клієнту? Подивіться посібник зі створення сценаріїв і посилання на API плагіна.
- Ваша рідна мова C? Тоді перегляньте посібник розробника та додайте свій внесок!

### **Особливості:**

- Універсальний додаток.
- Підтримка кількох серверів.
- Підтримка IRCCloud.
- Підключення у фоновому режимі (бета-версія).
- Збереження серверів і каналів.
- Автоматичне підключення під час запуску клієнта.
- Автоматичне приєднання до каналів.
- Закріпити сервери на початковому екрані.
- DCC Chat і DCC Send.
- Потужна підтримка сценаріїв (автоматичні дії).
- Виконує команди на основі Правил.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

- Автоматичний запуск команд після підключення.
- Псевдоніми.
- Перемикайте перегляди сервера, проводячи пальцем.
- Кольорові клички.
- Змінити кодування тексту.
- Підтримка кодів форматування тексту mIRC.
- Вбудовані зображення.
- Відтворення рідного буфера.
- Мітки часу.
- Автоматичне повторне підключення після відключення.
- Параметри шрифту повідомлення каналу.
- Автоматичні пропозиції для IRC-команд, псевдонімів і каналів.
- Автозаповнення (TAB).
- Підключіться за допомогою SSL.
- Ігноруйте помилки сертифіката SSL.
- Автентифікація за допомогою PLAIN SASL.
- Автоматична ідентифікація за допомогою Nickserv.
- Паролі серверів.
- Тип історії.
- Список URL.
- Список каналів сервера.
- Список ігнорування.
- Режим управління.
- HTTP, SOCKS4 і SOCK5 проксі.
- Шифрування FiSH.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

### 3.2 Розробка структурної схеми

Протокол IRC був розроблений для систем, що використовують TCP/IP мережевий протокол, хоча немає вимог, щоб він залишався єдиною сферою, в якій віна діє.

Сама IRC є системою телеконференцій, яка (за допомогою клієнт-серверної моделі) добре підходить для роботи на багатьох машинах розподіленим способом. Типове налаштування передбачає один процес (сервер), утворюючи центральну точку для клієнтів (або інших серверів) для підключення, виконуючи необхідну доставку/мультиплексування повідомлень та інші функції.

#### Сервери

Сервер утворює магістраль IRC, забезпечуючи точку, до якої клієнти можуть підключитися, щоб поговорити один з одним, і точка для інших сервери для підключення, утворюючи мережу IRC. Єдина мережа дозволеною конфігурацією для IRC-серверів є *spanning tree*, де кожен сервер діє як центральний вузол для решти *net* це бачить.

#### Клієнти

Клієнт – це все, що підключається до сервера, але не є іншим сервер. Кожен клієнт відрізняється від інших клієнтів унікальністю псевдонім із максимальною довжиною дев'яти (9) символів. Див правила граматики протоколу щодо того, що можна, а що не можна використовувати в а псевдонім. На додаток до псевдоніму всі сервери повинні мати наступна інформація про всіх клієнтів: справжнє ім'я хоста на якому працює клієнт, ім'я користувача клієнта на цьому хост і сервер, до якого підключений клієнт.

#### Оператори

Щоб дозволити зберігати в IRC розумну кількість замовлень мережі дозволено виконувати спеціальний клас клієнтів (операторів). загальні функції обслуговування в мережі. Хоча повноваження надані оператору, можна вважати «небезпечними». все ж потрібно. Оператори повинні вміти виконувати основні

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

мережеві завдання, такі як відключення та повторне підключення серверів необхідні для запобігання тривалому використанню поганої мережевої маршрутизації. Визнання цієї потреби передбачено протоколом, який тут обговорюється оператори лише для того, щоб мати можливість виконувати такі функції.

Більш суперечливою повноваженням операторів є можливість видаляти а користувача з підключеної мережі «примусово», тобто оператори можуть щоб закрити з'єднання між будь-яким клієнтом і сервером. The виправдання цього є делікатним, оскільки зловживання є і тим, і іншим руйнівний і дратівливий.

### **Канали**

Канал – це названа група з одного або кількох клієнтів, які будуть усі отримувати повідомлення, адресовані цьому каналу. Канал створений неявно, коли до нього приєднується перший клієнт, і канал припиняє існує, коли останній клієнт залишає його. Поки канал існує, будь-який клієнт може посилатися на канал, використовуючи назву каналу.

Назви каналів – це рядки (починаються з символу «&» або «#») з довжиною до 200 символів. Окрім вимоги, що перший символ – "&" або "#"; єдине обмеження на а назва каналу полягає в тому, що воно не може містити жодних пробілів (' '), елемент керування G (^G або ASCII 7), або кома (',', яка використовується як елемент списку роздільник за протоколом).

Існує два типи каналів, дозволених цим протоколом. Один – це а розподілений канал, який відомий усім наявним серверам підключений до мережі. Ці канали позначені першим персонаж, який є єдиним клієнтом на сервері, де він існує, може приєднатися це. Вони відрізняються символом «&» на початку. На вершині для цих двох типів доступні різні режими каналів змінювати характеристики окремих каналів.

Щоб створити новий канал або стати частиною існуючого каналу, користувач необхідний для ПРИЄДНАННЯ до каналу. Якщо канал не існує раніше щоб приєднатися, канал створюється, а користувач, який створює, стає а

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

оператор каналу. Якщо канал уже існує, незалежно від того, чи є ваш запит на ПРИЄДНАННЯ, що канал виконується, залежить від поточних режимів каналу. Наприклад, якщо канал призначений лише для запрошення, (+i), тоді ви можете приєднатися лише за запрошенням. Як частина протоколу користувач може бути частиною кількох каналів одночасно, але обмеження до десяти (10) каналів рекомендується як достатній як для досвідчених, так і для початківці користувачі.

Якщо мережа IRC роз'єднується через розкол між двома серверів, канал на кожній стороні складається лише з цих клієнтів які підключені до серверів на відповідних сторонах розриву, можливо, припинення існування на одній стороні розколу. Коли розкол зцілюється, підключені сервери оголошують один одному, хто вони think є в кожному каналі та в режимі цього каналу. Якщо канал існує з обох сторін, JOIN і MODE інтерпретуються в інклюзивним способом, щоб обидві сторони нового з'єднання були домовитися про те, які клієнти знаходяться в каналі та які режими канал має.

### **Оператори каналів**

Оператор каналу (також відомий як "chan" або "chanop") на а даний канал вважається "власником" цього каналу. На знак визнання У цьому статусі оператори каналів наділяються певними повноваженнями, які дозволити їм зберегти контроль і певну міру розсудливості у своєму каналі. Як власник каналу, оператор каналу не зобов'язаний мати причини своїх дій, хоча якщо їхні дії заг антисоціальний або інший образливий, було б розумно запитати IRC оператору втрутитися, або для користувачів просто залишити та піти в інше місце і створюють власний канал.

Оператор каналу позначається символом «@» поруч із ним псевдонім, коли він пов'язаний з каналом (тобто відповідає на NAMES, WHO та WHOIS команди).

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30



Наявність префікса позначається одним початковим символом ASCII символ двокрапки (':', 0x3b), який має бути першим символом у саме повідомлення. Між двокрапкою не повинно бути проміжків (пробілів). і префікс. Префікс використовується серверами для позначення істини походження повідомлення. Якщо в повідомленні відсутній префікс, він передбачається, що походить від зв'язку, з якого він був отримано. Клієнти не повинні використовувати префікс, надсилаючи повідомлення з самі; якщо вони використовують префікс, єдиним дійсним префіксом є зареєстрований псевдонім, пов'язаний з клієнтом. Якщо джерело ідентифікований префіксом неможливо знайти на внутрішньому сервері бази даних, або якщо джерело зареєстровано за іншим посиланням, ніж з якого надійшло повідомлення, сервер повинен ігнорувати повідомлення мовчки.

Команда має бути дійсною командою IRC або тризначною (3). число, представлене в тексті ASCII.

Повідомлення IRC завжди складаються з рядків символів, що закінчуються символом CR-LF (Повернення каретки – Переведення рядка) пари, і ці повідомлення не мають перевищує 512 символів, враховуючи всі символи включно задній CR-LF. Таким чином, максимально дозволено 510 символів для команди та її параметрів. Не передбачено рядки повідомлень продовження.

### **Структура IRC-мережі**

Представимо структурну схему IRC мережі. Тут об'єктами є додатки (не комп'ютери). Технічно все це може розташовуватися й на одному комп'ютері. Напрямок стрілок означає порядок з'єднання (хто до кого підключається).

Тут ви бачите 2 сервери, одні сервіси, і клієнтів, підключених до сервера. Тепер поясню, навіщо всього так багато, і як все це працює. Перше, про що обмовлюся, це кількість серверів. У вашій мережі повинен бути як мінімум один сервер. У більшості випадків його одного буде досить. Більша кількість серверів потрібна, якщо ви, наприклад, хочете об'єднати чат у своїй локальній мережі із

чатом у сусідній мережі. Далі буде обговорюватися мережа, що складається з одного сервера (але все сказане справедливо й для багатосерверної мережі).

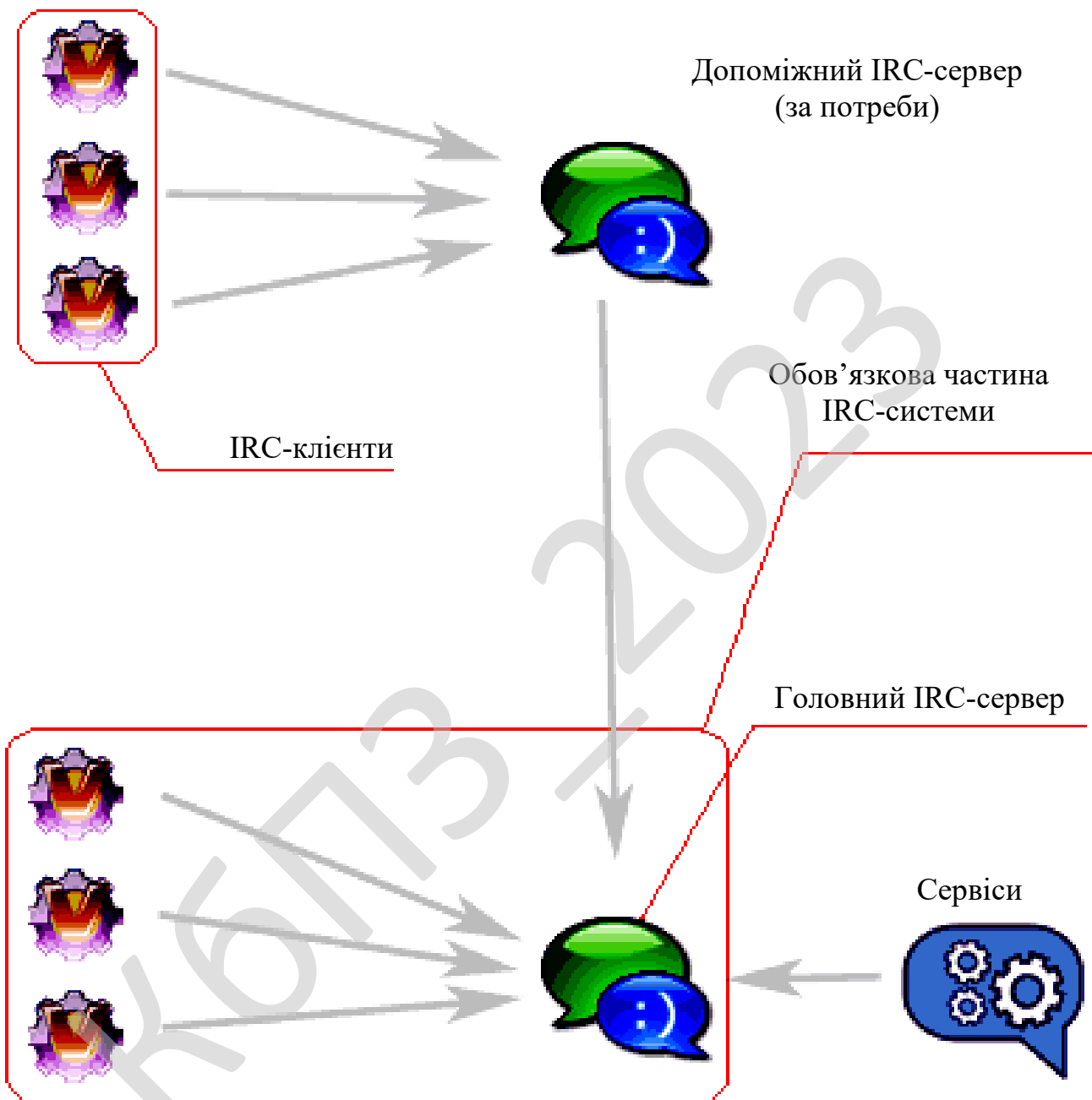


Рисунок 3.1 – Структурна схема системи

Сервер (IRC демон, IRCD). Для чого потрібний сервер, думаю, зрозуміло. До нього підключаються клієнти. Сервер типово працює на 6667 порту, тому для його запуску не потрібні (а також не рекомендуються) права суперкористувача (root). Серверне ПЗ кроссплатформене: ви можете його використовувати як на \*nix-подібних операційних системах, так і на Windows, також воно не вибагливо до заліза сервера (підійде й старий домашній комп'ютер). Настроювання серверного ПЗ буде розглянута в розділі настроювання.

Ваш сервер повинен бути доступний 24 години на добу, 7 днів у тиждень. У момент його відсутності люди не зможуть спілкуватися через IRC.

### **IRC-сервіси**

Сервіси – опціональна частина IRC мережі. Тобто ви можете вільно обійтися й без них. Але вони помітно полегшать вам життя. Сервіси здійснюють реєстрацію й контроль IRC каналів і ніків. Тобто без них, коли Ви виходите з IRC, ваш нік може бути використаний ким-небудь іншим, а оператором каналу стає перший хто зайде на нього. Сервіси запобігають такій ситуації. Вони також кроссплатформені й не вибагливі до ресурсів сервера.

Сервіси підключаються до одного із серверів (головному) по TCP/IP і тому можуть розташовуватися як на тому же фізичному сервері, що й IRCD, так і на окремому комп'ютері. Рекомендується тримати сервер і сервіси на одному комп'ютері.

IRC-сервіси – допоміжні боти в IRC, використовувані для керування й адміністрування мереж IRC, які звичайно містять у собі:

- NickServ – сервіс, що управляє користувачами.
- ChanServ – сервіс, що управляє каналами.
- MemoServ – сервіс, що дозволяє відправляти замітки, коли користувач не в мережі.
- OperServ (RootServ) – сервіс, що дозволяє операторам мережі управляти нею.
- HelpServ – надає довідку по сервісах IRC.

Також деякі версії сервісів містять:

– BotServ – надає можливість власникам каналів запрошувати на канал ботів.

– HostServ (іноді включений в NickServ) – сервіс, що дозволяє змінювати реальний IP-Адресу користувача на певну маску.

– StatServ – сервіс статистики.

– DevNull – ігнорує всі дані, що посилаються.

– SeenServ – виводить, коли востаннє користувач з'являвся на сервері.

Користувачі. Користувачі можуть використовувати будь-який IRC клієнт, який їм заманеться. Із цією роллю прекрасно впорається розроблена в ході виконання магістерського проектування програма, надавши користувачеві гарний сучасний інтерфейс, смайли, аватари, потужний скриптинговий движок.

### 3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Функціональні можливості розробленої системи:

1. Головне вікно інтерфейсу користувача програми.
2. Багатомовний інтерфейс.
3. Підтримка скинів, створення скинів користувачем.
4. Індикація активності IRC.
5. Повна сумісність ядра з Unicode UTF-8.
6. Робота через проксі сервери (SOCKSv4, SOCKSv5, HTTP).
7. Робота з IPv6, включаючи DCC.
8. Використання SSL, включаючи IPv6 і DCC.
9. Шифрування (можна використовувати на каналах, у приваті або в DCC):
  - Micryption.

- Rijndael256Base64.
- Rijndael192Base64.
- Rijndael128Base64.
- Rijndael256Hex.
- Rijndael192Hex.
- Rijndael128Hex.
- Lamerizer.

10. Підтримка модульних розширень завантажуваних і вивантажуваних за потребою.
  11. Необмежена можливість розширення за рахунок універсального API.
  12. Підтримка одночасного з'єднання з різними IRC-серверами з різною конфігурацією профілю.
  13. Упакування логів у фоновому режимі.
  14. DCC CHAT (чат) і SEND (з підтримкою дозавантаження).
  15. Додаткові DCC: VOICE, RSEND, GET, RECV.
  16. DCC CHAT (чат) через SSL.
  17. Фонове лімітування швидкості DCC з'єднання.
  18. Інтегрований движок скриптингу.
  19. Можливість використання об'єктно-орієнтованого програмування.
- Розглянемо деякі з понять які визначені на функціональній схемі.

Проксі-сервер (від англ. проху – «представник, уповноважений») – служба (комплекс програм) у комп'ютерних мережах, що дозволяє клієнтам виконувати непрямі запити до інших мережних служб. Спочатку клієнт підключається до проксі-сервера й запитує який-небудь ресурс (наприклад, e-mail), розташований на іншому сервері. Потім проксі-сервер або підключається до зазначеного сервера й одержує ресурс у нього, або повертає ресурс із власного кешу (у випадках, якщо проксі має свій кеш). У деяких випадках запит клієнта або відповідь сервера може бути змінений проксі-сервером у певних цілях. Також

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

проксі-сервер дозволяє захищати клієнтський комп'ютер від деяких мережних атак і допомагає зберігати анонімність клієнта.

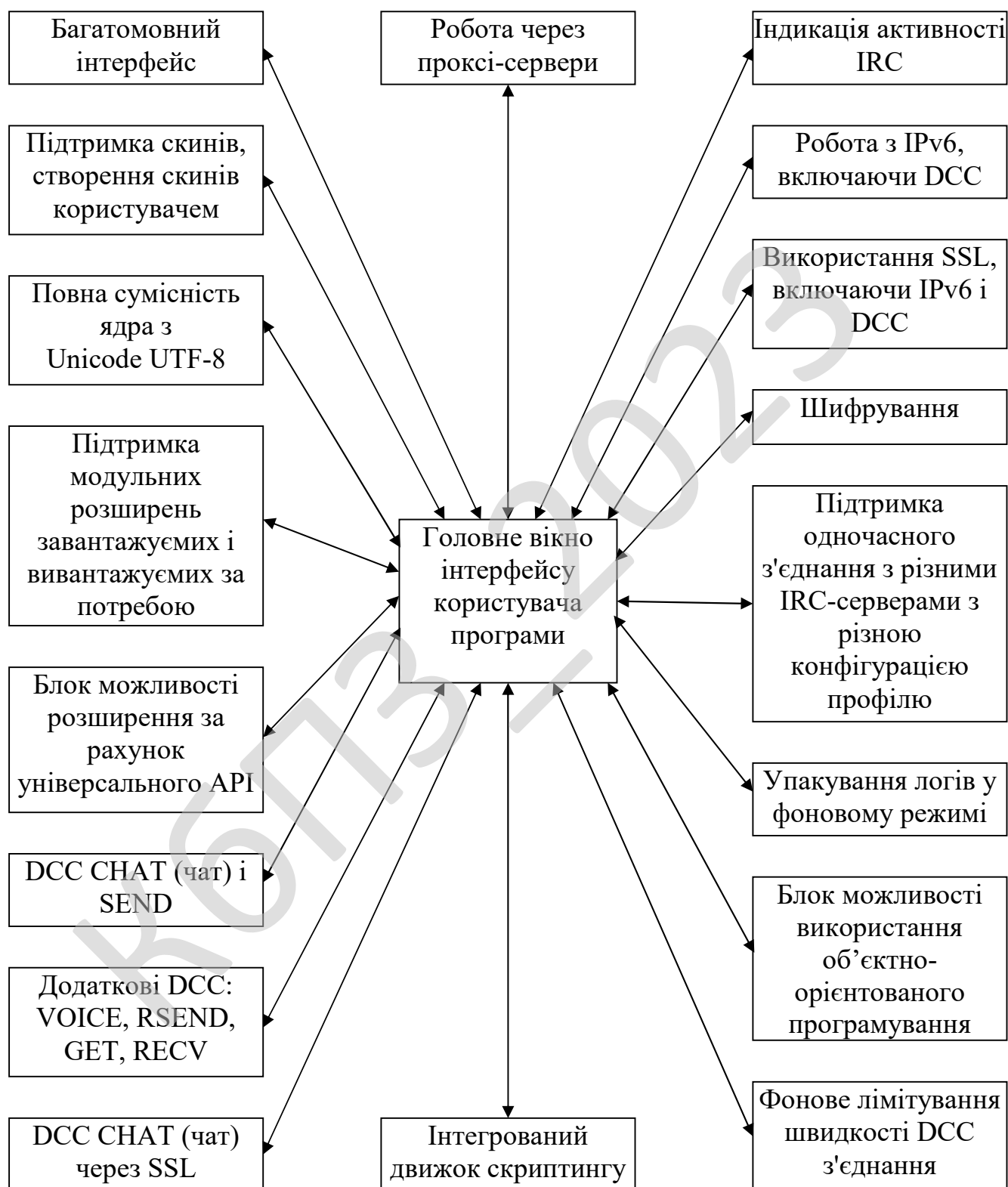


Рисунок 3.2 – Функціональна схема системи

SOCKS – мережний протокол, що дозволяє клієнт-серверним додаткам прозоро використовувати сервіси за міжмережними екранами (фаєрволами). SOCKS – це скорочення від «SOCKet Secure».

HTTP (англ. HyperText Transfer Protocol – «протокол передачі гіпертексту») – протокол прикладного рівня передачі даних (споконвічно – у вигляді гіпертекстових документів). Основою HTTP є технологія «клієнт-сервер», тобто передбачається існування споживачів (клієнтів), які ініціюють з'єднання й надсилають запит, і постачальників (серверів), які очікують з'єднання для одержання запиту, роблять необхідні дії й повертають назад повідомлення з результатом. HTTP у цей час повсюдно використовується у Всесвітній павутині для одержання інформації з веб-сайтів. В 2012 році в Північній Америці частка HTTP-трафіку перевищила частку P2P-мереж і склала 46 %, з яких майже половина – це передача потокового відео й звуку. HTTP використовується також у якості «транспорту» для інших протоколів прикладного рівня, таких як SOAP, XML-RPC, WebDAV. Основним об'єктом маніпуляції в HTTP є ресурс, на який указує URI (англ. Uniform Resource Identifier) у запиті клієнта. Звичайно такими ресурсами є файли, що зберігаються на сервері, але ними можуть бути логічні об'єкти або щось абстрактне. Особливістю протоколу HTTP є можливість указати в запиті й відповіді спосіб подання того самого ресурсу по різних параметрах: формату, кодуванню, мові й т.д. (Зокрема для цього використовується HTTP-заголовок.) Саме завдяки можливості вказівки способу кодування повідомлення клієнт і сервер можуть обмінюватися двійковими даними, хоча даний протокол є текстовим. HTTP – протокол прикладного рівня, аналогічними йому є FTP і SMTP. Обмін повідомленнями йде за звичайною схемою «запит-відповідь». Для ідентифікації ресурсів HTTP використовує глобальні URI. На відміну від багатьох інших протоколів, HTTP не зберігає свого стану. Це означає відсутність збереження проміжного стану між парами «запит-відповідь». Компоненти, що використовують HTTP, можуть самостійно здійснювати збереження інформації про стан, пов'язаної з останніми запитами й відповідями (наприклад, «кукі» на

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38



ТСР/ІР, причому для шифрування використовується асиметричний алгоритм із відкритим ключем. При шифруванні з відкритим ключем використовуються два ключі, відкритий і секретний, причому кожної з них може використовуватися для шифрування повідомлення. Якщо для шифрування повідомлення був використаний відкритий ключ, то для розшифровки повинен використовуватися секретний, і навпаки. У такій ситуації можливі два способи використання ключів. По-перше, сторона, що зберігає в таємниці секретний ключ і яка оприлюднила відкритий, може приймати від протилежної сторони повідомлення, зашифровані відкритим ключем, які не може прочитати ніхто, крім неї (адже для розшифровки потрібен секретний ключ, відомий тільки їй). По-друге, за допомогою закритого ключа сторона-власник закритого ключа може створювати зашифровані повідомлення, які може прочитати хто завгодно (адже для розшифровки потрібний відкритий ключ, доступний усім), але при цьому що прочитав може бути впевнений, що це повідомлення було створено стороною-власником секретного ключа.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

### 3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерського проектування, наведена на рисунку 3.3.

Першим процесом, який завантажується у системі, є процес виведення головного вікна програми. Він взаємодіє з наступними процесами:

- Процес налаштування ІРС-чату.
- Процес запису лог-файлу.
- Процес введення логіну та паролю.

Процес налаштування ІРС-чату взаємодіє з наступними процесами:

- Процес визначення режиму роботи.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

– Процес визначення параметрів з'єднання.

Процес введення логіну та паролю взаємодіє з процесом підключення до сервера / відключення від сервера, який, у свою чергу, взаємодіє з процесом визначення списку каналів. Останній процес взаємодіє з процесом вибору каналу, які, у свою чергу, взаємодіє з наступними процесами:

- Процес додавання вибраного каналу в «Обране».
- Процес визначення властивостей каналу.
- Процес підключення до каналу / відключення від каналу.

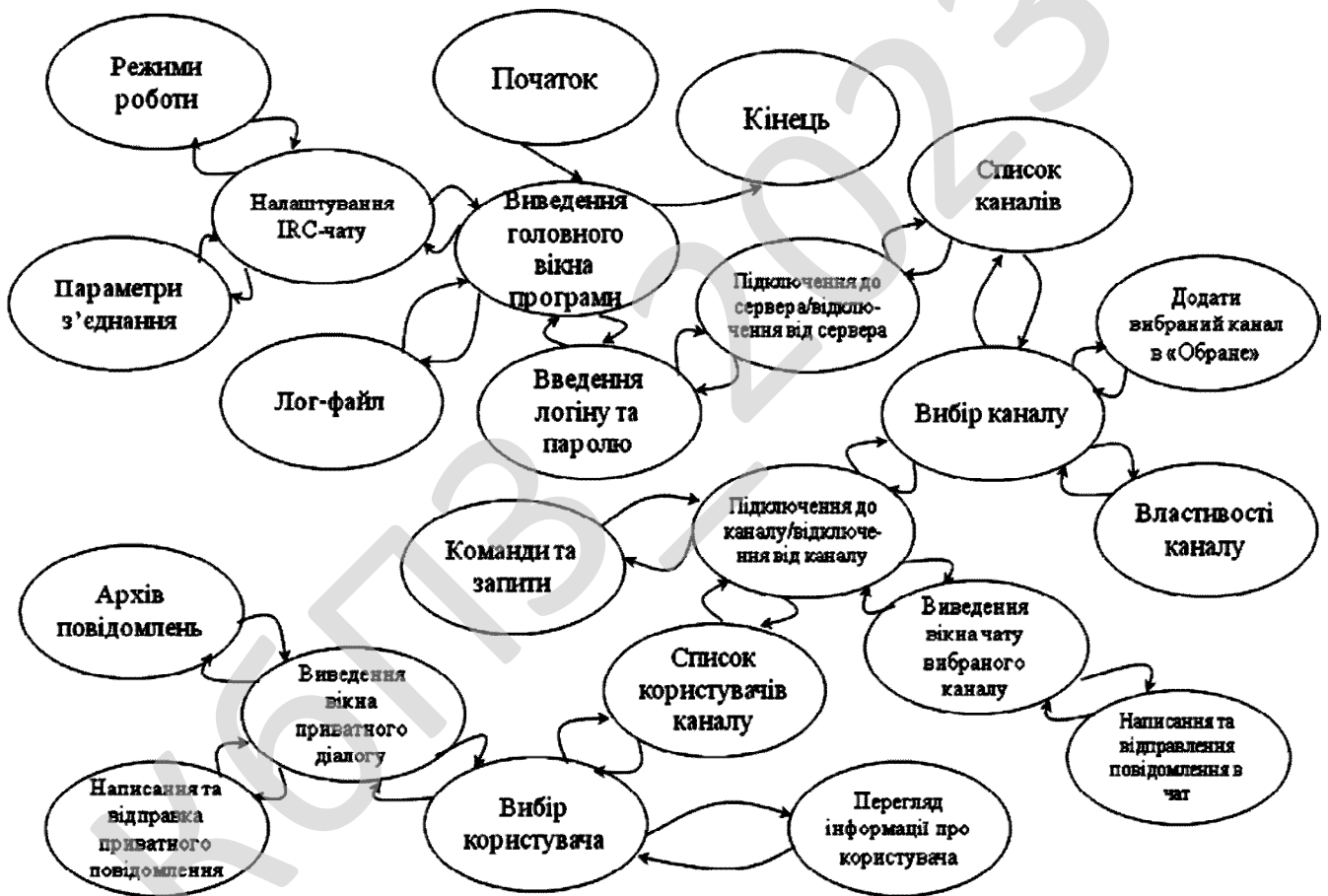


Рисунок 3.3 – Діаграма взаємодії процесів

Процес підключення до каналу / відключення від каналу взаємодіє з наступними процесами:

- Процес виведення вікна чату вибраного каналу, який взаємодіє з

процесом написання та відправлення повідомлення в чат.

- Процес визначення списку користувачів каналу.
- Процес реалізації команд та запиту.

Процес визначення списку користувачів каналу взаємодіє з процесом вибору користувача, який, у свою чергу взаємодіє з наступними процесами:

- Процес перегляду інформації про користувача.
- Процес виведення вікна приватного діалогу.

Процес виведення вікна приватного діалогу взаємодіє з наступними процесами:

- Процес архівації повідомлень.
- Процес написання та відправлення приватного повідомлення.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

# 4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

## 4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми.

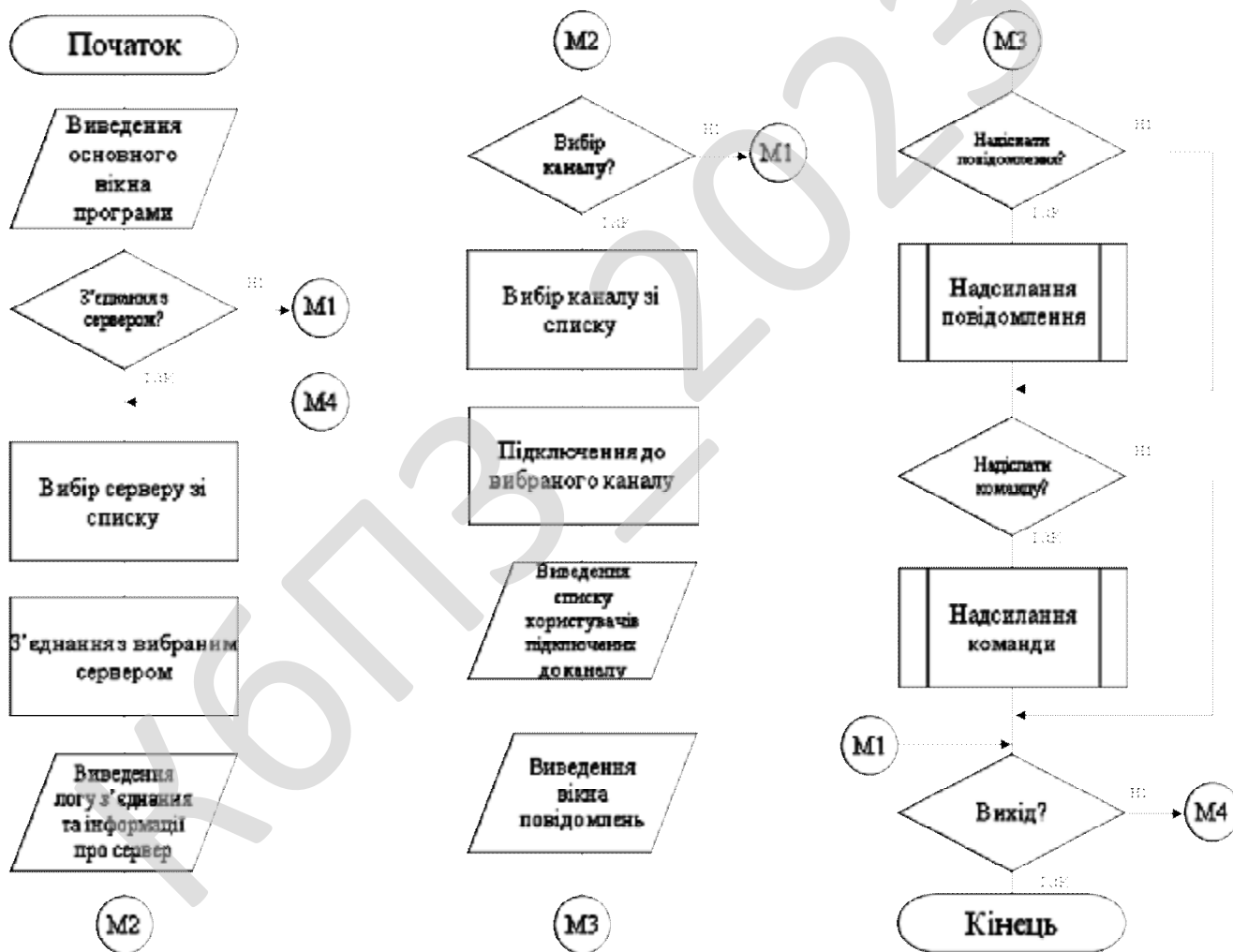


Рисунок 4.1 – Блок-схема основної програми

Її робота складається з виконання наступної послідовності кроків:

Крок 1. Виведення основного вікна програми на екран.

Крок 2. Якщо користувач бажає здійснити з'єднання з сервером, то відбувається вибір серверу з наявного списку або введення адреси серверу вручну вручну.

Крок 3. Здійснення з'єднання з сервером.

Крок 4. Виведення логу з'єднання та інформації про сервер.

Крок 5. Відбувається вибір каналу зі списку, або введення вручну назви каналу та команди з'єднання.

Крок 6. Підключення до вибраного каналу.

Крок 7. Виведення списку користувачів підключених до каналу.

Крок 8. Виведення вікна повідомлень.

Крок 9. Якщо відбувається передача або прийом повідомлень, то викликається підпрограма обробки текстових повідомлень.

Крок 10. Якщо відбувається насилання команд або запитів до сервера, то викликається підпрограма надсилання команд.

Крок 11. Здійснюється очікування та виконання дій користувача.

Крок 12. Якщо обрано дію вихід з програми, то відбувається від'єднання від відкритих каналів та серверу і закриття програми.

На рисунку 4.2 наведена робота підпрограми надсилання/прийняття текстових повідомлень. Робота даної підпрограми складається з наступної послідовності кроків:

Крок 1. Якщо відбувається відправка текстового повідомлення в загальний чат, то здійснюються кроки 2-5.

Крок 2. Введення тексту повідомлення.

Крок 3. Надсилання повідомлення в загальний чат.

Крок 4. Виведення ніку користувача у вікні чату.

Крок 5. Виведення повідомлення у вікні чату.

Крок 6. Якщо відбувається відправка приватного текстового

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

повідомлення, то здійснюються кроки 7-11.

Крок 7. Вибір користувача зі списку користувачів приєднаних до даного каналу.

Крок 8. Введення текстового повідомлення.

Крок 9. Надсилання приватного повідомлення вибраному користувачу.

Крок 10. Виведення ніку користувача у вікні приватних повідомлень.

Крок 11. Виведення повідомлення у вікні приватних повідомлень.

Крок 12. Якщо прийшло повідомлення із загального чату, то здійснюються кроки 13-14.

Крок 13. Виведення ніку користувача, від якого прийшло повідомлення, у вікні загального чату.

Крок 14. Виведення прийнятого повідомлення у вікні чату.

Крок 15. Якщо прийшло приватне повідомлення, то здійснюються кроки 16-18.

Крок 16. Відкриття вікна приватних повідомлень з користувачем, від якого надійшло повідомлення.

Крок 17. Виведення ніку у вікні приватних повідомлень.

Крок 18. Виведення повідомлення у вікні приватних повідомлень.

Розглянемо детальніше принципи роботи IRC-протоколу.

### **Протокол IRC**

Протокол IRC входить у стандартний набір Інтернет-протоколів та опублікований як RFC, саме як засіб зв'язку декількох чат-серверів працюючих у різних сегментах Інтернету. Основне достоїнство даного протоколу – можливість зв'язувати між собою декілька територіально розрізнених серверів.

Відправлення повідомлення в IRC на рівні протоколу виглядають у такий спосіб:

PRIVMSG АДРЕСАТ :текст повідомлення

Тут АДРЕСАТ може бути ім'ям каналу або ніком.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

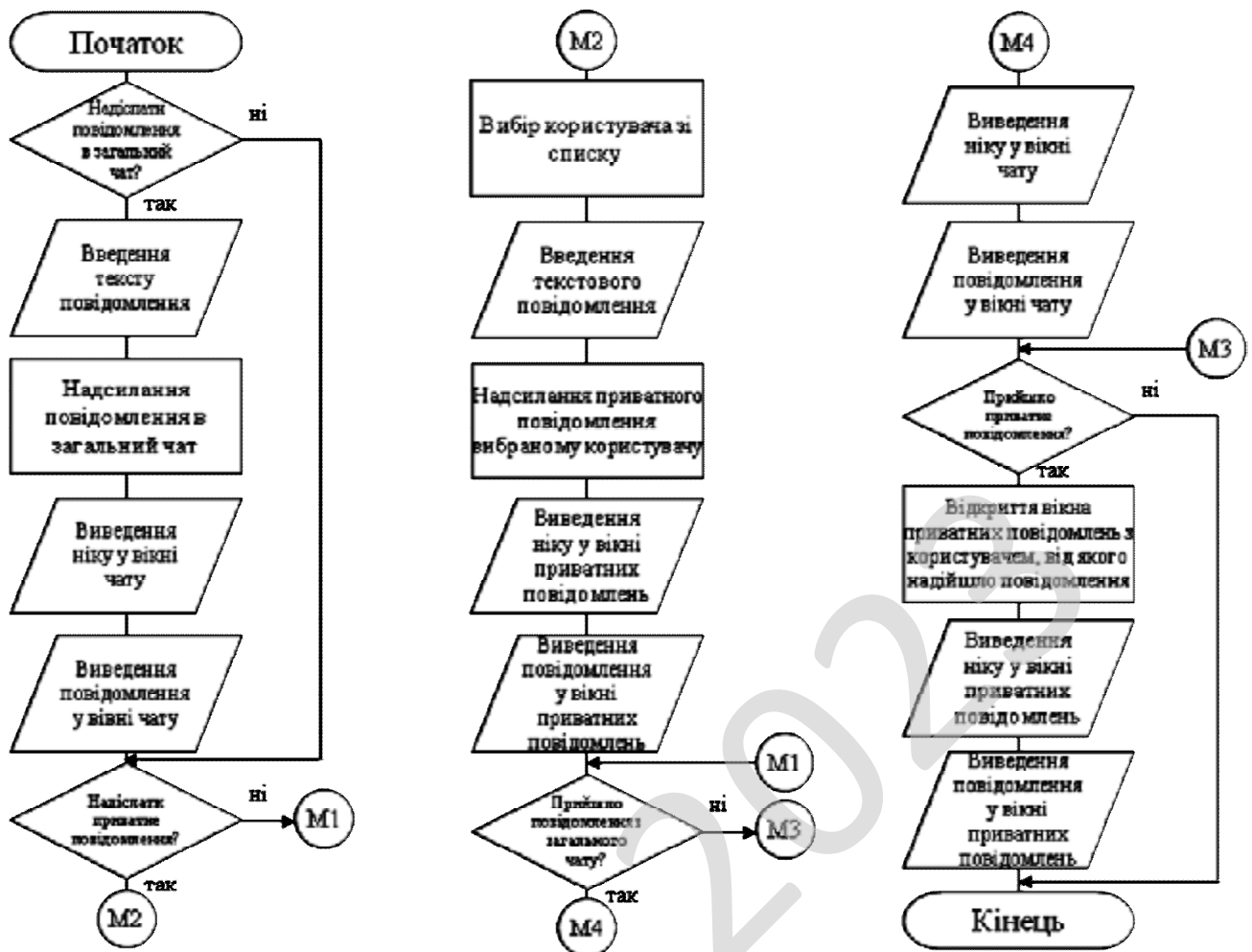


Рисунок 4.2 – Блок-схема підпрограми надсилання/прийняття текстових повідомлень

На рисунку 4.3 в центрі розташований світч, розглянемо маршрути пакетів:

- Пакет відправляється клієнтом на світч.
- Зі світча він направляється на сервер.
- На сервері створюється пакет для кожного іншого клієнта й всі вони відправляються на світч.
- Світч розсилає пакети, що прийшли із сервера, по одному кожному клієнту.

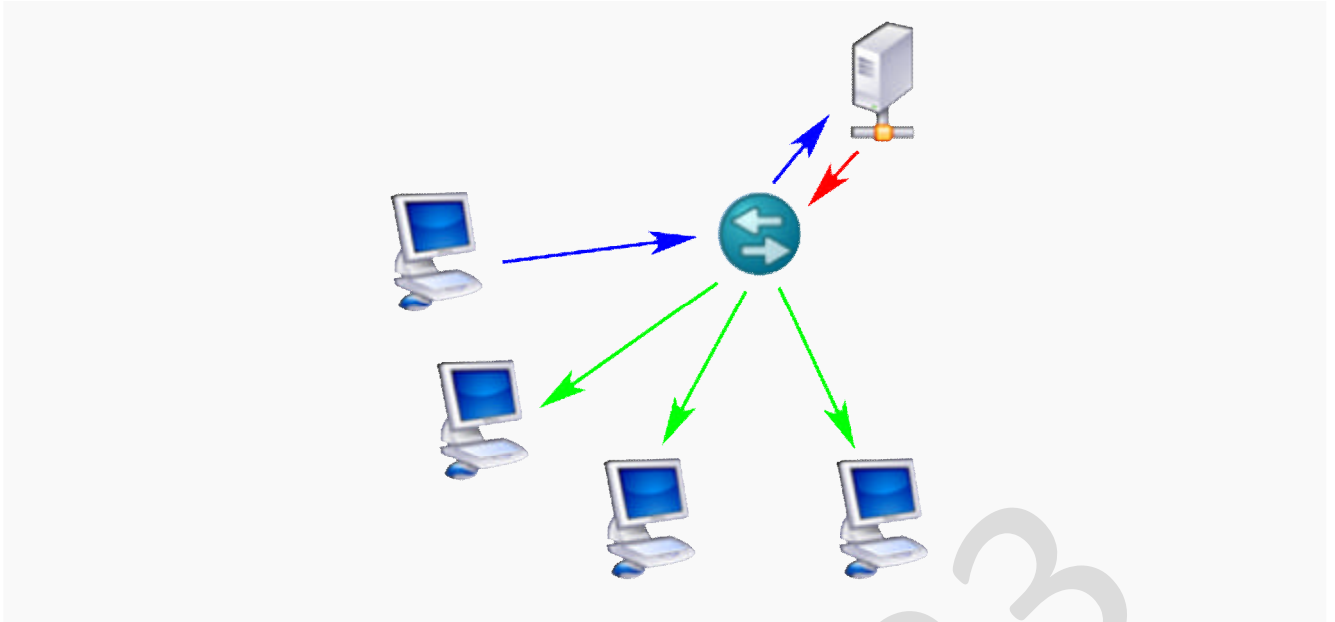


Рисунок 4.3 – Схема проходження пакетів при IRC з'єднанні

IRC-протокол розроблений для систем, що використовують мережний протокол TCP/IP.

IRC являє собою систему телеконференцій, які (через використання моделі клієнт-сервер) добре підходять для роботи на багатьох машинах у розповсюдженій зоні. Типове настроювання представляє собою процес (сервер), що надає доступ для клієнтів (або інших серверів), і виконуючу доставку/мультиплексування повідомлень та інших функцій.

### **Сервери**

Сервер формує бекбон IRC, що надає точки приєднання клієнтів для спілкування й приєднання інших серверів для формування IRC-мережі.

Сервери, доступні в мережі формують IRC-мережу, утворюючи мережне дерево (див. рис. 4.4), у якому кожний сервер є самостійним, але проте взаємодіючим з іншими серверами. (Подібна форма мережі припускає найбільшу безпеку мережі, навіть при падінні одного або декількох серверів).

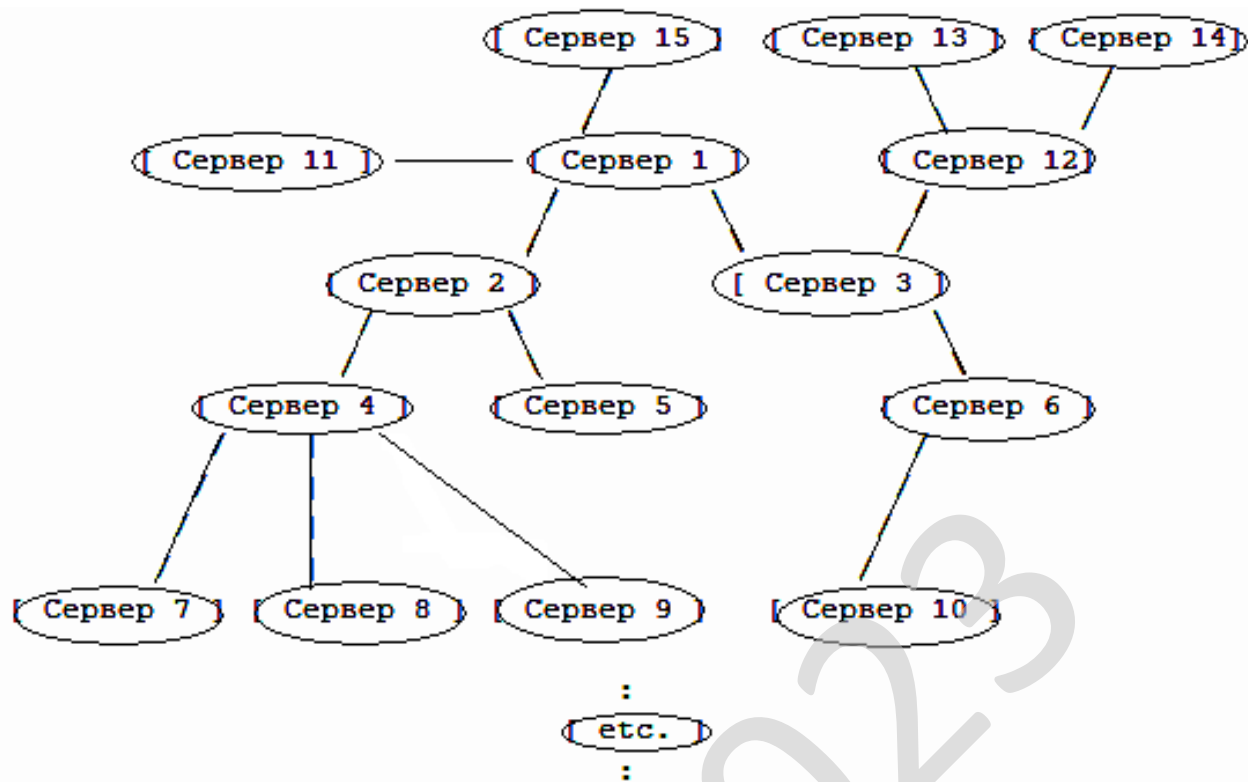


Рисунок 4.4 – Схематичне зображення IRC-мережі

### Клієнти

Клієнт це будь-яке приєднання до сервера, яке не є сервером. Кожний клієнт відрізняється від інших клієнтів наявністю унікального (не схожого ні на чий інший) нікнейма, що має довжину не більше 9-ти символів. На додаток до нікнейму, всі сервери повинні мати наступну інформацію про всіх клієнтів: справжнє ім'я хоста, з якого запустився клієнт, ім'я користувача клієнтів на цьому хості й сервер, до якого приєднався клієнт.

### Оператори

Для підтримки порядку в IRC-мережі, існує спеціальний клас клієнтів (оператори, у даний час їх називають «іркопи»). Хоча, можливості операторів можна розглядати як «небезпечні», вони не підкоряються наказам. Оператори виконують основні мережні завдання, такі як від'єднання й перез'єднання серверів для поліпшення стану мережі або виправлення яких-небудь мережних помилок.

На додаток до можливостей операторів можна додати, що в їхніх силах

так само закрити з'єднання між клієнтів і сервером. Правда, подібно це буває тільки при деструктивних і дратівних діях клієнта.

## Канали

Канал – це позначення групи з одного або більшого числа клієнтів, які одержують повідомлення, адресовані в цей канал. Канал створюється при з'єднанні першого клієнта з ним і канал зникає, коли його залишає останній клієнт. Поки канал відсутній, будь-які клієнти можуть заволодіти каналом, назвавши свій канал таким же ім'ям.

Імена каналів – рядок (що починається із символу '&' або '#') довжиною до 200 символів. Остеронь від вимог, що перший символ повинен бути або '&' або '#'; обмеження на те, що назва каналу не може містити пробілів (' '), Ctrl-G (^G або ASCII 7), або ком (',' яка використовується для створення списку каналів). Протоколом надається два типи каналів, один розповсюджуваний канал, що відомий всім серверам, приєднаним до мережі.

Ці канали позначені першим символом; доступні тільки тим клієнтам, на сервері яких він існує. Такі канали відрізняються початковим символом '&'.

Для створення нового каналу або входу в існуючий, користувач повинен запросити JOIN каналу. Якщо канал був відсутній, то канал створюється й користувач, що ввійшов, стає оператором каналу. Якщо канал уже існує, але так чи інакше він не відповідає на спроби ввійти, значить у налаштуваннях каналу встановлений який-небудь із нижчеподаних режимів. Можливо, цей канал тільки-для-запрошених (invite-only), (режим +i), і ви зможете на нього ввійти тільки будучи запрошеним. Користувач може перебувати в декількох каналах одночасно, але рекомендується обмежитися 10-ма каналами.

Якщо в IRC-мережі відбувається розрив, викликаний роз'єднанням двох серверів, канал так само розривається на декілька частин, у яких залишаються користувачі, що сидять на своїх серверах. Коли сервери з'єднуються знову, вони відновлюють частини каналу і його режими. Якщо канал доступний по різні сторони, входження й режими каналу інтерпретуються у своїх манерах.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49



описано в попередньому розділі, клієнту варто відповісти, але це не означає, що завжди можна дочекатися відповіді; зв'язок клієнт-сервер і сервер-сервер дуже розсинхронізовані по своїй природі.

Кожне IRC-повідомлення може містити до трьох головних частин: префікс (опціонально), команду й параметри команди (яких може бути до 15). Префікс, команда й всі параметри розділені одним (або більше) символом пробілу (' ', 0x20).

Префікс позначається одним символом, що стоїть спочатку (':', 0x3b), який повинен бути першим символом у повідомленні. Між префіксом і двокрапкою не повинно бути ніяких пробілів. Префікс використовується серверами для позначення джерела появи повідомлення.

Якщо префікс повідомлення загублений, то за джерело повідомлення беруть з'єднання, з якого було отримане повідомлення. Клієнтам не треба користуватися префіксами при відсиланні повідомлення; якщо вони почнуть використовувати префікси, то прийматися будуть тільки правильні й тільки з зареєстрованих нікнеймів. Якщо вихідні ідентифікатори префіксів не буде знайдені в серверних базах даних, або якщо вони зареєстровані з різних лінків, то сервер буде ігнорувати повідомлення.

Команда повинна містити правильну IRC-команду або тризначне число, представлене в ASCII-тексті.

IRC-повідомлення завжди виглядають як рядок символів, що закінчується парою символів CR-LF (Carriage Return – Line Feed. Повернення каретки – переведення рядка) і довжиною рядка, що не перевищує 512 символів (у ці 512 входять і CR-LF). Так що, максимальна довжина рядка для команд і параметрів – 510 символів. Перенос рядка неможливий.

### **Формат повідомлення в 'псевдо' BNF**

Протокол повідомлень повинен бути витягнутий із суміжних потоків октетів Поточним рішенням стало визначення двох символів, CR і LS, як роздільників повідомлень. Ігнорування порожніх повідомлень, які





отримані сервером, віддаляються. У всіх інших випадках, нумерований відповідь просто звичайне повідомлення, що містить змінну. Змінна має вигляд тризначного номера, що незмінно краще, ніж рядок символів.

### **Встановлення з'єднання сервер-клієнт**

При з'єднанні з IRC-сервером, клієнту посилається MOTD (якщо є присутнім), а так само поточне число серверів/клієнтів (подібно команді LUSER). Так само сервер вимагає повідомлення до клієнта, що містить ім'я й версію, як і будь-які інші повідомлення-вступи.

Після поділу із цим, сервер повинен відправити назовні новий нікнейм користувача й іншої інформації, як постачання себе (команда USER) і як відкриваючий сервер (з DNS/сервери ідентифікації). Сервер повинен послати цю інформацію з NICK першим наступної USER.

### **Встановлення з'єднання сервер-сервер**

Процес установки з'єднання сервер-до-сервера чреватий небезпекою, так як багато можливих областей, де виникають проблеми – найменші з яких – конфліктні ситуації.

Після з'єднання сервера, слідує PASS/SERVER, якими при з'єднанні, сервера обмінюються парами PASS/SERVER, за допомогою яких встановлюються паролльні лінки. За допомогою цих же команд і їх відповідей, сервера обмінюються інформацією про з'єднання, що описана нижче.

При ініціалізації серверних запитів пари PASS/SERVER, так само перевіряється, що сервер належним чином відповідає на ідентифікацію перед підтвердженням з'єднання (перевіряється для того, щоб переконатися в тому, що це дійсно сервер).

### **Розрив з'єднання сервер-клієнт**

Коли клієнт закриває з'єднання, створюється QUIT-повідомлення від імені клієнта сервером, до якого приєднаний клієнт. Не використовується й не створюється ніяке інше повідомлення.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54



```

begin
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormat
dString (TAG_COLOR+'4Виберіть будь ласка сервер');
ActiveControl:=Irc_Server_Address_ComboBox;
exit;
end
else
if Irc_Nik_Edit.Text='' then
begin
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormat
dString (TAG_COLOR+'4Введіть будь ласка нік');
ActiveControl:=Irc_Nik_Edit;
exit;
end
else
if Irc_Password_Edit.Text='' then
begin
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormat
dString (TAG_COLOR+'4Введіть будь ласка пароль');
ActiveControl:=Irc_Password_Edit;
exit;
end
else
Irc_PageControl.ActivePageIndex:=0;
IdIRC.Replies.ClientInfo:=TAG_BOLD+Form1.Caption;
IdIRC.Replies.Version:=IdIRC.Replies.ClientInfo;
IdIRC.Replies.Finger:=IrcSettingsForm.IrcSettings_CTCP_Finger_Edit.Text;
IdIRC.Replies.UserInfo:=IrcSettingsForm.IrcSettings_CTCP_Userinfo_Edit.Text;
IdIRC.Nick:=Irc_Nik_Edit.Text;
IdIRC.AltNick:=IrcSettingsForm.IrcSettings_AltNick_Edit.Text;
IdIRC.Username:=IrcSettingsForm.IrcSettings_Username_Edit.Text;
IdIRC.RealName:=IrcSettingsForm.IrcSettings_Username_Edit.Text;
IdIRC.Password:=Irc_Password_Edit.Text;
IdIRC.Host:=Irc_Server_Address_ComboBox.Text;
IdIRC.Port:=IrcSettingsForm.IrcSettings_Port_SpinEdit.Value;
try
    IdIRC.Connect (500);
except
    on e:Exception do Reconnect (E.Message);
end;
end;
end;
end;

```

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>56</b>



```

    if
AnsiUpperCase (Irc_PageControl.Pages [i].Caption) =AnsiUpperCase (AChannel.Name)
then
    begin
        Irc_PageControl.Pages [i].TabVisible:=true;
        Irc_PageControl.ActivePage:=Irc_PageControl.Pages [i];
        Exit;
    end;
end;
// Створюємо вкладку для каналу
NewPage:=TTabSheet.Create (Self);
NewPage.PageControl:=Irc_PageControl; // Вставляємо закладку
NewPage.Name:='TabSheet'+IntToStr (TabSheet_Integer);
NewPage.Caption:=AChannel.Name;
NewPage.ImageIndex:=4;
// Процедура показу вкладки каналу
NewPage.OnShow:=Irc_Channel_Show;
// Вставляємо панель
NewPanel:=TPanel.Create (Self);
NewPanel.Parent:=NewPage;
NewPanel.Name:=NewPage.Name+'_Panel';
NewPanel.BevelOuter:=bvNone;
NewPanel.Caption:='';
NewPanel.Height:=21;
NewPanel.Align:=alBottom;
NewEdit:=TEdit.Create (Self); // Вставляємо Edit
NewEdit.Parent:=NewPanel;
NewEdit.Name:=NewPage.Name+'_Edit';
NewEdit.Top:=0;
NewEdit.Width:=Irc_Server_Edit.Width;
NewEdit.Left:=0;
NewEdit.Text:='';
NewEdit.MaxLength:=510;
// Процедура відправлення повідомлень на канал
NewEdit.OnKeyPress:=Irc_Channels_MessageText_EditKeyPress;
// Процедура вставки спецтегів
NewEdit.OnKeyDown:=Irc_Channels_EditKeyDown;
NewListBox:=TListBox.Create (Self); // Вставляємо ListBox
NewListBox.Parent:=NewPage;
NewListBox.Name:=NewPage.Name+'_ListBox';
NewListBox.Width:=120;
NewListBox.Align:=alRight;

```

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>58</b>

```

NewListBox.Sorted:=true;
NewListBox.ShowHint:=true;
NewListBox.PopupMenu:=Irc_Channel_ListBox_PopupMenu;
// Процедура кліку за списком користувачів на каналі
NewListBox.OnClick:=Irc_Channels_ListBoxClick;
// Процедура подвійного кліку за списком користувачів
NewListBox.OnDbClick:=Irc_Channels_ListBox_PopupMenu_PrivatClick;
// Процедура показу кількості користувачів на каналі
NewListBox.OnMouseMove:=Irc_Channel_ListBoxMouseMove;
// Перехід на Edit при відпусканні миші
NewListBox.OnMouseUp:=Irc_Server_IRCMemoMouseUp;
// Вставляємо Splitter
NewSplitter:=TSplitter.Create(Self); // Вставляємо Splitter
NewSplitter.Parent:=NewPage;
NewSplitter.Name:=NewPage.Name+'_Splitter';
NewSplitter.ShowHint:=true;
NewSplitter.Align:=alRight;
NewSplitter.Beveled:=true;
NewIRCMemo:=TIRCMemo.Create(Self); //вставляємо IRCMemo
NewIRCMemo.Parent:=NewPage;
NewIRCMemo.Name:=NewPage.Name+'_IRCMemo';
NewIRCMemo.Align:=alClient;
NewIRCMemo.ReadOnly:=true;
NewIRCMemo.Clear;
// Перехід на Edit при відпусканні миші
NewIRCMemo.OnMouseUp:=Irc_Server_IRCMemoMouseUp;
// Процедура показу властивостей каналу
NewIRCMemo.OnDbClick:=Irc_ChannelSettings_SpeedButtonClick;
// Смайли
NewIRCMemo.SmilesInfo[0].Smile:=':)';
NewIRCMemo.SmilesInfo[0].ImageIndex:=0;
NewIRCMemo.SmilesInfo[1].Smile:= '=)';
NewIRCMemo.SmilesInfo[1].ImageIndex:=0;
NewIRCMemo.SmilesInfo[2].Smile:=':-)';
NewIRCMemo.SmilesInfo[2].ImageIndex:=0;
NewIRCMemo.SmilesInfo[3].Smile:=':(';
NewIRCMemo.SmilesInfo[3].ImageIndex:=1;
NewIRCMemo.SmilesInfo[4].Smile:=':-(';
NewIRCMemo.SmilesInfo[4].ImageIndex:=1;
NewIRCMemo.SmilesInfo[5].Smile:=':D';
NewIRCMemo.SmilesInfo[5].ImageIndex:=2;
NewIRCMemo.SmilesInfo[6].Smile:=':-D';

```

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>59</b>

```

NewIRCMemo.SmilesInfo[6].ImageIndex:=2;
NewIRCMemo.SmilesInfo[7].Smile:=':P';
NewIRCMemo.SmilesInfo[7].ImageIndex:=3;
NewIRCMemo.SmilesInfo[8].Smile:=':-P';
NewIRCMemo.SmilesInfo[8].ImageIndex:=3;
NewIRCMemo.SmilesInfo[9].Smile:=';)';
NewIRCMemo.SmilesInfo[9].ImageIndex:=4;
NewIRCMemo.SmilesInfo[10].Smile:=';-)';
NewIRCMemo.SmilesInfo[10].ImageIndex:=4;
NewIRCMemo.SmilesInfo[11].Smile:='8)';
NewIRCMemo.SmilesInfo[11].ImageIndex:=5;
NewIRCMemo.SmilesInfo[12].Smile:=' 8-8-)';
NewIRCMemo.SmilesInfo[12].ImageIndex:=5;
NewIRCMemo.SmilesInfo[13].Smile:=':\';
NewIRCMemo.SmilesInfo[13].ImageIndex:=6;
NewIRCMemo.SmilesInfo[14].Smile:=':-\';
NewIRCMemo.SmilesInfo[14].ImageIndex:=6;
NewIRCMemo.SmilesInfo[15].Smile:='#39+' (';
NewIRCMemo.SmilesInfo[15].ImageIndex:=7;
NewIRCMemo.SmilesInfo[16].Smile:=':'+#39+' (';
NewIRCMemo.SmilesInfo[16].ImageIndex:=7;
NewIRCMemo.Smiles:=IRC_ImageList;
// Перемикаємося на нову вкладку
Irc_PageControl.ActivePage:=NewPage;
// Збільшуємо значення TabSheet_Integer на 1
TabSheet_Integer:=TabSheet_Integer+1;
// Робота з файлами лога
IRCLogs(AChannel.Name, '', 'Захід на канал');
end;

```

### Підпрограма надсилення приватних повідомлень:

```

// Процедура створення нової вкладки для привата
procedure TForm1.IrcNewPrivatPage(Caption: String; UserNick: String;
MessageText: String; PageImage: Integer);
var
    NewPage      : TTabSheet;
    NewPanel     : TPanel;
    NewEdit      : TEdit;
    NewIRCMemo   : TIRCMemo;
begin
    if Caption='' then exit;
    if Caption[1]='я' then Caption[1]:='Я';

```

```

// Створюємо вкладку для привату
NewPage:=TTabSheet.Create(Self);
NewPage.PageControl:=Irc_PageControl; // Вставляємо закладку
NewPage.Name:='TabSheet'+IntToStr(TabSheet_Integer);
NewPage.Caption:=Caption;
NewPage.ImageIndex:=PageImage;
// Процедура показу вкладки привату
NewPage.OnShow:=Irc_Privat_Show;
// Вставляємо панель
NewPanel:=TPanel.Create(Self);
NewPanel.Parent:=NewPage;
NewPanel.Name:=NewPage.Name+'_Panel';
NewPanel.BevelOuter:=bvNone;
NewPanel.Caption:='';
NewPanel.Height:=21;
NewPanel.Align:=alBottom;
NewEdit:=TEdit.Create(Self); // Вставляємо Edit
NewEdit.Parent:=NewPanel;
NewEdit.Name:=NewPage.Name+'_Edit';
NewEdit.Top:=0;
NewEdit.Width:=Irc_Server_Edit.Width;
NewEdit.Left:=0;
NewEdit.Text:='';
NewEdit.MaxLength:=510;
// Процедура відправлення повідомлень у приват
NewEdit.OnKeyPress:=Irc_Channels_MessageText_EditKeyPress;
// Процедура вставки спецтегів
NewEdit.OnKeyDown:=Irc_Channels_EditKeyDown;
NewIRCMemo:=TIRCMemo.Create(Self); //вставляємо IRCMemo
NewIRCMemo.Parent:=NewPage;
NewIRCMemo.Name:=NewPage.Name+'_IRCMemo';
NewIRCMemo.Align:=alClient;
NewIRCMemo.ReadOnly:=true;
NewIRCMemo.Clear;
// Перехід на Edit при відпусканні миші
NewIRCMemo.OnMouseUp:=Irc_Server_IRCMemoMouseUp;
// Смайли
NewIRCMemo.SmilesInfo[0].Smile:=' :) ';
NewIRCMemo.SmilesInfo[0].ImageIndex:=0;
NewIRCMemo.SmilesInfo[1].Smile:=' =) ';
NewIRCMemo.SmilesInfo[1].ImageIndex:=0;
NewIRCMemo.SmilesInfo[2].Smile:=' :-) ';

```

```

NewIRCMemo.SmilesInfo[2].ImageIndex:=0;
NewIRCMemo.SmilesInfo[3].Smile:=':(';
NewIRCMemo.SmilesInfo[3].ImageIndex:=1;
NewIRCMemo.SmilesInfo[4].Smile:=':-(';
NewIRCMemo.SmilesInfo[4].ImageIndex:=1;
NewIRCMemo.SmilesInfo[5].Smile:=':D';
NewIRCMemo.SmilesInfo[5].ImageIndex:=2;
NewIRCMemo.SmilesInfo[6].Smile:=':-D';
NewIRCMemo.SmilesInfo[6].ImageIndex:=2;
NewIRCMemo.SmilesInfo[7].Smile:=':P';
NewIRCMemo.SmilesInfo[7].ImageIndex:=3;
NewIRCMemo.SmilesInfo[8].Smile:=':-P';
NewIRCMemo.SmilesInfo[8].ImageIndex:=3;
NewIRCMemo.SmilesInfo[9].Smile:=':)';
NewIRCMemo.SmilesInfo[9].ImageIndex:=4;
NewIRCMemo.SmilesInfo[10].Smile:=':;-)';
NewIRCMemo.SmilesInfo[10].ImageIndex:=4;
NewIRCMemo.SmilesInfo[11].Smile:=':8)';
NewIRCMemo.SmilesInfo[11].ImageIndex:=5;
NewIRCMemo.SmilesInfo[12].Smile:=': 8-8-)';
NewIRCMemo.SmilesInfo[12].ImageIndex:=5;
NewIRCMemo.SmilesInfo[13].Smile:=':\';
NewIRCMemo.SmilesInfo[13].ImageIndex:=6;
NewIRCMemo.SmilesInfo[14].Smile:=':-\';
NewIRCMemo.SmilesInfo[14].ImageIndex:=6;
NewIRCMemo.SmilesInfo[15].Smile:=':#39+'(';
NewIRCMemo.SmilesInfo[15].ImageIndex:=7;
NewIRCMemo.SmilesInfo[16].Smile:=':'+#39+'(';
NewIRCMemo.SmilesInfo[16].ImageIndex:=7;
NewIRCMemo.Smiles:=IRC_ImageList;
// Збільшуємо значення TabSheet_Integer на 1
TabSheet_Integer:=TabSheet_Integer+1;
// Додаємо текст повідомлення
if (UserNick='') or (MessageText='') then
exit
else NewIRCMemo.AddFormattedString(TAG_COLOR+'1<'+UserNick+'>
'+MessageText);
end;

```

## 4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму FEAL – блоковий шифр, запропонований Акіхіро Симідзу і Седзі Міягуті.

У ньому використовуються 64-бітовий блок і 64-бітовий ключ. Його ідея полягає і в тому, щоб створити алгоритм, подібний DES, але з більш сильною функцією етапу. Використовуючи менше етапів, цей алгоритм міг би працювати швидше. На жаль, дійсність виявилася далекою від цілей проекту.

Як вхід процесу шифрування використовується 64-бітовий блок відкритого тексту. Спочатку блок даних підлягає операції XOR з 64 бітами ключа. Потім блок даних розщеплюється на ліву і праву половини. Об'єднання лівої і правої половин за допомогою XOR утворює нову праву половину. Ліва половина і нова права половина проходять через N етапів (спочатку 4). На кожному етапі половина об'єднується за допомогою функції F[1] з 16 бітами ключа і за допомогою XOR – з лівою половиною, створюючи нову праву половину. Вихідна права половина (на початок етапу) стає новою лівою половиною. Після N етапів (ліва і права половини не переставляти після N-го етапу) ліва половина знову об'єднується з допомогою XOR з правою половиною, утворюючи нову праву половину, потім ліва і права об'єднуються разом в 64-бітове ціле. Блок даних об'єднується за допомогою XOR з іншими 64 бітами ключа і алгоритм завершується.

					VKPM-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунках 5.1 та 5.2 зображене головне вікно програми під час підключення до серверу та в процесі чату відповідно.

Програма має простий та інтуїтивно зрозумілий інтерфейс. Для початку роботи слід здійснити авторизацію: ввести логін, пароль та назву серверу IRC-чату. Після цього в полі повідомлень чату з'явиться лог підключення та довідкова інформація про обраний сервер.

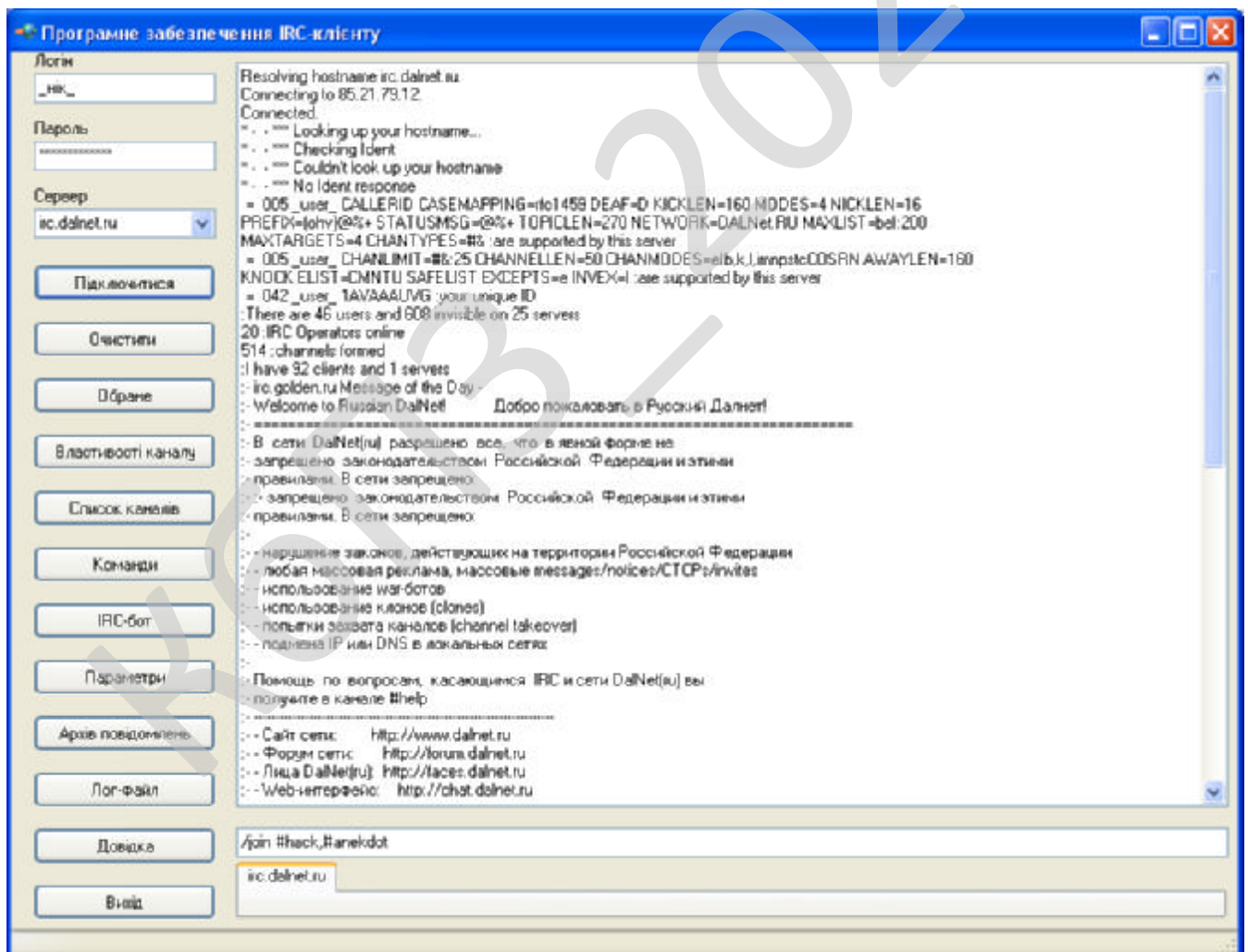


Рисунок 5.1 – Головне вікно програми (підключення до сервера)

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64



Інтерфейс програми надає доступ до наступних можливостей:

1. Вибір та підключення до IRC-сервера.
2. Перегляд списку наявних каналів та підключення до вибраного каналу.
3. Додавання підключених каналів в «Обране».
4. Перегляд властивостей каналу.
5. Очищення текстового поля повідомлень чату.
6. Перегляд списку команд та запитів, запуск вибраних команд, запитів.
7. Налаштування та використання IRC-бота.
8. Ведення та перегляд архіву повідомлень.
9. Ведення та перегляд лог-файлу.
10. Налаштування параметрів з'єднання та вибір режимів роботи IRC-чату.
11. Перегляд довідки.

**Формат текстових команд користувача IRC-клієнта:**

/join #ім'я\_каналу – зайти в канал,

/nick ваш\_новий\_нік – змінити нік,

/list – показати список каналів,

/msg нік\_користувача текст – відправити текстове повідомлення,

/me текст або наберіть текст і натисніть Ctrl+Enter – відправити Action на канал,

/notice нік\_користувача або #ім'я\_каналу і текст – відправити Notice,

/away текст – включити автовідповідь на приватні повідомлення

/away – виключити автовідповідь на приватні повідомлення,

/clear – очистити вікно показу повідомлень,

/help – показати довідку по IRC командам,

Ctrl+B – жирний текст,

Ctrl+U – підкреслений текст,

Ctrl+R – інвертований текст,

Ctrl+K+номер кольору – кольоровий текст,

Ctrl+K+номер кольору +,+номер кольору фона – кольоровий текст на

кольоровому фоні.

На рисунку 5.3 наведено вікно «Про програму...», що містить короткі відомості про розроблене програмне забезпечення.

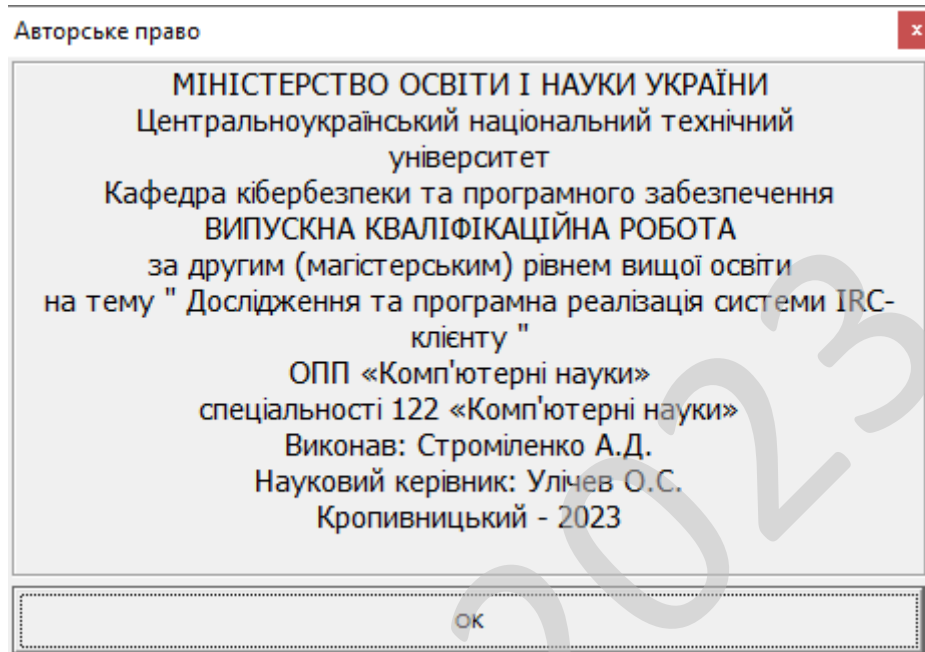


Рисунок 5.3 – Вікно довідки

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи IRC-клієнту.

*Метою розробки є дослідження та програмна реалізація системи IRC-клієнту.*

*Об'єктом дослідження є процес IRC-клієнту.*

*Предметом дослідження є методи IRC-клієнту.*

*Методи дослідження базуються на методах побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.*

**Наукова новизна отриманих результатів.** У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод IRC-клієнту.
- Розроблено вітчизняний продукт IRC-клієнту, який має більш широкі можливості, на відміну від існуючих аналогів.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

## 7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

### 7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі були проведені дослідження та виконана програмна реалізація системи IRC-клієнту.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	20
3. Запланований термін розробки, днів	Frq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	1
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	20000
33. Норматив додаткової зарплати, % :	Н <sub>д</sub>	10
34. Норматив відрахувань у соціальні фонди, %	Н <sub>с</sub>	22
35. Норматив загальногосподарських витрат, %	Н <sub>г</sub>	15
36. Норматив витрат на освоєння нових мов програмування, %	Н <sub>п</sub>	15
37. Рівень рентабельності програмної продукції, %	Р <sub>е</sub>	50
38. Ставка податку на додану вартість, %	Н <sub>дв</sub>	20

## 7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП як одна з найбільш тривалих і трудомістких робіт значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де:  $A$  – коефіцієнт Боема,  $A = 2,45$ ;

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де:  $W_i$  – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026.$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} PV_j, \quad (7.3)$$

де:  $PV_j$  – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1,1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33 + 0,2(B-1,01)} S, \quad (7.4)$$

де:  $C$  – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

$S$  – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 3,23 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 64 = 131 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	131	Ф 7.1-7.4
Впровадження	13	Д13
Всього	172	–

### 7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{nz} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де:  $F_{pq}$  – плановий фонд робочого часу одного спеціаліста, днів;

$T_{nz}$  – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{172 \cdot 1}{60 - 5} = 3,1 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	8	720	12
Монітор	60	8	480	8
Клавіатура	30	8	240	4
Маніпулятор «мишка»	30	8	240	4
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	2	60	1
Кабельні господарства ЛОМ на 1 м.п.	2,5	150	375	6,25
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 <sub>ч</sub>	40,91

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{\text{др}}^c = \frac{3_{\text{ч}} \cdot n_{\text{міс}}}{1,2}, \quad (7.6)$$

$$\Phi_{\text{др}}^c = \frac{41 \cdot 3}{1,2} = 102,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{\text{ел}} = \frac{\Phi_{\text{др}}^c}{F_{\text{др}} \cdot T_{\text{зм}}}, \quad (7.7)$$



Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	0,5	8000	12000
Продакт-менеджер	0,25	7000	5250
Інженер-програміст	3,1	8000	74400
Інженер-електронщик	0,2	7000	4200
Інженер-системотехнік	0,25	7000	5250
Адміністратор мережі	0,5	7000	10500
Системний програміст	0,25	7000	5250
Дизайнер WEB	0,25	7000	5250
Інженер-верстальник	0,25	7000	5250
Бухгалтер-економіст	0,5	7786	11679
Всього за період розробки	$R_{cn} = 6,05$	-	$\Phi_{роб} = 139029$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де:  $\Phi_{роб}$  – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{139029}{6,05 \cdot 60} = 383 \text{ грн.}$$

#### 7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

$$B_{yd} = R_{cn}^1 S_y \Pi_{nl}, \quad (7.9)$$

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

де:  $R_{cn}^1$  – кількість робочих місць виконавців, шт. Приймаємо 13 робочих місць;

$S_y$  – питома площа на одне робоче місце,  $m^2$ ;

$C_{пл}$  – вартість одного квадратного метра площі, грн.

Згідно даних інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./ $m^2$ . Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ $m^2$ . На кожне робоче місце у середньому потрібно  $8 m^2$ . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{нв} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де:  $C_m$  – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу Інтернет-магазину Компбест за 21.10.23 – джерело <https://compbest.com.ua>.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78



Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D ( 5ms, 300/3000:1, 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	12	10947	13136,4	144500,4
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	163051,9

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	163051,9	-	-
Всього по групі	163051,9	50	81525,95
Група 5, 6			
4. Вимірювальні пристрої	9031	25	2257,75
5. Транспортні засоби	143000	20	28600
6. Господарський інвентар	28000	25	7000
Всього по групі 5,6	180031	-	37857,75
Нематеріальні активи			
7. Нематеріальні активи	20000	10	2000
Разом	$K_p = 1771082,9$		$A_p = 189783,7$

Примітка: вартість автомобіля Renault Scenic 2008 взята за даними електронного ресурсу, джерело [https://auto.ria.com/uk/auto\\_renault\\_scenic\\_33598032.html](https://auto.ria.com/uk/auto_renault_scenic_33598032.html), складає 143000 грн.



Згідно прийнятих норм на підприємстві  $n_{\text{вип}}$  приймаємо 0,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає  $C_n=210$  грн., визначаємо вартість паперу за період розробки:

$$Z_{M1} = C_n \cdot N_m. \quad (7.16)$$

$$Z_{M1} = 210 \cdot 0,5 = 105 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 2):

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де:  $C_d$  – вартість дисків CD/DVD: CDR box – 21,5 грн./шт., DVD-R box – 39 грн./шт.

$$Z_{M2} = 21,5 \cdot 2 = 43 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де:  $C_z$  – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (105 + 43 + 1702) / 20 = 93 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ( $H_n = 15\%$ ) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де:  $H_n$  – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 3292 \cdot 15 \cdot 0,01 = 494 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ( $N_e = 20$  прим.):

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де:  $A_p$  – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 189783,7 \cdot 3 / (20 \cdot 12) = 2372 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 3292 + 392 + 797 + 494 + 93 + 494 + 2372 = 7934 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності ( $P_n$ ) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де:  $P_n$  – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 7934 = 3967 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1	2	3
1. Основна зарплата виконавців	$Z_o$	3292
2. Додаткова зарплата виконавців	$Z_d$	392
3. Відрахування на соціальні потреби	$C_{oc}$	797
4. Загальногосподарські витрати	$\Gamma_{ocn}$	494
5. Витрати на матеріали	$Z_m$	93
6. Освоєння нових операційних систем, мов програмування	$O_n$	494

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	$A_m$	2372
8. Повна собівартість програмного забезпечення	$C_{\pi}$	7934
9. Плановий прибуток	$P_p$	3967
10. Ціна підприємства $C_{\pi} = C_{\pi} + P_p$	$C_{\pi}$	11901
11. Податок на додану вартість $ПДВ = 0.01 \cdot N_{дв} \cdot C_{\pi}$	ПДВ	2380,2
12. Відпускна ціна програмної продукції $C = C_{\pi} + ПДВ$	$C$	14281,2

### 7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.9.

Таблиця 7.9 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	14281
Всього капітальних витрат	–	14281

## 7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування	$Z_p$	26840	12078
2. Витрати на електроенергію	$Z_{ел}$	2268	2016
3. Витрати на амортизацію	$Z_{ам}$	0	3570
Всього витрат за рік	I	29108	17664

Витрати на обслуговування системи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де:  $T_p$  – кількість годин обслуговування системи за рік, год.;

$Z_2$  – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 200 годин на рік до 90 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 200 \cdot 100 \cdot 1,1 \cdot 1,22 = 26840 \text{ грн.}$$

до:

$$Z_{p \text{ нов}} = 90 \cdot 100 \cdot 1,1 \cdot 1,22 = 12078 \text{ грн.}$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ( $P_{ел}$ ) в кіловатах, часу експлуатації технічних засобів ( $T_p$ ) в годинах та ціни однієї кіловат-години ( $C_{ел}$ ):

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

$$Z_{ел баз} = 0,45 \cdot 2400 \cdot 2,1 = 2268 \text{ грн.}$$

$$Z_{ел нов} = 0,40 \cdot 2400 \cdot 2,1 = 2016 \text{ грн.}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн., за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	14281	–	3570,25
Всього відрахувань	-	–	14281	–	3570,25

### 7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (C_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де:  $K_p$  – балансова вартість основних фондів розробника, грн.;  $E_p$  – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (11901 - 7934) \cdot 20 - (0,05 \cdot 1408000 + 0,5 \cdot 163051,9 + 0,2 \cdot 143000 + 0,25 \cdot 37031 + 0,1 \cdot 20000) \cdot 3/12 = 31394 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

де:  $K_p^*$  – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_{\sigma} = \frac{363083}{(11901 - 7934) \cdot 20 \cdot 12 / 3} = 1,1 \text{ року.}$$

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	20
2. Повна собівартість розробленої програми	Грн.	7934
3. Ціна розробленої програми	Грн.	11901
4. Плановий прибуток від реалізації розробленої програми	Грн.	3967
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1771083
7. Загальний прибуток від реалізації програмної продукції	Грн.	79340
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	31394
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Рік	1,1
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	14281
11. Величина економічного ефекту у користувача програмної продукції	Грн.	7874
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Рік	1,25

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\delta} - I_n) - E_n(K_n - K_{\delta}), \quad (7.27)$$

де:  $I_{\delta}$ ,  $I_n$  – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\delta}$ ,  $K_n$  – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (29108 - 17664) - 0,25 \cdot 14281 = 7874 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\delta}}{I_{\delta} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{14281}{29108 - 17664} = 1,25 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

## 7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості управлінських рішень.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Електронно-обчислювальна машина (ЕОМ) відіграє важливу роль у житті сучасної людини. Кожного дня мільйони людей використовують ЕОМ для пошуку необхідної інформації, спілкуванні у соціальних мережах, перегляду новин, роботи тощо. Багато людей користуються ЕОМ у професійних цілях, оскільки завдяки ЕОМ з'явилося багато нових професій. Тому для розробника хмарних сервісів так важливо розробити зручний інтерфейс для зручного сприйняття інформації, та необхідний функціонал, який буде відповідати необхідним вимогам та навантаженням. Все це вимагає багато багато часу та великого навантаження з боку розробників.

Тому так важливо слідкувати за умовами праці, в яких відбувається робочий процес. Оскільки захворювання можуть бути спричинені надмірним фізичним або розумовим навантаженням, через велику нервово-емоційну напругу, або через виробниче середовище. В даному розділі магістерської роботи проведемо аналіз основних чинників при роботі програміста.

### 8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

– ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.

– ризик виникнення пожежі;

– негативний вплив на органи зору людини;

– ризики ураження електричним струмом;

– недостатня, або надмірна освітленість робочого місця;

– електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);

– несприятливі мікрокліматичні умови;

– нервово-емоційна напруженість праці;

– інтелектуальні навантаження;

– монотонність праці;

– невідповідність ергономічних показників робочого місця діючим вимогам;

– шум;

– статичні навантаження на кістково-м'язовий апарат;

### 8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	6
Довжина	7
Висота	2,9

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого\*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м <sup>2</sup>	не менше 6.0	7
Об'єм, V	м <sup>3</sup>	не менше 20.0	20,3

\* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працюють 6 людей. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»). Тим чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Ia. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається

впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Воло- гість,%	Швидкість повітря, м/с	Температура, °С	Воло- гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-24	40-55	0,12
Тепла	23-25	50-70	0,1	24-25	50-65	0,1

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер Prinics PicKit M1 Smartphone Photo Printer White, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1], Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

#### **8.4. Розробка заходів з умов поліпшення охорони праці**

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язково наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

## 8.5 Розрахункова частина

Початкові дані для розрахунку захисного штучного заземлення: опір заземлювача, який нормується:  $R_{3H} = 4 \text{ Ом}$ .

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

Для захисного штучного заземлення застосовуються вертикальні електроди з металевого прутка діаметром 30 мм. ( $D=30 \text{ мм.}=0,03 \text{ м.}$ ) довжиною  $L=2,5 \text{ м.}$  та горизонтальний електрод — металева полоса з перетином  $40*4 \text{ мм.}$  тип ґрунта — глина (питомий опір  $40 \text{ Ом*м.}$ ). Відстань між вертикальними заземлювачами (електродами)  $A=3 \text{ м.}$

Глибина закладення горизонтального контура заземлення  $t=0,8 \text{ м.}$

Умовна товщина верхнього шару ґрунта:  $H=0,4 \text{ м.}$  Напруга — 220/380 В.

Розрахункова схема розташування заземлюючих електродів — у ряд.

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок захисного заземлення можна автоматизувати за допомогою програми, сирцевий код якої опублікован на стр. 13-16 [6], або аналогічної.

Розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,8+2,5/2=2,05 \text{ м.}$$

Розрахунковий питомий опір ґрунта (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунта):

$$\rho = \psi \rho_2 = 1,36 * 40 = 54,5 \text{ Ом*м.}$$

де  $\psi = 1,36$  — табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багатошаровому ґрунті [8];

$\rho_1 = 50 \text{ Ом*м.}$  — табличне значення питомого опору верхнього шару ґрунта [6];

$\rho_2 = 40 \text{ Ом*м.}$  — табличне значення питомого опору нижнього шару ґрунта [8].

Опір розтіканню електричного струму одного електрода вертикального заземлювача [8]:

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96



## Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

КБПЗ-2023

					VKPM-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи IRC-клієнту.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів IRC-клієнту.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем IRC-клієнту.
- Досліджена система IRC-клієнту.
- На основі отриманих результатів досліджень створена програмна реалізація системи IRC-клієнту.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання IRC-клієнту.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

Програма реалізована на мові високого рівня Delphi 10.4.1. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм FEAL.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 7874 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 1,25 роки.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Строміленко А.Д. Дослідження та програмна реалізація системи IRC-клієнту // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
3. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
4. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
5. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
6. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
7. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.
8. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.
9. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.
10. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and

cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

11. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

12. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

13. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

14. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

15. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

16. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

17. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019,

Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

18. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

19. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

20. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

21. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT-2019/ Lviv*, Ukraine, 2-6 July, 2019, P. 395-399.

22. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

23. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 347-352.

24. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special

Correlation Properties», *CEUR Workshop Proceedings* Volume 2353, *CEUR Workshop Proceedings* 2019, Pages 618-629.

25. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

26. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

27. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

28. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

29. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

30. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

					ВКРМ-122.23.0020.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

31. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

32. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.

33. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

34. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

35. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія*. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

36. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки*. № 2(33). с. 161-172, 2019.

37. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

38. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

39. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

40. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

41. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

42. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

43. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

44. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

45. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

46. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

47. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

48. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

49. Смірнов О.А., Стасєв Ю.В. Бараннік В.В. Захист інформації в автоматизованих системах управління. Навчальний посібник – Харків: ХУПС, 2015. – 264 с.

50. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.А. Смірнова Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 17.04.2012 року № 1/11-5249. – Кіровоград: КНТУ 2012. – 250 с.

51. Смірнов О.А., Євсєєв С.П., Жукарев В.Ю., Король О.Г., Сорокін В.Є., Мелешко Є.В. Технології і стандарти комп'ютерних мереж. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія» та 8.0925 «Автоматизація й комп'ютерно-інтегровані технології». За ред. О.А. Смірнова Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 1.12.2011 року № 1/11-11258. – Кіровоград: КНТУ 2012. – 454 с

					<b>ВКРМ-122.23.0020.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					<b>ВКРМ-122.23.0020.00.00.ТЗ</b>		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Строміленко А.Д.				Літ.	Аркуш	Аркушів
Перевірів	Улічев О.С.						
					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22М-1		
Затв.	Смірнов О.А.						

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи IRC-клієнту.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 32-13 від 04.08.2023 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи IRC-клієнту.

## 4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

					ВКРМ-122.23.0020.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи IRC-клієнту;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

## 5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

## 5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					<b>ВКРМ-122.23.0020.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

## 5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

## 5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

### 5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

### 5.8.2 Мова програмування

Середовище Delphi 10.4.1.

					ВКРМ-122.23.0020.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

## 7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз санітарно-гігієнічних умов праці на робочому місці програміста.

					ВКРМ-122.23.0020.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

## 9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 107 аркушів.

## 10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 11.12.2023 р.

					<b>ВКРМ-122.23.0020.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за  
другим (магістерським) рівнем вищої освіти

\_\_\_\_\_ Улічев О.С.

*Дослідження та програмна реалізація  
системи IRC-клієнту*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 84

Літера: РП

Кропивницький – 2023 року

```
// instruments.dpr - файл проекта
```

```
program instruments;

uses
  Forms,
  InstrumentsUnit in 'InstrumentsUnit.pas' {Form1},
  IrcSettingsUnit in 'IrcSettingsUnit.pas' {IrcSettingsForm},
  IrcCommandsUnit in 'IrcCommandsUnit.pas' {IrcCommandsForm},
  IrcColorUnit in 'IrcColorUnit.pas' {IrcColorForm},
  IrcFavoriteUnit in 'IrcFavoriteUnit.pas' {IrcFavoriteForm},
  IrcBotUnit in 'IrcBotUnit.pas' {IrcBotForm},
  IrcChannelSettingsUnit in 'IrcChannelSettingsUnit.pas'
  {IrcChannelSettingsForm},
  IrcDCCUnit in 'IrcDCCUnit.pas' {IrcDCCForm};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TIrcSettingsForm, IrcSettingsForm);
  Application.CreateForm(TIrcCommandsForm, IrcCommandsForm);
  Application.CreateForm(TIrcColorForm, IrcColorForm);
  Application.CreateForm(TIrcFavoriteForm, IrcFavoriteForm);
  Application.CreateForm(TIrcBotForm, IrcBotForm);
  Application.CreateForm(TIrcChannelSettingsForm, IrcChannelSettingsForm);
  Application.CreateForm(TIrcDCCForm, IrcDCCForm);
  Application.Run;
end.
```

```
// InstrumentsUnit.pas - модуль IRC-клиенту
```

```
unit InstrumentsUnit;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Forms, Variants, shellapi,  
CommCtrl, Menus, IdBaseComponent, IdComponent, IdTCPConnection,  
IdTCPClient, IdTelnet, Classes, ComCtrls, StdCtrls, ExtCtrls,  
Controls, ImgList,  
MMSystem, WWave32, IdIRC, Graphics, IniFiles, ExtActns,  
msxmldom, XMLDoc, IdAntiFreezeBase, IdAntiFreeze,  
Spin, Buttons, XPMAN, IRCMemo, MSNPopUp, WinInet, Speech, Active;
```

```
const
```

```
// IRC клиент  
IRC_OP=1;  
IRC_VOICE=2;
```

```
type
```

```
// IRC клиент  
TIRCUserState=(ircOp, ircVoice);  
TIRCUserStates=set of TIRCUserState;
```

```
TForm1 = class(TForm)
```

```
ImageList: TImageList;
```

```
Wave: TWave;
```

```
IdIRC: TIdIRC;
```

```
Irc_Channel_ListBox_PopupMenu: TPopupMenu;
```

```
Irc_Channels_ListBox_PopupMenu_Whois: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Privat: TMenuItem;
```

```
N6: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Control: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_CTCP: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_DCC: TMenuItem;
```

```
N8: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_CTCP_: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_CTCP_Time: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_CTCP_Version: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_DCC_Send: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_DCC_Chat: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Control_Op: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Control_DeOp: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Control_Voice: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Control_DeVoice: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Control_Kick: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_AddNick: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_CTCP_Finger: TMenuItem;
```

```
N4: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu: TPopupMenu;
```

```
Go_To_Channel: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_CTCP_Userinfo: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_CTCP_ClientInfo: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Action: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Action1: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Action_Ger11: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Action_Ger12: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Action4: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Action_Ger13: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Action_Ger14: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Action2: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_Action3: TMenuItem;
```

```
Irc_Channels_ListBox_PopupMenu_CTCP_Source: TMenuItem;
```

```
IdAntiFreeze: TIdAntiFreeze;
```

```
Irc_Channels_ListBox_PopupMenu_Control_Ban: TMenuItem;
```

```
Irc_Commands_PopupMenu: TPopupMenu;
```

```
Irc_Commands_PopupMenu_Join: TMenuItem;
```

```
Irc_Commands_PopupMenu_Part: TMenuItem;
Irc_Commands_PopupMenu_Msg: TMenuItem;
Irc_Commands_PopupMenu_Notice: TMenuItem;
Irc_Commands_PopupMenu_Whois: TMenuItem;
Irc_Commands_PopupMenu_Whowas: TMenuItem;
Irc_Commands_PopupMenu_Nick: TMenuItem;
Irc_Commands_PopupMenu_CTCP: TMenuItem;
Irc_Commands_PopupMenu_CTCP_: TMenuItem;
Irc_Commands_PopupMenu_CTCP_Time: TMenuItem;
Irc_Commands_PopupMenu_CTCP_Version: TMenuItem;
Irc_Commands_PopupMenu_CTCP_Finger: TMenuItem;
Irc_Commands_PopupMenu_CTCP_UserInfo: TMenuItem;
Irc_Commands_PopupMenu_CTCP_ClientInfo: TMenuItem;
Irc_Commands_PopupMenu_CTCP_Source: TMenuItem;
Irc_Commands_PopupMenu_CTCP_Sound: TMenuItem;
Irc_Commands_PopupMenu_Links: TMenuItem;
Irc_Commands_PopupMenu_Admin: TMenuItem;
Irc_Commands_PopupMenu_ChannelInfo: TMenuItem;
Irc_Commands_PopupMenu_Me: TMenuItem;
Irc_Channels_ListBox_PopupMenu_CTCP_Sound: TMenuItem;
Irc_Commands_PopupMenu_RegNick: TMenuItem;
Irc_Commands_PopupMenu_RegChannel: TMenuItem;
XPManifest: TXPManifest;
IRC_ImageList: TImageList;
MSNPoPUp: TMSNPoPUp;
Irc_Favorite_PopupMenu: TPopupMenu;
Irc_Favorite_PopupMenu_Del: TMenuItem;
Irc_Favorite_PopupMenu_: TMenuItem;
Irc_Favorite_PopupMenu_Add: TMenuItem;
Irc_Commands_PopupMenu_Away: TMenuItem;
Irc_Commands_PopupMenu_Away_On: TMenuItem;
Irc_Commands_PopupMenu_Away_Off: TMenuItem;
Irc_List_PopupMenu: TPopupMenu;
Irc_List_100_PopupMenu: TMenuItem;
Irc_List_50_PopupMenu: TMenuItem;
Irc_List_All_PopupMenu: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action6: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action_Gerl5: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action_Gerl6: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action_Gerl7: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action_Gerl8: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action_Gerl9: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action5: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action_Gerl: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action_Gerl10: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action_Gerl11: TMenuItem;
Irc_Ping_Timer: TTimer;
Irc_Channels_ListBox_PopupMenu_Action7: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Action_Gerl12: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Control_PoluOp: TMenuItem;
Irc_Channels_ListBox_PopupMenu_Control_DePoluOp: TMenuItem;
Inet_Timer: TTimer;
Irc_PageControl: TPageControl;
Irc_Server: TTabSheet;
Irc_Server_Panel: TPanel;
Irc_Server_Edit: TEdit;
Irc_Server_IRCMemo: TIRCMemo;
Channels: TTabSheet;
Channels_ListBox: TListBox;
Channels_IRCMemo: TIRCMemo;
Irc_Top_Panel: TPanel;
Irc_Close_SpeedButton: TSpeedButton;
Irc_Faq_SpeedButton: TSpeedButton;
Irc_Settings_SpeedButton: TSpeedButton;
Irc_Channels_SpeedButton: TSpeedButton;
Irc_Commands_SpeedButton: TSpeedButton;
Irc_Clear_SpeedButton: TSpeedButton;
Irc_Favorite_SpeedButton: TSpeedButton;
Irc_Log_SpeedButton: TSpeedButton;
```

```

Irc_Bot_SpeedButton: TSpeedButton;
Irc_Away_Status_Label: TLabel;
Irc_ChannelSettings_SpeedButton: TSpeedButton;
Irc_Nik_GroupBox: TGroupBox;
Irc_Nik_Edit: TEdit;
Irc_Password_GroupBox: TGroupBox;
Irc_Password_Edit: TEdit;
Irc_Server_Address_GroupBox: TGroupBox;
Irc_Server_Address_ComboBox: TComboBox;
Irc_Connected_Button: TButton;
StatusBar: TStatusBar;
Speak_GroupBox: TGroupBox;
Speak_ComboBox: TComboBox;

procedure FormCreate(Sender: TObject);
procedure Irc_Connected_ButtonClick(Sender: TObject);
procedure IdIRCConnected(Sender: TObject);
procedure IdIRCDisconnected(Sender: TObject);
procedure IdIRCJoin(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel);
procedure IdIRCPart(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel);
procedure IdIRCRaw(Sender: TObject; AUser: TIdIRCUser; ACommand,
  AContent: String; var Suppress: Boolean);
procedure Irc_Channels_MessageText_EditKeyPress(Sender: TObject;
  var Key: Char);
procedure IdIRCKicked(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel);
procedure IdIRCKick(Sender: TObject; AUser, AVictim: TIdIRCUser;
  AChannel: TIdIRCChannel);
procedure IdIRCJoined(Sender: TObject; AChannel: TIdIRCChannel);
procedure Irc_Close_SpeedButtonClick(Sender: TObject);
procedure Irc_ServerShow(Sender: TObject);
procedure Irc_Channel_Show(Sender: TObject);
procedure Irc_Privat_Show(Sender: TObject);
procedure ChannelsShow(Sender: TObject);
procedure IdIRCQuit(Sender: TObject; AUser: TIdIRCUser);
procedure IdIRCStatus(ASender: TObject; const AStatus: TIdStatus;
  const AStatusText: String);
procedure IdIRCSocketError(Sender: TObject; ASocket, AMsg: String);
procedure IdIRCError(Sender: TObject; AUser: TIdIRCUser; ANumeric,
  AError: String);
procedure IdIRCAction(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel; Content: String);
procedure IdIRCMessage(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel; Content: String);
procedure IdIRCNames(Sender: TObject; AUsers: TIdIRCUsers;
  AChannel: TIdIRCChannel);
procedure IdIRCPingPong(Sender: TObject);
procedure IdIRCCTCPQuery(Sender: TObject; User: TIdIRCUser;
  AChannel: TIdIRCChannel; Command, Args: String;
  var ASuppress: Boolean);
procedure IdIRCCTCPReply(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel; Command, Args: String);
procedure IdIRCNotice(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel; Content: String);
procedure IdIRCTopic(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel; const AChanName, ATopic: String);
procedure IdIRCNickChange(Sender: TObject; AUser: TIdIRCUser;
  ANewNick: String);
procedure IdIRCNickChanged(Sender: TObject; AOldNick: String);
procedure IdIRCNicksInUse(Sender: TObject; var ANick: String);
procedure Irc_Settings_SpeedButtonClick(Sender: TObject);
procedure Irc_Channels_EditKeyDown(Sender: TObject;
  var Key: Word; Shift: TShiftState);
procedure Irc_Channels_ListBoxClick(Sender: TObject);
procedure IdIRCSysop(Sender: TObject; AUser: TIdIRCUser;
  ACmdCode: Integer; ACommand, AContent: String);

```

```

procedure IdIRCUserMode(Sender: TObject; AModes: String);
procedure IdIRCReceive(Sender: TObject; ACommand: String);
procedure Irc_Channels_ListBox_PopupMenu_AddNickClick(Sender: TObject);
procedure Irc_Channels_ListBox_PopupMenu_Control_KickClick(Sender: TObject);
procedure Irc_Channels_ListBox_PopupMenu_Control_OpClick(
  Sender: TObject);
procedure Irc_Channels_ListBox_PopupMenu_Control_DeOpClick(Sender: TObject);
procedure Irc_Channels_ListBox_PopupMenu_Control_VoiceClick(Sender:
TObject);
  procedure Irc_Channels_ListBox_PopupMenu_Control_DeVoiceClick(Sender:
TObject);
  procedure IdIRCUnknownCommand(Sender: TObject; AUser: TIdIRCUser;
    ACommand, AContent: String);
  procedure IdIRCSend(Sender: TObject; ACommand: String);
  procedure IdIRCChannelMode(Sender: TObject; AUser: TIdIRCUser;
    AChannel: TIdIRCChannel; AChanName, AModes: String);
  procedure Irc_Channels_ListBox_PopupMenu_WhoisClick(Sender: TObject);
  procedure Irc_Channels_ListBox_PopupMenu_CTCP_Click(
    Sender: TObject);
  procedure Go_To_ChannelClick(Sender: TObject);
  procedure Irc_Channels_ListBox_PopupMenu_PrivatClick(Sender: TObject);
  procedure Irc_Faq_SpeedButtonClick(Sender: TObject);
  procedure IdIRCInvite(Sender: TObject; AUser: TIdIRCUser;
    AChannel: String);
  procedure FormDestroy(Sender: TObject);
  procedure IdIRCDCCAccept(Sender: TObject; ANick, APort, AFileName,
    APosition: String);
  procedure IdIRCDCCChat(Sender: TObject; ANick, AIP, APort: String);
  procedure IdIRCDCCResume(Sender: TObject; ANick, APort, AFileName,
    APosition: String);
  procedure IdIRCDCCSend(Sender: TObject; ANick, AIP, APort, AFileName,
    AFileSize: String);
  procedure Irc_Channels_ListBox_PopupMenu_Action_Click(Sender: TObject);
  procedure Irc_Clear_SpeedButtonClick(Sender: TObject);
  procedure Channels_ListBoxClick(Sender: TObject);
  procedure Irc_Channels_ListBox_PopupMenu_Control_BanClick(
    Sender: TObject);
  procedure Irc_Commands_SpeedButtonClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_JoinClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_PartClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_MsgClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_NoticeClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_MeClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_WhoisClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_WhowasClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_NickClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_CTCP_Click(Sender: TObject);
  procedure Irc_Commands_PopupMenu_LinksClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_AdminClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_ChannelInfoClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_RegNickClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_RegChannelClick(Sender: TObject);
  procedure Irc_Favorite_SpeedButtonClick(Sender: TObject);
  procedure Irc_Favorite_PopupMenu_AddClick(Sender: TObject);
  procedure Irc_Favorite_PopupMenu_DelClick(Sender: TObject);
  procedure IdIRCUnAway(Sender: TObject; AContent: String);
  procedure IdIRCNowAway(Sender: TObject; AContent: String);
  procedure Irc_Commands_PopupMenu_Away_OnClick(Sender: TObject);
  procedure Irc_Commands_PopupMenu_Away_OffClick(Sender: TObject);
  procedure IrcNewPrivatPage(Caption: String; UserNick: String; MessageText:
String; PageImage: Integer);
  procedure Irc_List_100_PopupMenuClick(Sender: TObject);
  procedure Irc_List_50_PopupMenuClick(Sender: TObject);
  procedure Irc_List_All_PopupMenuClick(Sender: TObject);
  procedure Irc_Channels_SpeedButtonClick(Sender: TObject);
  procedure Irc_Ping_TimerTimer(Sender: TObject);
  procedure Irc_Log_SpeedButtonClick(Sender: TObject);
  procedure Reconnect(Err: String);
  procedure IdIRCParted(Sender: TObject; AChannel: TIdIRCChannel);

```

```

procedure IRCLogs(filename: String; User: String; Content: String);
procedure IrcSpeak(User: String; Text: String; Channel: Boolean);
procedure IrcFindText(User: String; Text: String; Channel: String);
procedure Irc_Channel_ListBoxMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
procedure Irc_Bot_SpeedButtonClick(Sender: TObject);
procedure IrcBotKick(User: String; Text: String; Channel: String);
procedure IrcBotOp(User: String; Channel: String);
procedure IrcBotVoice(User: String; Channel: String);
procedure IrcBotCommands(User: String; Text: String; Channel: String);
procedure Irc_ChannelSettings_SpeedButtonClick(Sender: TObject);
procedure IrcColor(ComboBox: TComboBox; Edit: TEdit; var Key: Word; Shift:
TShiftState);
procedure Irc_Channels_ListBox_PopupMenu_Control_PoluOpClick(
  Sender: TObject);
procedure Irc_Channels_ListBox_PopupMenu_Control_DePoluOpClick(
  Sender: TObject);
procedure Irc_Server_IRCMemoMouseUp(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure Channels_ListBoxMeasureItem(Control: TWinControl;
  Index: Integer; var Height: Integer);
procedure Channels_ListBoxDrawItem(Control: TWinControl;
  Index: Integer; Rect: TRect; State: TOwnerDrawState);
procedure Inet_TimerTimer(Sender: TObject);
procedure FormResize(Sender: TObject);
procedure Speak_ComboBoxChange(Sender: TObject);

private
  {Центральний інтерфейс, через який }
  {здійснюються всі дії з мовою}
  fITTSCentral      : ITTSCentral;
  {Інтерфейс для зв'язку з аудіоупристроями}
  fIAMM             : IAudioMultimediaDevice;
  {Інтерфейс для перебору движків}
  aTTSEnum         : ITTSEnum;
  {Показчик на параметри двіжка}
  fpModeInfo       : PTTSMODEInfo;

  // Клік по меню обране в IRC
  procedure ItemClick(sender:Tobject);
  // Кольоровий список каналів
  function CheckItemSize(Index: Integer): Integer;

protected

public
  { Public declarations }
end;

//////////
const
  // Кольоровий список каналів
  Кольори: Array [0 .. 15] TColor = ( / / MIRC кольори
  $ 00FFFFFF, / / Білий
  $ 00000000 / / Чорний
  $ 007F0000, / / Синій
  $ 00009300 / / Зелений
  $ 000000ff / / Червоний
  $ 0000007F / / Коритневий
  $ 009C009C / / Бузковий
  $ 00007FFC / / Помаранчевий
  $ 0000FFFF, / / Жовтий
  $ 0000FC00 / / Світло-зелений
  $ 00939300 / / Синьо-зелений
  $ 00FFFF00 / / Блакитний
  $ 00FC0000 / / Фіолетовий
  $ 00FF00FF / / Рожевий

```

```

    $ 007F7F7F / / Темно-сірий
    $ 00D2D2D2 / / світло-сірий
  );
var
  Form1          : TForm1;
  // Привласнюємо початкове значення додатковій цифрі до імені вкладки каналу
  й привата
  TabSheet_Integer : Integer=10;
  // Привласнюємо початкове значення кількості перепідключень
  Reconnect_Integer : Integer=0;
  away_message      : String;

```

```
implementation
```

```
uses
```

```

  IrcSettingsUnit, IrcCommandsUnit, IrcColorUnit, IrcFavoriteUnit,
  IrcBotUnit, IrcChannelSettingsUnit, IrcDCCUnit;

```

```

{$R *.dfm}
{$R sound.res}
{$R irc.res}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// IRC

```

```

// Функція видалення IRC тегів
function ParseColorString(s:string):string;
var

```

```

  i, j, m, n, o : integer;
begin
  Result:='';
  if s='' then exit;
  i:=0;
  while i<=Length(s) do
  begin
    inc(i);
    if i>1 then
      if ord(s[ i-1])<=31 then dec(i);
//жирний
    if s[i]=#2 then Delete(s,i,1);
//підкреслений
    if s[i]=#31 then Delete(s,i,1);
//інвернований
    if s[i]=#22 then Delete(s,i,1);
//рівний
    if s[i]=#15 then Delete(s,i,1);
//кольори
    if s[i]=#3 then
      begin
        Delete(s,i,1);

        try
          j:=StrToInt(s[i]);
        except
          j:=-1;
        end;

        try
          m:=StrToInt(s[i+1]);
        except
          m:=-1;
        end;

```

```

if ((m=-1) and (s[i+1]=',')) or ((m>-1) and (s[i+2]=',')) then //if
begin

    try
    if m=-1 then
        n:=StrToInt(s[i+2])
    else
        n:=StrToInt(s[i+3]);
    except
    n:=-1;
    end;

    try
    if m=-1 then
        o:=StrToInt(s[i+3])
    else
        o:=StrToInt(s[i+4]);
    except
    o:=-1;
    end;

end //if

else

begin
n:=-1;o:=-1;
end; //if
if (j=-1) then Continue;
if ((j>-1) and (m>-1)) then Delete(s,i,2);
if ((j>-1) and (m=-1)) then Delete(s,i,1);
if ((n>-1) and (o>-1)) then Delete(s,i,3);
if ((n>-1) and (o=-1)) then Delete(s,i,2);
end;

end;

Result:=s;
end;

// Проказування повідомлень активного каналу й приватів
procedure TForm1.IrcSpeak(User: String; Text: String; Channel: Boolean);
var
    SData : TSDData;
    BufRich : String;
begin
    try
        if Channel=true then BufRich:=User+' говорить '+Text else BufRich:=User+'
говорить у приват '+Text;
        SData.dwSize := length(BufRich) + 1;
        SData.pData := pChar(BufRich);
        fITTSentral.TextData(CHARSET_TEXT, 1, SData, nil, IID_ITTSBufNotifySink);
    except
        end;
end;

// Вивід повідомлення при згадуванні слів зі списку
procedure TForm1.IrcFindText(User: String; Text: String; Channel: String);
var
    i : Integer;
begin
begin
if IrcSettingsForm.IrcSettings_Show_Find_Messages_CheckBox.Checked then
begin
for i:=0 to IrcSettingsForm.IrcSettings_Show_Find_Messages_Memo.Lines.Count-
1 do
begin
while
IrcSettingsForm.IrcSettings_Show_Find_Messages_Memo.Lines[i][Length(IrcSettingsF
orm.IrcSettings_Show_Find_Messages_Memo.Lines[i])]=' ' do

```

```

IrcSettingsForm.IrcSettings_Show_Find_Messages_Memo.Lines[i]:=Copy(IrcSettingsFo
rm.IrcSettings_Show_Find_Messages_Memo.Lines[i],1,Length(IrcSettingsForm.IrcSett
ings_Show_Find_Messages_Memo.Lines[i])-1);
    if
StrPos (PChar (AnsiUpperCase (ParseColorString (Text))), PChar (AnsiUpperCase (IrcSetti
ngsForm.IrcSettings_Show_Find_Messages_Memo.Lines[i])))<>nil then
    begin
        MSNPopUp.Title:='IRC '+Channel+' '+User;
        MSNPopUp.Text:=ParseColorString (Text);
        MSNPopUp.GradientColor1:=clInfoBk;
        MSNPopUp.ShowPopUp;
    end;
end;
end;
end;

// Банити користувачів за використання слів зі списку
procedure TForm1.IrcBotKick(User: String; Text: String; Channel: String);
var
    i : Integer;
begin
    if IrcBotForm.IrcBot_CheckBox.Checked then
    begin
        if IrcBotForm.IrcBot_Kick_CheckBox.Checked then
        begin
            for i:=0 to IrcBotForm.IrcBot_Kick_Memo.Lines.Count-1 do
            begin
                while
IrcBotForm.IrcBot_Kick_Memo.Lines[i][Length(IrcBotForm.IrcBot_Kick_Memo.Lines[i]
)]=' ' do

IrcBotForm.IrcBot_Kick_Memo.Lines[i]:=Copy(IrcBotForm.IrcBot_Kick_Memo.Lines[i],
1,Length(IrcBotForm.IrcBot_Kick_Memo.Lines[i])-1);
                if
StrPos (PChar (AnsiUpperCase (ParseColorString (Text))), PChar (AnsiUpperCase (IrcBotFo
rm.IrcBot_Kick_Memo.Lines[i])))<>nil then
                begin
                    // Для всіх каналів
                    if IrcBotForm.IrcBot_Channels_RadioButton.Checked then
                    begin
                        try
                            IdIRC.WriteLine('KICK '+Channel+' '+User+'
:'+IrcBotForm.IrcBot_Kick_Edit.Text);
                        except
                            on e:Exception do Reconnect (E.Message);
                        end;
                    end;
                    // Для одного каналу
                    if IrcBotForm.IrcBot_Channel_RadioButton.Checked then
                    begin
                        if
AnsiUpperCase (Channel)=AnsiUpperCase (IrcBotForm.IrcBot_Channel_Edit.Text) then
                        begin
                            try
                                IdIRC.WriteLine('KICK '+Channel+' '+User+'
:'+IrcBotForm.IrcBot_Kick_Edit.Text);
                            except
                                on e:Exception do Reconnect (E.Message);
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

// Давати +o користувачам зі списку

```

```

procedure TForm1.IrcBotOp(User: String; Channel: String);
var
  i : Integer;
begin
  if IrcBotForm.IrcBot_CheckBox.Checked then
    begin
      if IrcBotForm.IrcBot_Op_CheckBox.Checked then
        begin
          for i:=0 to IrcBotForm.IrcBot_Op_Memo.Lines.Count-1 do
            begin
              while
                IrcBotForm.IrcBot_Op_Memo.Lines[i][Length(IrcBotForm.IrcBot_Op_Memo.Lines[i])]='
                ' do

                IrcBotForm.IrcBot_Op_Memo.Lines[i]:=Copy(IrcBotForm.IrcBot_Op_Memo.Lines[i],1,Le
                ngth(IrcBotForm.IrcBot_Op_Memo.Lines[i])-1);
                if
                StrPos(PChar(AnsiUpperCase(User)),PChar(AnsiUpperCase(IrcBotForm.IrcBot_Op_Memo.
                Lines[i])))<>nil then
                  begin
                    // Для всіх каналів
                    if IrcBotForm.IrcBot_Channels_RadioButton.Checked then
                      begin
                        try
                          IdIRC.WriteLine('MODE '+Channel+' +o '+User);
                        except
                          on e:Exception do Reconnect(E.Message);
                        end;
                      end;
                    // Для одного каналу
                    if IrcBotForm.IrcBot_Channel_RadioButton.Checked then
                      begin
                        if
                        AnsiUpperCase(Channel)=AnsiUpperCase(IrcBotForm.IrcBot_Channel_Edit.Text) then
                          begin
                            try
                              IdIRC.WriteLine('MODE '+Channel+' +o '+User);
                            except
                              on e:Exception do Reconnect(E.Message);
                            end;
                          end;
                        end;
                      end;
                    end;
                  end;
                end;
              end;
            end;
          end;
        end;
      // Давати +v користувачам зі списку
    procedure TForm1.IrcBotVoice(User: String; Channel: String);
    var
      i : Integer;
    begin
      if IrcBotForm.IrcBot_CheckBox.Checked then
        begin
          if IrcBotForm.IrcBot_Voice_CheckBox.Checked then
            begin
              for i:=0 to IrcBotForm.IrcBot_Voice_Memo.Lines.Count-1 do
                begin
                  while
                    IrcBotForm.IrcBot_Voice_Memo.Lines[i][Length(IrcBotForm.IrcBot_Voice_Memo.Lines[
                    i])]=' ' do

                    IrcBotForm.IrcBot_Voice_Memo.Lines[i]:=Copy(IrcBotForm.IrcBot_Voice_Memo.Lines[i
                    ],1,Length(IrcBotForm.IrcBot_Voice_Memo.Lines[i])-1);
                    if
                    StrPos(PChar(AnsiUpperCase(User)),PChar(AnsiUpperCase(IrcBotForm.IrcBot_Voice_Me
                    mo.Lines[i])))<>nil then
                      begin

```



```

        end;
    end;
end;
// Для одного каналу
if IrcBotForm.IrcBot_Channel_RadioButton.Checked then
begin
    if
AnsiUpperCase(Channel)=AnsiUpperCase(IrcBotForm.IrcBot_Channel_Edit.Text) then
        begin
            if
StrPos(PChar(AnsiUpperCase(ParseColorString(Text))),PChar(AnsiUpperCase(IrcBotFo
rm.IrcBot_Commands_1_Edit.Text)))<>nil then
                begin
                    for i:=0 to Irc_PageControl.PageCount-1 do
                    begin
                        if Irc_PageControl.Pages[i].Caption=Channel then
                            begin
                                try
                                    IdIRC.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+'Користувачів на каналі '+Channel+'
'+IntToStr(TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Cou
nt));
                                except
                                    on e:Exception do Reconnect(E.Message);
                                end;

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormatedStr
ing(TAG_COLOR+'1<'+IdIRC.Nick+'> '+TAG_BOLD+'Користувачів на каналі '+Channel+'
'+IntToStr(TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Cou
nt));

                                // Робота з файлами лога
                                IRCLogs(Channel,IdIRC.Nick,'Користувачів на каналі
'+Channel+'
'+IntToStr(TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Cou
nt));
                            end;
                        end;
                    end;
                end;
            end;
        end;
end;
// Дати Voice користувачеві
if IrcBotForm.IrcBot_Commands_2_CheckBox.Checked then
begin
    // Для всіх каналів
    if IrcBotForm.IrcBot_Channels_RadioButton.Checked then
        begin
            if
StrPos(PChar(AnsiUpperCase(ParseColorString(Text))),PChar(AnsiUpperCase(IrcBotFo
rm.IrcBot_Commands_2_Edit.Text)))<>nil then
                begin
                    try
                        IdIRC.WriteLine('MODE '+Channel+' +v '+User);
                    except
                        on e:Exception do Reconnect(E.Message);
                    end;
                end;
            end;
        end;
    // Для одного каналу
    if IrcBotForm.IrcBot_Channel_RadioButton.Checked then
        begin
            if
AnsiUpperCase(Channel)=AnsiUpperCase(IrcBotForm.IrcBot_Channel_Edit.Text) then
                begin
                    if
StrPos(PChar(AnsiUpperCase(ParseColorString(Text))),PChar(AnsiUpperCase(IrcBotFo
rm.IrcBot_Commands_2_Edit.Text)))<>nil then
                        begin

```

```

        try
            IdIRC.WriteLine('MODE '+Channel+' +v '+User);
        except
            on e:Exception do Reconnect(E.Message);
        end;
    end;
end;
end;
end;

// Забрати Voice у користувача
if IrcBotForm.IrcBot_Commands_3_CheckBox.Checked then
begin
    // Для всіх каналів
    if IrcBotForm.IrcBot_Channels_RadioButton.Checked then
    begin
        if
StrPos (PChar (AnsiUpperCase (ParseColorString (Text))), PChar (AnsiUpperCase (IrcBotFo
rm.IrcBot_Commands_3_Edit.Text))) <> nil then
        begin
            try
                IdIRC.WriteLine('MODE '+Channel+' -v '+User);
            except
                on e:Exception do Reconnect(E.Message);
            end;
        end;
    end;
    // Для одного каналу
    if IrcBotForm.IrcBot_Channel_RadioButton.Checked then
    begin
        if
AnsiUpperCase (Channel) = AnsiUpperCase (IrcBotForm.IrcBot_Channel_Edit.Text) then
        begin
            if
StrPos (PChar (AnsiUpperCase (ParseColorString (Text))), PChar (AnsiUpperCase (IrcBotFo
rm.IrcBot_Commands_3_Edit.Text))) <> nil then
            begin
                try
                    IdIRC.WriteLine('MODE '+Channel+' -v '+User);
                except
                    on e:Exception do Reconnect(E.Message);
                end;
            end;
        end;
    end;
end;

// Довідка по командам
if IrcBotForm.IrcBot_Commands_4_CheckBox.Checked then
begin
    // Для всіх каналів
    if IrcBotForm.IrcBot_Channels_RadioButton.Checked then
    begin
        if
StrPos (PChar (AnsiUpperCase (ParseColorString (Text))), PChar (AnsiUpperCase (IrcBotFo
rm.IrcBot_Commands_4_Edit.Text))) <> nil then
        begin
            if IrcBotForm.IrcBot_Commands_1_CheckBox.Checked then
            begin
                for i:=0 to Irc_PageControl.PageCount-1 do
                begin
                    if Irc_PageControl.Pages[i].Caption=Channel then
                    begin
                        try
                            IdIRC.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+IrcBotForm.IrcBot_Commands_1_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_1_CheckBox.Caption);
                        except

```

```

        on e:Exception do Reconnect(E.Message);
    end;

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedString (TAG_COLOR+'1<'+IdIRC.Nick+'>
'+TAG_BOLD+IrcBotForm.IrcBot_Commands_1_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_1_CheckBox.Caption);
    // Работа з файлами лога

IRCLogs (Channel, IdIRC.Nick, IrcBotForm.IrcBot_Commands_1_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_1_CheckBox.Caption);
    IdAntiFreeze.Sleep(1000);
end;
end;

if IrcBotForm.IrcBot_Commands_2_CheckBox.Checked then
begin
    for i:=0 to Irc_PageControl.PageCount-1 do
    begin
        if Irc_PageControl.Pages[i].Caption=Channel then
        begin
            try
                IdIRC.WriteLine ('PRIVMSG '+Channel+'
:'+TAG_BOLD+IrcBotForm.IrcBot_Commands_2_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_2_CheckBox.Caption);
            except
                on e:Exception do Reconnect (E.Message);
            end;
        end;

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedString (TAG_COLOR+'1<'+IdIRC.Nick+'>
'+TAG_BOLD+IrcBotForm.IrcBot_Commands_2_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_2_CheckBox.Caption);
    // Работа з файлами лога

IRCLogs (Channel, IdIRC.Nick, IrcBotForm.IrcBot_Commands_2_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_2_CheckBox.Caption);
    IdAntiFreeze.Sleep(1000);
end;
end;

if IrcBotForm.IrcBot_Commands_3_CheckBox.Checked then
begin
    for i:=0 to Irc_PageControl.PageCount-1 do
    begin
        if Irc_PageControl.Pages[i].Caption=Channel then
        begin
            try
                IdIRC.WriteLine ('PRIVMSG '+Channel+'
:'+TAG_BOLD+IrcBotForm.IrcBot_Commands_3_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_3_CheckBox.Caption);
            except
                on e:Exception do Reconnect (E.Message);
            end;
        end;

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedString (TAG_COLOR+'1<'+IdIRC.Nick+'>
'+TAG_BOLD+IrcBotForm.IrcBot_Commands_3_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_3_CheckBox.Caption);
    // Работа з файлами лога

IRCLogs (Channel, IdIRC.Nick, IrcBotForm.IrcBot_Commands_3_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_3_CheckBox.Caption);
    IdAntiFreeze.Sleep(1000);
end;
end;
end;
end;

```

```

if IrcBotForm.IrcBot_Commands_4_CheckBox.Checked then
begin
  for i:=0 to Irc_PageControl.PageCount-1 do
  begin
    begin
      if Irc_PageControl.Pages[i].Caption=Channel then
      begin
        try
          IdIRC.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+IrcBotForm.IrcBot_Commands_4_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_4_CheckBox.Caption);
        except
          on e:Exception do Reconnect(E.Message);
        end;

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'1<'+IdIRC.Nick+'>
'+TAG_BOLD+IrcBotForm.IrcBot_Commands_4_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_4_CheckBox.Caption);
        // Работа з файлами лога

IRCLogs (Channel, IdIRC.Nick, IrcBotForm.IrcBot_Commands_4_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_4_CheckBox.Caption);
        IdAntiFreeze.Sleep(1000);
      end;
    end;
  end;
end;

// Для одного каналу
if IrcBotForm.IrcBot_Channel_RadioButton.Checked then
begin
  if
AnsiUpperCase (Channel)=AnsiUpperCase (IrcBotForm.IrcBot_Channel_Edit.Text) then
  begin
    if
StrPos (PChar (AnsiUpperCase (ParseColorString (Text))), PChar (AnsiUpperCase (IrcBotFo
rm.IrcBot_Commands_4_Edit.Text)))<>nil then
  begin
    if IrcBotForm.IrcBot_Commands_1_CheckBox.Checked then
    begin
      for i:=0 to Irc_PageControl.PageCount-1 do
      begin
        if Irc_PageControl.Pages[i].Caption=Channel then
        begin
          try
            IdIRC.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+IrcBotForm.IrcBot_Commands_1_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_1_CheckBox.Caption);
          except
            on e:Exception do Reconnect(E.Message);
          end;

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'1<'+IdIRC.Nick+'>
'+TAG_BOLD+IrcBotForm.IrcBot_Commands_1_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_1_CheckBox.Caption);
          // Работа з файлами лога

IRCLogs (Channel, IdIRC.Nick, IrcBotForm.IrcBot_Commands_1_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_1_CheckBox.Caption);
          IdAntiFreeze.Sleep(1000);
        end;
      end;
    end;
  end;

  if IrcBotForm.IrcBot_Commands_2_CheckBox.Checked then

```

```

begin
    for i:=0 to Irc_PageControl.PageCount-1 do
        begin
            if Irc_PageControl.Pages[i].Caption=Channel then
                begin
                    try
                        IdIRC.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+IrcBotForm.IrcBot_Commands_2_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_2_CheckBox.Caption);
                    except
                        on e:Exception do Reconnect(E.Message);
                    end;

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'1<'+IdIRC.Nick+'>
:'+TAG_BOLD+IrcBotForm.IrcBot_Commands_2_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_2_CheckBox.Caption);
                // Робота з файлами лога

IRCLogs (Channel, IdIRC.Nick, IrcBotForm.IrcBot_Commands_2_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_2_CheckBox.Caption);
                IdAntiFreeze.Sleep(1000);
            end;
        end;
    end;

    if IrcBotForm.IrcBot_Commands_3_CheckBox.Checked then
        begin
            for i:=0 to Irc_PageControl.PageCount-1 do
                begin
                    if Irc_PageControl.Pages[i].Caption=Channel then
                        begin
                            try
                                IdIRC.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+IrcBotForm.IrcBot_Commands_3_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_3_CheckBox.Caption);
                            except
                                on e:Exception do Reconnect(E.Message);
                            end;

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'1<'+IdIRC.Nick+'>
:'+TAG_BOLD+IrcBotForm.IrcBot_Commands_3_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_3_CheckBox.Caption);
                // Робота з файлами лога

IRCLogs (Channel, IdIRC.Nick, IrcBotForm.IrcBot_Commands_3_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_3_CheckBox.Caption);
                IdAntiFreeze.Sleep(1000);
            end;
        end;
    end;

    if IrcBotForm.IrcBot_Commands_4_CheckBox.Checked then
        begin
            for i:=0 to Irc_PageControl.PageCount-1 do
                begin
                    if Irc_PageControl.Pages[i].Caption=Channel then
                        begin
                            try
                                IdIRC.WriteLine('PRIVMSG '+Channel+'
:'+TAG_BOLD+IrcBotForm.IrcBot_Commands_4_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_4_CheckBox.Caption);
                            except
                                on e:Exception do Reconnect(E.Message);
                            end;

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'1<'+IdIRC.Nick+'>

```

```

'+TAG_BOLD+IrcBotForm.IrcBot_Commands_4_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_4_CheckBox.Caption);
    // Робота з файлами лога

IRCLogs(Channel, IdIRC.Nick, IrcBotForm.IrcBot_Commands_4_Edit.Text+' -
'+IrcBotForm.IrcBot_Commands_4_CheckBox.Caption);
    IdAntiFreeze.Sleep(1000);
    end;
    end;
    end;

    end;
    end;
    end;

    end;
end;

// Перепідключення при помилці
//Перевірка наявності підключення до інтернету
function IsConnectedToInternet: Boolean;
var
    dwConnectionTypes: DWORD;
begin

dwConnectionTypes:=INTERNET_CONNECTION_MODEM+INTERNET_CONNECTION_LAN+INTERNET_CO
NNECTION_PROXY;
    Result:=InternetGetConnectedState(@dwConnectionTypes, 0);
end;

// Перевірка наявності Інтернету
procedure TForm1.Inet_TimerTimer(Sender: TObject);
begin
    if IsConnectedToInternet=false then
    begin
        if IdIRC.Connected then
        begin
            Irc_Ping_Timer.Enabled:=false;

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_BOLD+TAG_COLOR+'4Немає підключення до Інтернету');
            IdIRC.Disconnect;
        end;
    end;
end;

procedure TForm1.Reconnect (Err: String);
begin
    if Reconnect_Integer=3 then
    begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_BOLD+TAG_COLOR+'4'+Err);
        IdIRC.Disconnect;
        exit;
    end
    else
    begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_BOLD+TAG_COLOR+'4'+Err);
        Reconnect_Integer:=Reconnect_Integer+1;
        IdIRC.Disconnect;
        IdAntiFreeze.Sleep(1000);
        Irc_Connected_ButtonClick(nil);
    end;
end;

```

```

// Ведення логів каналів і приватів
procedure TForm1.IRCLogs(filename: String; User: String; Content: String);
var
  LogFile      : TextFile;
begin
  if (IrcSettingsForm.IrcSettings_Servis_Logs_CheckBox.Checked) and
    (Irc_PageControl.TabIndex>0) then
    begin
      AssignFile(LogFile, ExtractFileDir(ParamStr(0))+'\IRC\Logs\'+IdIRC.Host+'\'+filename+'.txt');
      if
        FileExists(ExtractFilePath(paramstr(0))+'\IRC\Logs\'+IdIRC.Host+'\'+filename+'.txt')=false then
          begin
            CreateDirectory(PChar(ExtractFileDir(ParamStr(0))+'\IRC'), nil);
            CreateDirectory(PChar(ExtractFileDir(ParamStr(0))+'\IRC\Logs'), nil);

            CreateDirectory(PChar(ExtractFileDir(ParamStr(0))+'\IRC\Logs\'+IdIRC.Host), nil);
            Rewrite(LogFile);
            CloseFile(LogFile);
          end;
          Append(LogFile);
          Writeln(LogFile, '['+DateTimeToStr(Now)+' ']+ '<'+User+'>'
            '+ParseColorString(Content));
          CloseFile(LogFile);
        end;
      end;

// Процедура вставки спецтегів у поля введення
procedure TForm1.IrcColor(ComboBox: TComboBox; Edit: TEdit; var Key: Word;
  Shift: TShiftState);
var
  C : Char;
begin
  C := UpCase(Char(Key));
  if (C in ['B', 'U', 'R', 'K', #13]) and (ssCtrl in Shift) then
    begin
      case C of
        'B' : begin
          if ComboBox<>nil then ComboBox.SelText:=TAG_BOLD;
          if Edit<>nil then Edit.SelText:=TAG_BOLD;
        end;
        'U' : begin
          if ComboBox<>nil then ComboBox.SelText:=TAG_UNDERLINE;
          if Edit<>nil then Edit.SelText:=TAG_UNDERLINE;
        end;
        'R' : begin
          if ComboBox<>nil then ComboBox.SelText:=TAG_INVERSE;
          if Edit<>nil then Edit.SelText:=TAG_INVERSE;
        end;
        'K' : begin
          if ComboBox<>nil then ComboBox.SelText:=TAG_COLOR;
          if Edit<>nil then Edit.SelText:=TAG_COLOR;
        end;
      end;
      Key := 0;
    end;
  end;

// Процедура створення нової вкладки для привата
procedure TForm1.IrcNewPrivatPage(Caption: String; UserNick: String;
  MessageText: String; PageImage: Integer);
var
  NewPage      : TTabSheet;
  NewPanel     : TPanel;
  NewEdit      : TEdit;
  NewIRCMemo   : TIRCMemo;
begin

```

```

if Caption='' then exit;
if Caption[1]='я' then Caption[1]:='Я';

// Створюємо вкладку для привату
NewPage:=TTabSheet.Create(Self);
NewPage.PageControl:=Irc_PageControl; // Вставляємо закладку
NewPage.Name:='TabSheet'+IntToStr(TabSheet_Integer);
NewPage.Caption:=Caption;
NewPage.ImageIndex:=PageImage;
// Процедура показу вкладки привату
NewPage.OnShow:=Irc_Privat_Show;

// Вставляємо панель
NewPanel:=TPanel.Create(Self);
NewPanel.Parent:=NewPage;
NewPanel.Name:=NewPage.Name+'_Panel';
NewPanel.BevelOuter:=bvNone;
NewPanel.Caption:='';
NewPanel.Height:=21;
NewPanel.Align:=alBottom;

NewEdit:=TEdit.Create(Self); // Вставляємо Edit
NewEdit.Parent:=NewPanel;
NewEdit.Name:=NewPage.Name+'_Edit';
NewEdit.Top:=0;
NewEdit.Width:=Irc_Server_Edit.Width;
NewEdit.Left:=0;
NewEdit.Text:='';
NewEdit.MaxLength:=510;
// Процедура відправлення повідомлень у приват
NewEdit.OnKeyPress:=Irc_Channels_MessageText_EditKeyPress;
// Процедура вставки спецтегів
NewEdit.OnKeyDown:=Irc_Channels_EditKeyDown;

NewIRCMemo:=TIRCMemo.Create(Self); //вставляємо IRCMemo
NewIRCMemo.Parent:=NewPage;
NewIRCMemo.Name:=NewPage.Name+'_IRCMemo';
NewIRCMemo.Align:=alClient;
NewIRCMemo.ReadOnly:=true;
NewIRCMemo.Clear;
// Перехід на Edit при відпусканні миші
NewIRCMemo.OnMouseUp:=Irc_Server_IRCMemoMouseUp;
// Смайли
NewIRCMemo.SmilesInfo[0].Smile:=' :) ';
NewIRCMemo.SmilesInfo[0].ImageIndex:=0;
NewIRCMemo.SmilesInfo[1].Smile:=' =) ';
NewIRCMemo.SmilesInfo[1].ImageIndex:=0;
NewIRCMemo.SmilesInfo[2].Smile:=' :-) ';
NewIRCMemo.SmilesInfo[2].ImageIndex:=0;

NewIRCMemo.SmilesInfo[3].Smile:=' (: ';
NewIRCMemo.SmilesInfo[3].ImageIndex:=1;
NewIRCMemo.SmilesInfo[4].Smile:=' :-( ';
NewIRCMemo.SmilesInfo[4].ImageIndex:=1;

NewIRCMemo.SmilesInfo[5].Smile:=' :D ';
NewIRCMemo.SmilesInfo[5].ImageIndex:=2;
NewIRCMemo.SmilesInfo[6].Smile:=' :-D ';
NewIRCMemo.SmilesInfo[6].ImageIndex:=2;

NewIRCMemo.SmilesInfo[7].Smile:=' :P ';
NewIRCMemo.SmilesInfo[7].ImageIndex:=3;
NewIRCMemo.SmilesInfo[8].Smile:=' :-P ';
NewIRCMemo.SmilesInfo[8].ImageIndex:=3;

NewIRCMemo.SmilesInfo[9].Smile:=' ;) ';
NewIRCMemo.SmilesInfo[9].ImageIndex:=4;
NewIRCMemo.SmilesInfo[10].Smile:=' ;-) ';

```

```

NewIRCMemo.SmilesInfo[10].ImageIndex:=4;

NewIRCMemo.SmilesInfo[11].Smile:='8)';
NewIRCMemo.SmilesInfo[11].ImageIndex:=5;
NewIRCMemo.SmilesInfo[12].Smile:=' 8-8-);
NewIRCMemo.SmilesInfo[12].ImageIndex:=5;

NewIRCMemo.SmilesInfo[13].Smile:=':\';
NewIRCMemo.SmilesInfo[13].ImageIndex:=6;
NewIRCMemo.SmilesInfo[14].Smile:=':-\';
NewIRCMemo.SmilesInfo[14].ImageIndex:=6;

NewIRCMemo.SmilesInfo[15].Smile:='#39+'(';
NewIRCMemo.SmilesInfo[15].ImageIndex:=7;
NewIRCMemo.SmilesInfo[16].Smile:=':'+'#39+'(';
NewIRCMemo.SmilesInfo[16].ImageIndex:=7;

NewIRCMemo.Smiles:=IRC_ImageList;

// Збільшуємо значення TabSheet_Integer на 1
TabSheet_Integer:=TabSheet_Integer+1;

// Додаємо текст повідомлення
if (UserNick='') or (MessageText='') then
  exit
else NewIRCMemo.AddFormattedString(TAG_COLOR+'1<'+UserNick+'> '+MessageText);
end;

// Підключення
procedure TForm1.Irc_Connected_ButtonClick(Sender: TObject);
begin
  // Очищаємо стан режиму Away
  Irc_Away_Status_Label.Visible:=false;
  if IdIRC.Connected then
  begin
    try
      IdIRC.Quit(IrcSettingsForm.IrcSettings_Quit_Edit.Text+' -
'+IdIRC.Replies.ClientInfo);
    except
      // Помилка
      on e:Exception do
        TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4'+E.Message);
      end;
      IdIRC.Disconnect;
    end
  else
  begin
    Irc_Server_IRCMemo.Clear;
    if (IrcSettingsForm.IrcSettings_Username_Edit.Text='') or
(IrcSettingsForm.IrcSettings_AltNick_Edit.Text='') then
    begin
      IrcSettingsForm.ShowModal;
      exit;
    end
  else
    if Irc_Server_Address_Combobox.Text='' then
    begin
      TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Виберіть будь ласка сервер');
      ActiveControl:=Irc_Server_Address_Combobox;
      exit;
    end
  else
    if Irc_Nik_Edit.Text='' then
    begin
      TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Введіть будь ласка нік');
      ActiveControl:=Irc_Nik_Edit;
    end
  end
end

```

```

exit;
end
else
if Irc_Password_Edit.Text='' then
begin
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Введіть будь ласка пароль');
ActiveControl:=Irc_Password_Edit;
exit;
end
else
Irc_PageControl.ActivePageIndex:=0;
IdIRC.Replies.ClientInfo:=TAG_BOLD+Form1.Caption;
IdIRC.Replies.Version:=IdIRC.Replies.ClientInfo;
IdIRC.Replies.Finger:=IrcSettingsForm.IrcSettings_CTCP_Finger_Edit.Text;
IdIRC.Replies.UserInfo:=IrcSettingsForm.IrcSettings_CTCP_Userinfo_Edit.Text;
IdIRC.Nick:=Irc_Nik_Edit.Text;
IdIRC.AltNick:=IrcSettingsForm.IrcSettings_AltNick_Edit.Text;
IdIRC.Username:=IrcSettingsForm.IrcSettings_Username_Edit.Text;
IdIRC.RealName:=IrcSettingsForm.IrcSettings_Username_Edit.Text;
IdIRC.Password:=Irc_Password_Edit.Text;
IdIRC.Host:=Irc_Server_Address_Combobox.Text;
IdIRC.Port:=IrcSettingsForm.IrcSettings_Port_SpinEdit.Value;
try
    IdIRC.Connect (500);
except
    on e:Exception do Reconnect (E.Message);
end;
end;
end;

// Підключений
procedure TForm1.IdIRCConnected(Sender: TObject);
begin
    Irc_Connected_Button.Caption:='Відключитися';
    Irc_Nik_Edit.Enabled:=false;
    Irc_Password_Edit.Enabled:=false;
    Irc_Server_Address_Combobox.Enabled:=false;
    Irc_Server.Caption:=IdIRC.Host;
    Reconnect_Integer:=0;
end;

// Відключення
procedure TForm1.IdIRCDisconnected(Sender: TObject);
var
    i : Integer;
begin
    Irc_Connected_Button.Caption:='Підключитися';
    Irc_Nik_Edit.Enabled:=true;
    Irc_Password_Edit.Enabled:=true;
    Irc_Server_Address_Combobox.Enabled:=true;
    // Закриття вкладок каналів, якщо не стоїть галочка про перезахід
    if IrcSettingsForm.IrcSettings_Servis_Quit_CheckBox.Checked=false then
        begin
            Irc_PageControl.ActivePageIndex:=0;
            for i:=1 to Irc_PageControl.PageCount-1 do
                begin
                    if Irc_PageControl.PageCount-1>1 then
                        Irc_PageControl.Pages[Irc_PageControl.PageCount-1].Free
                    else Irc_PageControl.Pages[Irc_PageControl.PageCount-
1].TabVisible:=false;
                end;
            end;
        end;
end;

// Заходимо в канал (/join) і створюємо нову вкладку з назвою каналу
procedure TForm1.IdIRCJoined(Sender: TObject; AChannel: TIdIRCChannel);
var
    NewPage      : TTabSheet;

```

```

NewPanel      : TPanel;
NewEdit       : TEdit;
NewListBox    : TListBox;
NewSplitter   : TSplitter;
NewIRCMemo    : TIRCMemo;
i             : Integer;
begin
// Якщо вкладка з такою назвою вже існує, то показати її, очищаємо всі поля й
зупиняємо процедуру
for i:=0 to Irc_PageControl.PageCount-1 do
  begin
    if
AnsiUpperCase(Irc_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
then
  begin
    Irc_PageControl.Pages[i].TabVisible:=true;
    Irc_PageControl.ActivePage:=Irc_PageControl.Pages[i];
    Exit;
  end;
end;

// Створюємо вкладку для каналу
NewPage:=TTabSheet.Create(Self);
NewPage.PageControl:=Irc_PageControl; // Вставляємо закладку
NewPage.Name:='TabSheet'+IntToStr(TabSheet_Integer);
NewPage.Caption:=AChannel.Name;
NewPage.ImageIndex:=4;
// Процедура показу вкладки каналу
NewPage.OnShow:=Irc_Channel_Show;

// Вставляємо панель
NewPanel:=TPanel.Create(Self);
NewPanel.Parent:=NewPage;
NewPanel.Name:=NewPage.Name+'_Panel';
NewPanel.BevelOuter:=bvNone;
NewPanel.Caption:='';
NewPanel.Height:=21;
NewPanel.Align:=alBottom;

NewEdit:=TEdit.Create(Self); // Вставляємо Edit
NewEdit.Parent:=NewPanel;
NewEdit.Name:=NewPage.Name+'_Edit';
NewEdit.Top:=0;
NewEdit.Width:=Irc_Server_Edit.Width;
NewEdit.Left:=0;
NewEdit.Text:='';
NewEdit.MaxLength:=510;
// Процедура відправлення повідомлень на канал
NewEdit.OnKeyPress:=Irc_Channels_MessageText_EditKeyPress;
// Процедура вставки спецтегів
NewEdit.OnKeyDown:=Irc_Channels_EditKeyDown;

NewListBox:=TListBox.Create(Self); // Вставляємо ListBox
NewListBox.Parent:=NewPage;
NewListBox.Name:=NewPage.Name+'_ListBox';
NewListBox.Width:=120;
NewListBox.Align:=alRight;
NewListBox.Sorted:=true;
NewListBox.ShowHint:=true;
NewListBox.PopupMenu:=Irc_Channel_ListBox_PopupMenu;
// Процедура кліку за списком користувачів на каналі
NewListBox.OnClick:=Irc_Channels_ListBoxClick;
// Процедура подвійного кліку за списком користувачів
NewListBox.OnDblClick:=Irc_Channels_ListBox_PopupMenu_PrivatClick;
// Процедура показу кількості користувачів на каналі
NewListBox.OnMouseMove:=Irc_Channel_ListBoxMouseMove;
// Перехід на Edit при відпусканні миші
NewListBox.OnMouseUp:=Irc_Server_IRCMemoMouseUp;

```

```

// Вставляємо Splitter
NewSplitter:=TSplitter.Create(Self); // Вставляємо Splitter
NewSplitter.Parent:=NewPage;
NewSplitter.Name:=NewPage.Name+'_Splitter';
NewSplitter.ShowHint:=true;
NewSplitter.Align:=alRight;
NewSplitter.Beveled:=true;

NewIRCMemo:=TIRCMemo.Create(Self); //вставляємо IRCMemo
NewIRCMemo.Parent:=NewPage;
NewIRCMemo.Name:=NewPage.Name+'_IRCMemo';
NewIRCMemo.Align:=alClient;
NewIRCMemo.ReadOnly:=true;
NewIRCMemo.Clear;
// Перехід на Edit при відпусканні миші
NewIRCMemo.OnMouseUp:=Irc_Server_IRCMemoMouseUp;
// Процедура показу властивостей каналу
NewIRCMemo.OnDblClick:=Irc_ChannelSettings_SpeedButtonClick;

// Смайли
NewIRCMemo.SmilesInfo[0].Smile:=' :) ';
NewIRCMemo.SmilesInfo[0].ImageIndex:=0;
NewIRCMemo.SmilesInfo[1].Smile:=' =) ';
NewIRCMemo.SmilesInfo[1].ImageIndex:=0;
NewIRCMemo.SmilesInfo[2].Smile:=' :-) ';
NewIRCMemo.SmilesInfo[2].ImageIndex:=0;

NewIRCMemo.SmilesInfo[3].Smile:=' (: ';
NewIRCMemo.SmilesInfo[3].ImageIndex:=1;
NewIRCMemo.SmilesInfo[4].Smile:=' :- ( ';
NewIRCMemo.SmilesInfo[4].ImageIndex:=1;

NewIRCMemo.SmilesInfo[5].Smile:=' :D ';
NewIRCMemo.SmilesInfo[5].ImageIndex:=2;
NewIRCMemo.SmilesInfo[6].Smile:=' :-D ';
NewIRCMemo.SmilesInfo[6].ImageIndex:=2;

NewIRCMemo.SmilesInfo[7].Smile:=' :P ';
NewIRCMemo.SmilesInfo[7].ImageIndex:=3;
NewIRCMemo.SmilesInfo[8].Smile:=' :-P ';
NewIRCMemo.SmilesInfo[8].ImageIndex:=3;

NewIRCMemo.SmilesInfo[9].Smile:=' ;) ';
NewIRCMemo.SmilesInfo[9].ImageIndex:=4;
NewIRCMemo.SmilesInfo[10].Smile:=' ;-) ';
NewIRCMemo.SmilesInfo[10].ImageIndex:=4;

NewIRCMemo.SmilesInfo[11].Smile:=' 8) ';
NewIRCMemo.SmilesInfo[11].ImageIndex:=5;
NewIRCMemo.SmilesInfo[12].Smile:=' 8-8-) ';
NewIRCMemo.SmilesInfo[12].ImageIndex:=5;

NewIRCMemo.SmilesInfo[13].Smile:=' : \ ';
NewIRCMemo.SmilesInfo[13].ImageIndex:=6;
NewIRCMemo.SmilesInfo[14].Smile:=' :- \ ';
NewIRCMemo.SmilesInfo[14].ImageIndex:=6;

NewIRCMemo.SmilesInfo[15].Smile:=' #39+ ( ';
NewIRCMemo.SmilesInfo[15].ImageIndex:=7;
NewIRCMemo.SmilesInfo[16].Smile:=' :'+#39+ ( ';
NewIRCMemo.SmilesInfo[16].ImageIndex:=7;

NewIRCMemo.Smiles:=IRC_ImageList;

// Перемикаємося на нову вкладку
Irc_PageControl.ActivePage:=NewPage;

// Збільшуємо значення TabSheet_Integer на 1

```

```

TabSheet_Integer:=TabSheet_Integer+1;

// Робота з файлами лога
IRCLogs(AChannel.Name,',' , 'Захід на канал');
end;

// Одержування списоку користувачів в каналі
function SneakyGetUserState(Data: Integer): TIRCUserStates;
begin
    Result := [];
    if (Data and IRC_OP) <> 0 then Include(Result, ircOp);
    if (Data and IRC_VOICE) <> 0 then Include(Result, ircVoice);
end;

procedure TForm1.IdIRCNames(Sender: TObject; AUsers: TIdIRCUsers;
    AChannel: TIdIRCChannel);
var
    iUser,iIndex : Integer;
    pState       : TIRCUserStates;
    i            : Integer;
begin
    try
    for i:=0 to Irc_PageControl.PageCount-1 do
        begin
            if
                AnsiUpperCase(Irc_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
            then
                begin
                    TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Clear;
                    for iUser := 0 to AUsers.Count - 1 do
                        begin
                            Application.ProcessMessages;

                            if AChannel.Find(AUsers.Items[iUser].Nick, iIndex) then
                                begin
                                    pState :=
                                        SneakyGetUserState(Integer(AChannel.Names.Objects[iIndex]));

                                    if ircOp in pState then
                                        TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add('@'+
                                            AUsers.Items[iUser].Nick);

                                    if ircVoice in pState then
                                        TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add('+'+
                                            AUsers.Items[iUser].Nick);

                                    if pState=[] then
                                        TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add(AUsers.Items[iUser].Nick);
                                    end;
                                end;
                            end;
                        end;
                    end;
                except
                end;
            end;

// Прийом даних із сервера
// Одержуємо список каналів
// Відображаємо другий нік
// Обробка цифрових відповідей, які створюються сервером як відповідна реакція на
введені команди
procedure TForm1.IdIRCRaw(Sender: TObject; AUser: TIdIRCUser; ACommand,
    AContent: String; var Suppress: Boolean);
var
    b,i,n,m      : integer;
    s,d,f,g,l    : String;
begin

```

```

for i:=0 to Irc_PageControl.PageCount-1 do
begin
while
TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).Lines.Count>IrcSettingsForm.IrcSettings_Servis_MaxLines_SpinEdit.Value do

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).Lines.Delete(0);
end;
i:=Pos(' ',AContent);

// Одержуємо список каналів (/list)
// Показуємо вкладку й очищуємо список каналів
if ACommand=IntToStr(321) then
begin
if Channels.TabVisible=false then
begin
Channels.TabVisible:=true;
Channels_IRCMemo.Clear;
Irc_PageControl.ActivePageIndex:=1;
end;
Channels_ListBox.Clear;
end;
// Одержуємо список каналів (/list)
if ACommand=IntToStr(322) then
begin
Application.ProcessMessages;
Channels_ListBox.Items.Add(Copy(AContent,i+1,Length(AContent)-i));
end;

// Відображаємо другий нік, якщо перший уже зайнятий
if ACommand=IntToStr(433) then Irc_Nik_Edit.Text:=IdIRC.AltNick;

// Обробка цифрових відповідей, які створюються сервером як відповідна реакція на вводяться команди
if ACommand=IntToStr(200) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(201) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(202) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(203) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(204) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(205) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(206) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(208) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(211) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(212) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));
if ACommand=IntToStr(213) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(Copy(AContent,i+1,Length(AContent)-i));

```



```

if ACommand=IntToStr(303) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(305) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(306) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(311) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(312) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(313) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(314) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(315) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(317) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(318) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(319) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));

// Ці команди вже оброблені /LIST
//if ACommand=IntToStr(321) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
//if ACommand=IntToStr(322) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
//if ACommand=IntToStr(323) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));

// Ці команди вже оброблені в процедурі IdIRCChannelMode
// Одержання режимів каналу
if (ACommand=IntToStr(324)) and (IrcChannelSettingsForm.Showing) then
begin
s:=Copy (AContent, i+1, Length (AContent) -i);
n:=Pos (' ', s);
d:=Copy (s, n+1, Length (s) -n);
m:=Pos (' ', d);
f:=Copy (d, 0, m-1);
g:=Copy (d, m+1, Length (d) -m);

// +t
if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('t')))<>nil then
IrcChannelSettingsForm.IrcChannelSettings_Mode_t_CheckBox.Checked:=true;
// +n
if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('n')))<>nil then
IrcChannelSettingsForm.IrcChannelSettings_Mode_n_CheckBox.Checked:=true;
// +i
if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('i')))<>nil then
IrcChannelSettingsForm.IrcChannelSettings_Mode_i_CheckBox.Checked:=true;
// +m
if StrPos (PChar (AnsiUpperCase (f)), PChar (AnsiUpperCase ('m')))<>nil then
IrcChannelSettingsForm.IrcChannelSettings_Mode_m_CheckBox.Checked:=true;
// +p

```

```

    if StrPos(PChar(AnsiUpperCase(f)), PChar(AnsiUpperCase('p')))<>nil then
IrcChannelSettingsForm.IrcChannelSettings_Mode_p_CheckBox.Checked:=true;
    // +s
    if StrPos(PChar(AnsiUpperCase(f)), PChar(AnsiUpperCase('s')))<>nil then
IrcChannelSettingsForm.IrcChannelSettings_Mode_s_CheckBox.Checked:=true;
    // +l
    if StrPos(PChar(AnsiUpperCase(f)), PChar(AnsiUpperCase('l')))<>nil then
    begin
        IrcChannelSettingsForm.IrcChannelSettings_Mode_l_CheckBox.Checked:=true;
        if StrPos(PChar(AnsiUpperCase(f)), PChar(AnsiUpperCase('k')))<>nil then
        begin
            for b:=0 to (Length(g)-1) do
            begin
                if (Ord(g[b])>47) and (Ord(g[b])<58) then
                begin
                    l:=l+g[b];
                end;
            end;

IrcChannelSettingsForm.IrcChannelSettings_Mode_l_SpinEdit.Value:=StrToInt(l);
            end
            else
            begin

IrcChannelSettingsForm.IrcChannelSettings_Mode_l_SpinEdit.Value:=StrToInt(Copy(g
, 0, Length(g)-1));
            end;
        end;
        // +k
        if StrPos(PChar(AnsiUpperCase(f)), PChar(AnsiUpperCase('k')))<>nil then
        begin
            IrcChannelSettingsForm.IrcChannelSettings_Mode_k_CheckBox.Checked:=true;
            if StrPos(PChar(AnsiUpperCase(f)), PChar(AnsiUpperCase('l')))<>nil then
            begin

IrcChannelSettingsForm.IrcChannelSettings_Mode_k_Edit.Text:=Copy(g, Length(l)+1, L
ength(g)-1);
            end
            else
            begin

IrcChannelSettingsForm.IrcChannelSettings_Mode_k_Edit.Text:=Copy(g, 0, Length(g)-
1);
            end;
            end;
            // Показ режиму каналу в заголовку
            IrcChannelSettingsForm.IrcChannelSettings_Mode_GroupBox.Caption:='MODE '+d;
        end;

//if ACommand=IntToStr(329) then
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(Copy(AContent, i+1, Length(AContent)-i));

// Немає топика на каналі
//if ACommand=IntToStr(331) then
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(Copy(AContent, i+1, Length(AContent)-i));

// Ця команда вже оброблена в процедурі IdIRCTopic
//if ACommand=IntToStr(332) then
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(Copy(AContent, i+1, Length(AContent)-i));

if ACommand=IntToStr(341) then
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(Copy(AContent, i+1, Length(AContent)-i));
if ACommand=IntToStr(342) then
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(Copy(AContent, i+1, Length(AContent)-i));

```

```

if ACommand=IntToStr(351) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(352) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));

// NAMES
if ACommand=IntToStr(353) then
begin
s:=Copy (AContent, i+3, Length (AContent) -i);
n:=Pos (' ', s);
if
AnsiUpperCase (Irc_PageControl.ActivePage.Caption) <> AnsiUpperCase (Copy (s, 0, n-1))
then

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+3, Length (AContent) -i));
end;
// End of /NAMES list.
if ACommand=IntToStr(366) then
begin
s:=Copy (AContent, i+1, Length (AContent) -i);
n:=Pos (' ', s);
if
AnsiUpperCase (Irc_PageControl.ActivePage.Caption) <> AnsiUpperCase (Copy (s, 0, n-1))
then

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
end;

if ACommand=IntToStr(364) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(365) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
// Список банів
if ACommand=IntToStr(367) then
begin
s:=Copy (AContent, i+1, Length (AContent) -i);
n:=Pos (' ', s);
d:=Copy (s, n+1, Length (s) -n);
m:=Pos (' ', d);
f:=Copy (d, 0, m-1);
if IrcChannelSettingsForm.Showing then
IrcChannelSettingsForm.IrcChannelSettings_Ban_ListBox.Items.Add (f)
else

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
end;
//if ACommand=IntToStr(368) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(369) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(371) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(372) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));
if ACommand=IntToStr(374) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent) -i));

```

```

if ACommand=IntToStr(375) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent)-i));
if ACommand=IntToStr(376) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent)-i));
if ACommand=IntToStr(381) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent)-i));
if ACommand=IntToStr(382) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent)-i));
if ACommand=IntToStr(391) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent)-i));
if ACommand=IntToStr(392) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent)-i));
if ACommand=IntToStr(393) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent)-i));
if ACommand=IntToStr(394) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent)-i));
if ACommand=IntToStr(395) then
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (Copy (AContent, i+1, Length (AContent)-i));
end;

// Показувати отримані команди у вікні сервера
procedure TForm1.IdIRCReceive(Sender: TObject; ACommand: String);
begin
  if IrcSettingsForm.IrcSettings_Show_Receive_Commands_CheckBox.Checked then
    Irc_Server_IRCMemo.Lines.Add(ACommand);
end;

// Показувати відправлені команди у вікні сервера
procedure TForm1.IdIRCSend(Sender: TObject; ACommand: String);
begin
  if IrcSettingsForm.IrcSettings_Show_Send_Commands_CheckBox.Checked then
    Irc_Server_IRCMemo.Lines.Add('-> '+ACommand);
end;

// Прийом повідомлень у каналі й прийом особистих повідомлень
procedure TForm1.IdIRCMessage(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel; Content: String);
var
  i, n      : Integer;
begin
  // Дані про стан пінгу
  Irc_Ping_Timer.Enabled:=False;
  Irc_Ping_Timer.Enabled:=True;
  // Видалення рядків
  for i:=0 to Irc_PageControl.PageCount-1 do
    begin
      while
TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).Lines.Count>Irc
cSettingsForm.IrcSettings_Servis_MaxLines_SpinEdit.Value do
TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).Lines.Delete(0
);
      end;
    // Прийом повідомлень у каналі
    try
      for i:=0 to Irc_PageControl.PageCount-1 do
        begin
          Application.ProcessMessages;

```

```

        if
AnsiUpperCase(Irc_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
then
        begin

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedString(TAG_COLOR+'1<'+AUser.Nick+'> '+Content);

        // Показати повідомлення
        if
IrcSettingsForm.IrcSettings_Show_Channels_Messages_CheckBox.Checked then
        Begin
            MSNPopUp.Title:='IRC '+AChannel.Name+' '+AUser.Nick;
            MSNPopUp.Text:=ParseColorString(Content);
            MSNPopUp.GradientColor1:=$00FFCC99;
            MSNPopUp.ShowPopUp;
        end;

        // Робота з файлами лога
        IRCLogs(AChannel.Name,AUser.Nick,Content);

        // Вивід повідомлення при згадуванні слів зі списку
        IrcFindText(AUser.Nick,Content,AChannel.Name);

        // Банити користувачів за використання слів зі списку
        IrcBotKick(AUser.Nick,Content,AChannel.Name);

        // Команди
        IrcBotCommands(AUser.Nick,Content,AChannel.Name);

        // Озвучити повідомлення в активному каналі
        if
(IrcSettingsForm.IrcSettings_Sound_Speak_Channel_CheckBox.Checked) and
(AnsiUpperCase(Irc_PageControl.ActivePage.Caption)=AnsiUpperCase(AChannel.Name))
then
        begin
            IrcSpeak(AUser.Nick,ParseColorString(Content),True);
        end;

        // Озвучити повідомлення у всіх каналах
        if IrcSettingsForm.IrcSettings_Sound_Speak_Channels_CheckBox.Checked
then
        begin
            IrcSpeak(AUser.Nick,ParseColorString(Content),True);
        end;

        // Звук
        if IrcSettingsForm.IrcSettings_Sound_Message_CheckBox.Checked then
        begin
            Wave.WaveName:='message';
            Wave.Execute;
        end;

        // Іконка повідомлення
        if Irc_PageControl.ActivePage.Caption<>AChannel.Name then

TTabSheet(FindComponent(Irc_PageControl.Pages[i].Name)).ImageIndex:=5;
        end;
    end;
except
    // Прийом особистих повідомлень
    // Показати повідомлення
    if IrcSettingsForm.IrcSettings_Show_Privat_Messages_CheckBox.Checked then
    Begin
        MSNPopUp.Title:='IRC '+AUser.Nick;
        MSNPopUp.Text:=ParseColorString(Content);
        MSNPopUp.GradientColor1:=$00D8BEDA;
        MSNPopUp.ShowPopUp;
    end;
end;

```

```

// Робота з файлами лога
IRCLogs (AUser.Nick,AUser.Nick,Content);

// Озвучити повідомлення
if IrcSettingsForm.IrcSettings_Sound_Speak_Privat_CheckBox.Checked then
begin
  IrcSpeak (AUser.Nick,ParseColorString (Content),False);
end;

for i:=0 to Irc_PageControl.PageCount-1 do
begin
  Application.ProcessMessages;
  if
AnsiUpperCase (Irc_PageControl.Pages[i].Caption)=AnsiUpperCase (AUser.Nick) then
begin
  // Прийом особистих повідомлень до окремої вкладки

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing (TAG_COLOR+'1<'+AUser.Nick+'> '+Content);

  // Звук для особистого повідомлення
  if IrcSettingsForm.IrcSettings_Sound_Privat_CheckBox.Checked then
begin
  Wave.WaveName:='icq_o_o';
  Wave.Execute;
end;

  // Іконка повідомлення
  if Irc_PageControl.ActivePage.Caption<>AUser.Nick then
TTabSheet (FindComponent (Irc_PageControl.Pages[i].Name)).ImageIndex:=6;
end;
end;
// Якщо вкладка привата з такою назвою вже існує, то зупиняємо процедуру
for n:=0 to Irc_PageControl.PageCount-1 do
begin
  if
AnsiUpperCase (Irc_PageControl.Pages[n].Caption)=AnsiUpperCase (AUser.Nick) then
begin
  // Зупиняємо процедуру
  Exit;
end;
end;
// Створюємо нову вкладку й додаємо в неї повідомлення
IrcNewPrivatPage (AUser.Nick,AUser.Nick,Content,6);

// Звук для особистого повідомлення
if IrcSettingsForm.IrcSettings_Sound_Privat_CheckBox.Checked then
begin
  Wave.WaveName:='icq_o_o';
  Wave.Execute;
end;

// Прийом повідомлень у режимі AWAY
if IdIRC.Away then
begin
  for i:=0 to Irc_PageControl.PageCount-1 do
begin
Irc_Server_IRCMemo.Lines.Add (Irc_PageControl.Pages[i].Caption+'=='+AUser.Nick);
  if
AnsiUpperCase (Irc_PageControl.Pages[i].Caption)=AnsiUpperCase (AUser.Nick) then
begin
  try
    IdIRC.WriteLine ('PRIVMSG '+AUser.Nick+' :'+away_message);
  except
    on e:Exception do Reconnect (E.Message);
  end;
end;
end;
end;

```

```

end;

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedString (TAG_COLOR+'1<'+IdIRC.Nick+'> '+away_message);
    // Робота з файлами лога
    IRCLogs (AUser.Nick,AUser.Nick,Content);
end;
end;
end;
end;

// Прийом Action
procedure TForm1.IdIRCAction(Sender: TObject; AUser: TIdIRCUser;
    AChannel: TIdIRCChannel; Content: String);
var
    i : Integer;
begin
    for i:=0 to Irc_PageControl.PageCount-1 do
        begin
            if
                AnsiUpperCase (Irc_PageControl.Pages[i].Caption)=AnsiUpperCase (AChannel.Name)
            then
                begin

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedString (TAG_COLOR+'6* '+AUser.Nick+' '+Content);

                    // Показати повідомлення
                    if IrcSettingsForm.IrcSettings_Show_Channels_Messages_CheckBox.Checked
then
                        Begin
                            MSNPopUp.Title:='IRC '+AChannel.Name;
                            MSNPopUp.Text:=ParseColorString (Content);
                            MSNPopUp.GradientColor1:=$00FFCC99;
                            MSNPopUp.ShowPopUp;
                        end;

                    // Вивід повідомлення при згадуванні слів зі списку
                    IrcFindText (AUser.Nick,Content,AChannel.Name);

                    // Банити користувачів за використання слів зі списку
                    IrcBotKick (AUser.Nick,Content,AChannel.Name);

                    // Озвучити повідомлення в активному каналі
                    if (IrcSettingsForm.IrcSettings_Sound_Speak_Channel_CheckBox.Checked)
and
                (AnsiUpperCase (Irc_PageControl.ActivePage.Caption)=AnsiUpperCase (AChannel.Name))
then
                    begin
                        IrcSpeak (AUser.Nick, ParseColorString (Content) , True);
                    end;

                    // Озвучити повідомлення у всіх каналах
                    if IrcSettingsForm.IrcSettings_Sound_Speak_Channels_CheckBox.Checked
then
                    begin
                        IrcSpeak (AUser.Nick, ParseColorString (Content) , True);
                    end;

                    // Звук
                    if IrcSettingsForm.IrcSettings_Sound_Message_CheckBox.Checked then
                    begin
                        Wave.WaveName:='message';
                        Wave.Execute;
                    end;

                    // Робота з файлами лога
                    IRCLogs (AChannel.Name,AUser.Nick,Content);
                end;
            end;
        end;
    end;
end;

```

```

        // Іконка повідомлення
        if
AnsiUpperCase(Irc_PageControl.ActivePage.Caption) <> AnsiUpperCase(AChannel.Name)
then
    TTabSheet(FindComponent(Irc_PageControl.Pages[i].Name)).ImageIndex:=5;
    end;
    end;
end;

procedure TForm1.IdIRCKick(Sender: TObject; AUser, AVictim: TIdIRCUser;
    AChannel: TIdIRCChannel);
var
    i      : Integer;
begin
for i:=0 to Irc_PageControl.PageCount-1 do
    begin
        if
AnsiUpperCase(Irc_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
then
        begin
            // Виводимо повідомлення

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'3* '+AUser.Nick+' забанив користувача '+AVictim.Nick+'
('+AVictim.Address+') - '+AUser.Reason);

            // Перевіряємо наявність ім'я користувача в списку
            if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
AVictim.Nick) <> -1 then
                // Видаляємо ім'я користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(A
Victim.Nick));

                // Якщо забанили оператора @
                // Перевіряємо наявність ім'я користувача в списку
                if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'@'+AVictim.Nick) <> -1 then
                    // Видаляємо ім'я користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('
@'+AVictim.Nick));

                    // Якщо забанили Voice +
                    // Перевіряємо наявність ім'я користувача в списку
                    if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'+'+AVictim.Nick) <> -1 then
                        // Видаляємо ім'я користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('
'+AVictim.Nick));

                        // Якщо забанили Хопа %
                        // Перевіряємо наявність ім'я користувача в списку
                        if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'%'+AVictim.Nick) <> -1 then
                            // Видаляємо ім'я користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('
%'+AVictim.Nick));

```







```

TListBox (FindComponent (Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox (FindComponent (Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('
@'+Copy (AModes,m+1,Length (AModes) -m));
//Додаємо користувача в список

TListBox (FindComponent (Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add (Copy
(AModes,m+1,Length (AModes) -m));
end;
end;
except

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing (TAG_COLOR+'3* Режим каналу: '+AModes);
end;
end;
end;

// Невідома команда
procedure TForm1.IdIRCUnknownCommand (Sender: TObject; AUser: TIdIRCUser;
ACommand, AContent: String);
begin
if AnsiUpperCase (ACommand)='ERROR' then
begin
Reconnect (AUser.Nick+' '+ACommand+' '+AContent);
end
else
begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).Lines.Add ('
'+AUser.Nick+' '+ACommand+' '+AContent);
end;
end;

// Ваш статус
procedure TForm1.IdIRCUserMode (Sender: TObject; AModes: String);
begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'3* Ваш користувальницький статус змінений на '+AModes);
end;

// Новий користувач прийшов на канал
procedure TForm1.IdIRCJoin (Sender: TObject; AUser: TIdIRCUser;
AChannel: TIdIRCChannel);
var
i : Integer;
begin
for i:=0 to Irc_PageControl.PageCount-1 do
begin
if
AnsiUpperCase (Irc_PageControl.Pages[i].Caption)=AnsiUpperCase (AChannel.Name)
then
begin
//Виводимо повідомлення
if IrcSettingsForm.IrcSettings_Show_Join_CheckBox.Checked then
begin

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing (TAG_COLOR+'3* '+AUser.Nick+' Заходить ('+AUser.Address+');
end;

// Перевіряємо наявність ім'я користувача в списку
if
TListBox (FindComponent (Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf (
AUser.Nick)=-1 then
// Додаємо користувача в список

```

```

TListBox (FindComponent (Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add(AUser.Nick);

    // Давати +o користувачам зі списку
    IrcBotOp (AUser.Nick,AChannel.Name);

    // Давати +v користувачам зі списку
    IrcBotVoice (AUser.Nick,AChannel.Name);

    // Звук
    if IrcSettingsForm.IrcSettings_Sound_Join_CheckBox.Checked then
    begin
        Wave.WaveName:='join';
        Wave.Execute;
    end;
end;

// Перевірка наявності вікна привата з користувачем
for i:=0 to Irc_PageControl.PageCount-1 do
begin
    if
AnsiUpperCase (Irc_PageControl.Pages[i].Caption)=AnsiUpperCase (AUser.Nick) then
    begin
        // Міняємо іконку
        Irc_PageControl.Pages[i].ImageIndex:=7;
    end;
end;

// Користувач пішов з каналу
procedure TForm1.IdIRCPart (Sender: TObject; AUser: TIdIRCUser;
    AChannel: TIdIRCChannel);
var
    i      : Integer;
begin
for i:=0 to Irc_PageControl.PageCount-1 do
    begin
        if
AnsiUpperCase (Irc_PageControl.Pages[i].Caption)=AnsiUpperCase (AChannel.Name)
then
        begin
            //Виводимо повідомлення
            if IrcSettingsForm.IrcSettings_Show_Part_CheckBox.Checked then
            begin
TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing (TAG_COLOR+'3* '+AUser.Nick+' Вийшов ('+AUser.Address+')');
                end;

                // Перевіряємо наявність ім'я користувача в списку
                if
TListBox (FindComponent (Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf (
AUser.Nick)<>-1 then
                    //Видаляємо користувача зі списку

TListBox (FindComponent (Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete (T
ListBox (FindComponent (Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf (A
User.Nick));

                // Якщо пішов оператор @
                // Перевіряємо наявність ім'я користувача в списку
                if
TListBox (FindComponent (Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf (
'@'+AUser.Nick)<>-1 then
                    //Видаляємо користувача зі списку

TListBox (FindComponent (Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete (T

```

```

ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('@'+AUser.Nick));

    // Якщо пішов Voice +
    // Перевіряємо наявність ім'я користувача в списку
    if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(''+AUser.Nick)<>-1 then
    //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(''+AUser.Nick));

    // Якщо пішов Хоп %
    // Перевіряємо наявність ім'я користувача в списку
    if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('%'+AUser.Nick)<>-1 then
    //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('%'+AUser.Nick));

    // Звук
    if IrcSettingsForm.IrcSettings_Sound_Part_CheckBox.Checked then
        begin
            Wave.WaveName:='part';
            Wave.Execute;
        end;
    end;
end;

// Приховання вкладки при виході з каналу
procedure TForm1.IdIRCParted(Sender: TObject; AChannel: TIdIRCChannel);
var
    i      : Integer;
begin
    for i:=0 to Irc_PageControl.PageCount-1 do
        begin
            if
AnsiUpperCase(Irc_PageControl.Pages[i].Caption)=AnsiUpperCase(AChannel.Name)
then
                begin
                    Irc_PageControl.Pages[i].TabVisible:=false;
                end;
            end;
        end;
end;

// Користувач відключився (QUIT)
procedure TForm1.IdIRCQuit(Sender: TObject; AUser: TIdIRCUser);
var
    i : Integer;
begin
    // Повідомлення в канали де був користувач
    for i:=0 to Irc_PageControl.PageCount-1 do
        begin
            if Irc_PageControl.Pages[i].Caption[1]='#' then
                begin
                    // Перевіряємо наявність ім'я користувача в списку
                    if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(AUser.Nick)<>-1 then
                        begin
                            //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T

```

```

ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(A
User.Nick));
    // Виводимо повідомлення
    if IrcSettingsForm.IrcSettings_Show_Quit_CheckBox.Checked then
        begin

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'2* '+AUser.Nick+' Відключився ('+AUser.Address+' -
'+AUser.Reason);
            end;
        end;

        // Якщо пішов оператор @
        // Перевіряємо наявність ім'я користувача в списку
        if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'@'+AUser.Nick)<>-1 then
            begin
                //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('
@'+AUser.Nick));
                // Виводимо повідомлення
                if IrcSettingsForm.IrcSettings_Show_Quit_CheckBox.Checked then
                    begin

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'2* '+AUser.Nick+' Відключився ('+AUser.Address+' -
'+AUser.Reason);
                    end;
                end;

                // Якщо пішов Voice +
                // Перевіряємо наявність ім'я користувача в списку
                if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'+'+AUser.Nick)<>-1 then
                    begin
                        //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('
'+AUser.Nick));
                        // Виводимо повідомлення
                        if IrcSettingsForm.IrcSettings_Show_Quit_CheckBox.Checked then
                            begin

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'2* '+AUser.Nick+' Відключився ('+AUser.Address+' -
'+AUser.Reason);
                            end;
                        end;

                        // Якщо пішов Хоп %
                        // Перевіряємо наявність ім'я користувача в списку
                        if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'%'+AUser.Nick)<>-1 then
                            begin
                                //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf('
%'+AUser.Nick));
                                // Виводимо повідомлення
                                if IrcSettingsForm.IrcSettings_Show_Quit_CheckBox.Checked then
                                    begin

```

```

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedString(TAG_COLOR+'2* '+AUser.Nick+' Відключився ('+AUser.Address+') - '+AUser.Reason);
    end;
    end;
end;

// Повідомлення у вікно привата з користувачем
for i:=0 to Irc_PageControl.PageCount-1 do
begin
    if AnsiUpperCase(Irc_PageControl.Pages[i].Caption)=AnsiUpperCase(AUser.Nick)
then
    begin
        if IrcSettingsForm.IrcSettings_Show_Quit_CheckBox.Checked then
        begin
            TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedString(TAG_COLOR+'2* '+AUser.Nick+' Відключився ('+AUser.Address+') - '+AUser.Reason);
                end;
                // Мінємо іконку
                Irc_PageControl.Pages[i].ImageIndex:=16;
            end;
        end;
    end;

// Користувач мінє нік
procedure TForm1.IdIRCNickChange(Sender: TObject; AUser: TIdIRCUser;
    ANewNick: String);
var
    i : integer;
begin
    for i:=0 to Irc_PageControl.PageCount-1 do
    begin
        // Зміна ім'я вкладки для привата
        if AnsiUpperCase(Irc_PageControl.Pages[i].Caption)=AnsiUpperCase(AUser.Nick)
then Irc_PageControl.Pages[i].Caption:=ANewNick;

        if Irc_PageControl.Pages[i].Caption[1]='#' then
        begin
            // Якщо перемінив нік звичайний користувач
            // Перевіряємо наявність ім'я користувача в списку
            if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
AUser.Nick)<>-1 then
            begin
                //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(A
User.Nick));
                // Додаємо користувача в список з новим ніком

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add(ANew
Nick);

                // Виводимо повідомлення
                if IrcSettingsForm.IrcSettings_Show_Nick_CheckBox.Checked then
                begin

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'3* '+AUser.Nick+' мінє нік на '+ANewNick);
                    end;
                end;

                // Якщо перемінив нік оператор @
                // Перевіряємо наявність ім'я користувача в списку

```

```

    if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'@'+AUser.Nick)<>-1 then
    begin
        //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'@'+AUser.Nick));
        // Додаємо користувача в список з новим ніком

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add('@'+
ANewNick);
        // Виводимо повідомлення
        if IrcSettingsForm.IrcSettings_Show_Nick_CheckBox.Checked then
            begin

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'3* '+AUser.Nick+' міняє нік на '+ANewNick);
            end;
            end;

        // Якщо перемінив нік Voice +
        // Перевіряємо наявність ім'я користувача в списку
        if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'+'+AUser.Nick)<>-1 then
            begin
                //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'+'+AUser.Nick));
                // Додаємо користувача в список з новим ніком

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add('+'+
ANewNick);
                // Виводимо повідомлення
                if IrcSettingsForm.IrcSettings_Show_Nick_CheckBox.Checked then
                    begin

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'3* '+AUser.Nick+' міняє нік на '+ANewNick);
                    end;
                    end;

                // Якщо перемінив нік Хоп %
                // Перевіряємо наявність ім'я користувача в списку
                if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'%' +AUser.Nick)<>-1 then
                    begin
                        //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
'%' +AUser.Nick));
                        // Додаємо користувача в список з новим ніком

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add('%'+
ANewNick);
                        // Виводимо повідомлення
                        if IrcSettingsForm.IrcSettings_Show_Nick_CheckBox.Checked then
                            begin

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'3* '+AUser.Nick+' міняє нік на '+ANewNick);
                            end;
                            end;
                    end;
            end;

```

```

        end;
    end;
    // Перевірка наявності вікна привата з користувачем
    for i:=0 to Irc_PageControl.PageCount-1 do
        begin
            if AnsiUpperCase(Irc_PageControl.Pages[i].Caption)=AnsiUpperCase(ANewNick)
            then
                begin
                    // Міняємо іконку
                    Irc_PageControl.Pages[i].ImageIndex:=7;
                end;
            end;
        end;

    // Ви міняєте нік
    procedure TForm1.IdIRCNickChanged(Sender: TObject; AOldNick: String);
    var
        i : integer;
    begin
        Irc_Nik_Edit.Text:=IdIRC.Nick;
        for i:=0 to Irc_PageControl.PageCount-1 do
            begin
                if Irc_PageControl.Pages[i].Caption[1]='#' then
                    begin
                        // Якщо ви звичайний користувач
                        // Перевіряємо наявність ім'я користувача в списку
                        if
                            TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
                                AOldNick)<>-1 then
                                begin
                                    //Видаляємо користувача зі списку

                                    TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(
                                        TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
                                            AOldNick));
                                    // Додаємо користувача в список з новим ніком

                                    TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add(IdIRC
                                        .Nick);
                                    // Виводимо повідомлення

                                    TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
                                        ing(TAG_COLOR+'3* Ваш новий нік '+IdIRC.Nick);
                                    end;

                                    // Якщо ви оператор @
                                    // Перевіряємо наявність ім'я користувача в списку
                                    if
                                        TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
                                            '@'+AOldNick)<>-1 then
                                            begin
                                                //Видаляємо користувача зі списку

                                                TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(
                                                    TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
                                                        '@'+AOldNick));
                                                // Додаємо користувача в список з новим ніком

                                                TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add('@'+
                                                    IdIRC.Nick);
                                                // Виводимо повідомлення

                                                TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
                                                    ing(TAG_COLOR+'3* Ваш новий нік '+IdIRC.Nick);
                                                end;

                                                // Якщо ви Voice +
                                                // Перевіряємо наявність ім'я користувача в списку

```

```

    if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
+'+AOldNick)<>-1 then
    begin
        //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
+'+AOldNick));
        // Додаємо користувача в список з новим ніком

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add(''+
IdIRC.Nick);
        // Виводимо повідомлення

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'3* Ваш новий нік '+IdIRC.Nick);
        end;

        // Якщо ви Хоп %
        // Перевіряємо наявність ім'я користувача в списку
        if
TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
+'%'+AOldNick)<>-1 then
            begin
                //Видаляємо користувача зі списку

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Delete(T
ListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.IndexOf(
+'%'+AOldNick));
                // Додаємо користувача в список з новим ніком

TListBox(FindComponent(Irc_PageControl.Pages[i].Name+'_ListBox')).Items.Add('%'+
IdIRC.Nick);
                // Виводимо повідомлення

TIRCMemo(FindComponent(Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'3* Ваш новий нік '+IdIRC.Nick);
                end;
            end;
        end;
    end;

// Нік і AltNick перебувають у використанні.
procedure TForm1.IdIRCNicksInUse(Sender: TObject; var ANick: String);
begin
    IdIRC.Disconnect;

TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Ваші ніки '+IdIRC.Nick+' і '+IdIRC.AltNick+' уже
використовуються іншими користувачами.'+#10#13+'Перемініте нік і підключитися
заново.');
```

```

end;

// Статус
procedure TForm1.IdIRCStatus(ASender: TObject; const AStatus: TIdStatus;
    const AStatusText: String);
begin
    Irc_Server_IRCMemo.Lines.Add(AStatusText);
end;

// Процедура аутентифікація ніка, перезахід на відкриті канали й автозахід на
канали з Вибраного при з'єднанні
procedure TForm1.IdIRCSysyem(Sender: TObject; AUser: TIdIRCUser;
    ACmdCode: Integer; ACommand, AContent: String);
var
    i
        : Integer;
    FavoriteList
        : TStringList;
begin
```

```

if ACmdCode=255 then
begin
  try
    // Аутентифікація ніка
    IdIRC.WriteLine('PRIVMSG NICKSERV :IDENTIFY '+Irc_Password_Edit.Text);
  except
    on e:Exception do Reconnect(E.Message);
  end;
end;

// Перезахід на відкриті канали
if (ACmdCode=255) and (IrcSettingsForm.IrcSettings_Servis_Quit_CheckBox.Checked)
then
begin
  for i:=0 to Irc_PageControl.PageCount-1 do
  begin
    if Irc_PageControl.Pages[i].Caption[1]='#' then
    begin
      try
        IdIRC.Join(Irc_PageControl.Pages[i].Caption);

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'4* Перезахід на канал '+Irc_PageControl.Pages[i].Caption);
      except
        // Помилка
        on e:Exception do
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4'+E.Message);
        end;
      end;
    end;
  end;

// Автозахід при з'єднанні
if (ACmdCode=255) and (IrcFavoriteForm.IrcFavorite_CheckBox.Checked) and
(Irc_PageControl.PageCount=2) then
begin
  FavoriteList:=TStringList.Create;
  try

FavoriteList.LoadFromFile (ExtractFileDir (ParamStr (0))+'\IRC\Favorite\'+Irc_Serve
r_Address_Combobox.Text+'.fav');
    for i:=0 to FavoriteList.Count-1 do
    begin
      IdIRC.Join(FavoriteList.Strings[i]);
    end;
  except
    // Помилка
    on e:Exception do
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4'+E.Message);
    end;
    FavoriteList.Free;
  end;
end;

// Користувач надіслав Вам запит CTCP
procedure TForm1.IdIRCCTCPQuery(Sender: TObject; User: TIdIRCUser;
  AChannel: TIdIRCChannel; Command, Args: String; var ASuppress: Boolean);
begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'12* ['+User.Nick+'] відправив Вам CTCP запит : '+Command+'
'+Args);
end;

// Отримана відповідь на запит CTCP
procedure TForm1.IdIRCCTCPReply(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel; Command, Args: String);

```

```

begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'12* ['+AUser.Nick+'] відповідь на СТСР запит : '+Command+'
'+Args);
end;

// Прийом повідомлень (Notice)
procedure TForm1.IdIRCNotice(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel; Content: String);
var
  i      : Integer;
begin
  try
    for i:=0 to Irc_PageControl.PageCount-1 do
      begin
        if
AnsiUpperCase (Irc_PageControl.Pages[i].Caption)=AnsiUpperCase (AChannel.Name)
then
          begin

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'7* - '+AUser.Nick+ ' - '+Content);
          end;
        end;
      except

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'7* - '+AUser.Nick+ ' - '+Content);
        end;
      end;

// Режим AWAY включений
procedure TForm1.IdIRCNOWAway(Sender: TObject; AContent: String);
begin
  Irc_Away_Status_Label.Visible:=true;
end;

// Назва топика на каналі
procedure TForm1.IdIRCTopic(Sender: TObject; AUser: TIdIRCUser;
  AChannel: TIdIRCChannel; const AChanName, ATopic: String);
var
  i      : Integer;
  LogFile  : TextFile;
  TopicList : TStringList;
begin
  for i:=0 to Irc_PageControl.PageCount-1 do
    begin
      if
AnsiUpperCase (Irc_PageControl.Pages[i].Caption)=AnsiUpperCase (AChannel.Name)
then
        begin

TIRCMemo (FindComponent (Irc_PageControl.Pages[i].Name+'_IRCMemo')).AddFormattedStr
ing(TAG_COLOR+'7* Топик '+AUser.Nick+ ' '+ATopic);

          // Збереження топика у файл

AssignFile (LogFile,ExtractFileDir (ParamStr (0))+'\IRC\Channel_Settings\Topic\'+IdIRC.Hos
t+'\'+AChannel.Name+'.txt');
          if
FileExists (ExtractFilePath (paramstr (0))+'\IRC\Channel_Settings\Topic\'+IdIRC.Hos
t+'\'+AChannel.Name+'.txt')=false then
            begin
              CreateDirectory (PChar (ExtractFileDir (ParamStr (0))+'\IRC'),nil);
CreateDirectory (PChar (ExtractFileDir (ParamStr (0))+'\IRC\Channel_Settings'),nil);

```

```

CreateDirectory(PChar(ExtractFileDir(ParamStr(0))+'\IRC\Channel_Settings\Topic')
, nil);

CreateDirectory(PChar(ExtractFileDir(ParamStr(0))+'\IRC\Channel_Settings\Topic\'
+IdIRC.Host), nil);
    Rewrite(LogFile);
    CloseFile(LogFile);
end;

    TopicList:=TStringList.Create;

TopicList.LoadFromFile(ExtractFilePath(paramstr(0))+'\IRC\Channel_Settings\Topic
\'+IdIRC.Host+'\'+AChannel.Name+'.txt');
    if (TopicList.IndexOf(ATopic)=-1) and (ATopic<>'') then
    begin
        Append(LogFile);
        Writeln(LogFile, ATopic);
        TopicList.Free;
        CloseFile(LogFile);
    end;
end;
end;

// Режим AWAY виключений
procedure TForm1.IdIRCUnAway(Sender: TObject; AContent: String);
begin
    Irc_Away_Status_Label.Visible:=false;
end;

procedure TForm1.IdIRCPingPong(Sender: TObject);
begin
    // Дані про стан пінга
    Irc_Ping_Timer.Enabled:=False;
    Irc_Ping_Timer.Enabled:=True;

    if IrcSettingsForm.IrcSettings_Show_PingPong_CheckBox.Checked then
        Irc_Server_IRCMemo.AddFormattedString(TAG_COLOR+'3* Ping! Pong!
'+TimeToStr(Time));
end;

// помилка сокету
procedure TForm1.IdIRCsocketError(Sender: TObject; ASocket, AMsg: String);
begin

TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormatedS
tring(TAG_COLOR+'4=SocketError='+ASocket+' '+AMsg);
end;

// помилка
procedure TForm1.IdIRCError(Sender: TObject; AUser: TIdIRCUser; ANumeric,
    AError: String);
begin

TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormatedS
tring(TAG_COLOR+'4=Error='+AUser.Nick+' '+ANumeric+' '+AError);
end;

// Користувач запросив вас на канал
procedure TForm1.IdIRCInvite(Sender: TObject; AUser: TIdIRCUser;
    AChannel: String);
begin
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormatedS
tring(TAG_COLOR+'6* '+AUser.Nick+' запрошує вас на канал '+AChannel);

// Заходити в канал при запрошенні
if IrcSettingsForm.IrcSettings_Servis_Invite_CheckBox.Checked then
begin

```

```

try
  IdIRC.Join(AChannel);
except
  // Помилка
  on e:Exception do
TIRCMemo (FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4'+E.Message);
  end;
end;
end;

// Дії при показі вкладок
// Процедура показу вкладки вікна сервера
procedure TForm1.Irc_ServerShow(Sender: TObject);
begin
  Irc_Close_SpeedButton.Enabled:=false;
  Irc_Close_SpeedButton.ShowHint:=false;
  try
    // Міняємо іконку для вікна каналу
    if Irc_PageControl.ActivePage.ImageIndex=5 then
Irc_PageControl.ActivePage.ImageIndex:=4;
    // Міняємо іконку для вікна привата
    if Irc_PageControl.ActivePage.ImageIndex=6 then
Irc_PageControl.ActivePage.ImageIndex:=7;
  except
    end;
end;

// Процедура показу вкладки списку каналів
procedure TForm1.ChannelsShow(Sender: TObject);
begin
  Irc_Close_SpeedButton.Enabled:=true;
  Irc_Close_SpeedButton.ShowHint:=true;
  try
    // Міняємо іконку для вікна каналу
    if Irc_PageControl.ActivePage.ImageIndex=5 then
Irc_PageControl.ActivePage.ImageIndex:=4;
    // Міняємо іконку для вікна привата
    if Irc_PageControl.ActivePage.ImageIndex=6 then
Irc_PageControl.ActivePage.ImageIndex:=7;
  except
    end;
end;

// Процедура показу вкладки каналу
procedure TForm1.Irc_Channel_Show(Sender: TObject);
begin
  Irc_Close_SpeedButton.Enabled:=true;
  Irc_Close_SpeedButton.ShowHint:=true;
  try
    // Міняємо іконку для вікна каналу
    if Irc_PageControl.ActivePage.ImageIndex=5 then
Irc_PageControl.ActivePage.ImageIndex:=4;
    // Міняємо іконку для вікна привата
    if Irc_PageControl.ActivePage.ImageIndex=6 then
Irc_PageControl.ActivePage.ImageIndex:=7;
  except
    end;
end;

// Процедура показу вкладки привата
procedure TForm1.Irc_Privat_Show(Sender: TObject);
begin
  Irc_Close_SpeedButton.Enabled:=true;
  Irc_Close_SpeedButton.ShowHint:=true;
  try
    // Міняємо іконку для вікна каналу
    if Irc_PageControl.ActivePage.ImageIndex=5 then
Irc_PageControl.ActivePage.ImageIndex:=4;

```

```

        // Міняємо іконку для вікна привата
        if Irc_PageControl.ActivePage.ImageIndex=6 then
Irc_PageControl.ActivePage.ImageIndex:=7;
        except
        end;
end;

// Кнопки
// Очистити вікно показу повідомлень
procedure TForm1.Irc_Clear_SpeedButtonClick(Sender: TObject);
begin
    TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).Clear;
end;

// Показати балка каналу
procedure TForm1.Irc_Log_SpeedButtonClick(Sender: TObject);
begin
    ShellExecute
(0, 'open', Pchar (ExtractFileDir (ParamStr (0))+'\IRC\Logs\'+IdIRC.Host+'\'+Irc_Page
Control.ActivePage.Caption+'.txt'), '', '', sw_show);
end;

// Вибране
procedure TForm1.Irc_Favorite_SpeedButtonClick(Sender: TObject);
var
    item          : TMenuItem;
    i              : integer;
    FavoriteList  : TStringList;
begin
for i:=0 to Irc_Favorite_PopupMenu.Items.Count-1 do
    begin
        if Irc_Favorite_PopupMenu.Items.Count-1>2 then
Irc_Favorite_PopupMenu.Items[Irc_Favorite_PopupMenu.Items.Count-1].Free;
        end;
if
FileExists (ExtractFileDir (ParamStr (0))+'\IRC\Favorite\'+Irc_Server_Address_Combo
Box.Text+'.fav')=true then
    begin
        FavoriteList:=TStringList.Create;

FavoriteList.LoadFromFile (ExtractFileDir (ParamStr (0))+'\IRC\Favorite\'+Irc_Serve
r_Address_Combobox.Text+'.fav');
        FavoriteList.Sorted:=true;
        for i:=0 to FavoriteList.Count-1 do
            begin
                item:=TMenuItem.Create (Irc_Favorite_PopupMenu);
                item.Caption:=FavoriteList.Strings[i];
                item.OnClick:=ItemClick;
                Irc_Favorite_PopupMenu.Items.Add (item);
            end;
            FavoriteList.Free;
        end;
Irc_Favorite_PopupMenu.Popup (Mouse.CursorPos.X, Mouse.CursorPos.Y);
end;

// Клік по пункті ім'я каналу меню вибраного й захід на канал
procedure TForm1.ItemClick (sender:Tobject);
begin
if IdIRC.Connected then
    begin
        try
            with sender as TMenuItem do
                begin
                    IdIRC.Join (Caption);
                end;
            except
                // Помилка

```

```

        on e:Exception do
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'4'+E.Message);
        end;
        end;
end;

// Додати у вибране
procedure TForm1.Irc_Favorite_PopupMenu_AddClick(Sender: TObject);
var
    i      : Integer;
    // Робота з файлами Вибраного
    LogFile : TextFile;
begin
for i:=0 to Irc_Favorite_PopupMenu.Items.Count-1 do
    begin
        if
AnsiUpperCase (Irc_PageControl.ActivePage.Caption)=AnsiUpperCase (Irc_Favorite_Pop
upMenu.Items[i].Caption) then exit;
        end;
if Irc_PageControl.ActivePage.Caption[1]='#' then
    begin
        // Робота з файлами Вибраного

AssignFile (LogFile,ExtractFileDir (ParamStr (0))+'\IRC\Favorite\'+'+IdIRC.Host+'.fav
');
        if
FileExists (ExtractFileDir (ParamStr (0))+'\IRC\Favorite\'+'+IdIRC.Host+'.fav')=false
then
            begin
                CreateDirectory (PChar (ExtractFileDir (ParamStr (0))+'\IRC'),nil);
                CreateDirectory (PChar (ExtractFileDir (ParamStr (0))+'\IRC\Favorite'),nil);
                Rewrite (LogFile);
                CloseFile (LogFile);
            end;
            Append (LogFile);
            Writeln (LogFile,Irc_PageControl.ActivePage.Caption);
            CloseFile (LogFile);
        end;
end;

// Упорядкувати вибране
procedure TForm1.Irc_Favorite_PopupMenu_DelClick(Sender: TObject);
begin
    IrcFavoriteForm.ShowModal;
end;

// Команди
procedure TForm1.Irc_Commands_SpeedButtonClick(Sender: TObject);
begin
if IdIRC.Connected then
    begin
        Irc_Commands_PopupMenu.Popup (Mouse.CursorPos.X,Mouse.CursorPos.Y);
    end
    else
        begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'4Ви не підключені до IRC сервера');
        end;
end;

// Вікно властивостей каналу
procedure TForm1.Irc_ChannelSettings_SpeedButtonClick(Sender: TObject);
begin
    if not IdIRC.Connected then
        begin

```

```

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Ви не підключені до IRC сервера');
    exit;
    end;
    if Irc_PageControl.ActivePage.Caption[1]='#' then
    begin
        IrcChannelSettingsForm.Caption:='Властивості каналу
'+Irc_PageControl.ActivePage.Caption;
        IrcChannelSettingsForm.ShowModal;
    end
    else
    begin
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Перемкнетеся на вкладку каналу');
    end;
end;

// Меню списку каналів
procedure TForm1.Irc_Channels_SpeedButtonClick(Sender: TObject);
begin
if IdIRC.Connected then
    begin
        Irc_List_PopupMenu.Popup(Mouse.CursorPos.X,Mouse.CursorPos.Y);
    end
    else
    begin
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Ви не підключені до IRC сервера');
    end;
end;

// Більше 100 користувачів на каналах
procedure TForm1.Irc_List_100_PopupMenuClick(Sender: TObject);
begin
if IdIRC.Connected then
    begin
        Irc_PageControl.ActivePageIndex:=1;
        try
            // Відправляємо повідомлення серверу
            IdIRC.WriteLine('LIST >100');
        except
            on e:Exception do Reconnect(E.Message);
        end;
    end
    else
    begin
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Ви не підключені до IRC сервера');
    end;
end;

// Більше 50 користувачів на каналах
procedure TForm1.Irc_List_50_PopupMenuClick(Sender: TObject);
begin
if IdIRC.Connected then
    begin
        Irc_PageControl.ActivePageIndex:=1;
        try
            // Відправляємо повідомлення серверу
            IdIRC.WriteLine('LIST >50');
        except
            on e:Exception do Reconnect(E.Message);
        end;
    end
    else

```

```

begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Ви не підключені до IRC сервера');
end;
end;

// Всі канали
procedure TForm1.Irc_List_All_PopupMenuClick(Sender: TObject);
begin
if IdIRC.Connected then
begin
Irc_PageControl.ActivePageIndex:=1;
try
// Відправляємо повідомлення серверу
IdIRC.WriteLine('LIST');
except
on e:Exception do Reconnect(E.Message);
end;
end
else
begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Ви не підключені до IRC сервера');
end;
end;

// Показ вікна налаштувань IRC бота
procedure TForm1.Irc_Bot_SpeedButtonClick(Sender: TObject);
begin
IrcBotForm.ShowModal;
end;

// Показ вікна налаштувань IRC
procedure TForm1.Irc_Settings_SpeedButtonClick(Sender: TObject);
begin
IrcSettingsForm.ShowModal;
end;

// Показ загального FAQ по IRC
procedure TForm1.Irc_Faq_SpeedButtonClick(Sender: TObject);
var
Res : TResourceStream;
begin
Res:= TResourceStream.Create(HInstance, 'IRCHELP', 'HTML');
Res.SaveToFile(ExtractFileDir(ParamStr(0))+ '\'+ 'irc_faq.html');
Res.Free;
ShellExecute(Application.handle, 'open', pchar(ExtractFileDir(ParamStr(0))+
'\'+ 'irc_faq.html'),nil,nil,SW_SHOW); //відкрили в браузері за замовчуванням
TIdAntiFreezeBase.Sleep(10000);
// Видаляємо файл irc_faq.html
DeleteFile(ExtractFileDir(ParamStr(0))+ '\'+ 'irc_faq.html');
end;

// Закриття вкладки каналу й вихід з каналу, приховання вкладки списку каналів,
закриття вікна привата
procedure TForm1.Irc_Close_SpeedButtonClick(Sender: TObject);
begin
// Приховуємо вкладку списку каналів
if Irc_PageControl.ActivePage.Name='Channels' then
begin
Channels.TabVisible:=false;
end
else
begin
if Irc_PageControl.ActivePage.Caption[1]='#' then
begin
if IdIRC.Connected then

```

```

begin
  try
    // Відправляємо повідомлення серверу
    IdIRC.WriteLine('PART '+Irc_PageControl.ActivePage.Caption);
  except
    on e:Exception do Reconnect(E.Message);
  end;
end;
// Закриваємо вкладку каналу
Irc_PageControl.ActivePage.Free;
end
else
begin
  // Закриваємо вкладку привата
  Irc_PageControl.ActivePage.Free;
end;
end;
end;

// Відправлення повідомлень у канал і команд серверу
procedure TForm1.Irc_Channels_MessageText_EditKeyPress(Sender: TObject;
  var Key: Char);
var
  i,c      : Integer;
  s        : String;
  // Створення вкладки для привата - команда QUERY
  ip,np    : Integer;
begin
  i:=Pos(' ',TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text);
  s:=Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text)-i);
  c:=Pos(' ',s);
  if (key = #13) and (idIRC.Connected) then
    begin
      if TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text<>' '
      then
        begin
          try
            if
              TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text[1]='/' then
                begin
                  // me (Action)
                  if
                    (AnsiUpperCase(Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text,1, i-1))='/ME') and (Irc_PageControl.ActivePage.Caption[1]='#')then
                      begin
                        IdIRC.WriteLine('PRIVMSG '+Irc_PageControl.ActivePage.Caption+'
: ACTION
'+Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text)-i)+'
');
                        TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(TAG_COLOR+'6* '+IdIRC.Nick+'
'+Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text)-i));

                        // Робота з файлами лога

                        IRCLogs(Irc_PageControl.ActivePage.Caption,IdIRC.Nick,Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text)-i));

                        TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
                      end
                    else

```

```

        // QUIT
        if
AnsiUpperCase(Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit'))
.Text,1, i-1))='/QUIT' then
        begin

IdIRC.Quit(Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Te
xt,i+1,Length(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text
)-i)+' - '+IdIRC.Replies.ClientInfo);
        IdIRC.Disconnect;

TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
        end

        else

        // AWAY
        if
AnsiUpperCase(Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit'))
.Text,1, i-1))='/AWAY' then
        begin
        IdIRC.WriteLine('AWAY
:'+Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text,i+1,L
ength(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text)-i));

away_message:=Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit'))
.Text,i+1,Length(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).T
ext)-i);

TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
        end

        else

        // MSG (Особисте повідомлення)
        if
AnsiUpperCase(Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit'))
.Text,1, i-1))='/MSG' then
        begin
        // Робота з файлами лога

IRCLogs(Irc_PageControl.ActivePage.Caption,IdIRC.Nick,Copy(TEdit(FindComponent(I
rc_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(FindComponent(Irc
_PageControl.ActivePage.Name+'_Edit')).Text)-i));

        // Якщо вкладка привата з такою назвою вже існує, то
відкриваємо її й відправляємо повідомлення
        for ip:=0 to Irc_PageControl.PageCount-1 do
        begin
        if Irc_PageControl.Pages[ip].Caption=Copy(s,0, c-1) then
        begin
        // Відправляємо повідомлення
        IdIRC.WriteLine('PRIVMSG '+Copy(s,0, c-1)+'
:'+Copy(s,c+1,Length(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit'
)).Text)-c+1));

TIRCMemo(FindComponent(Irc_PageControl.Pages[ip].Name+'_IRCMemo')).AddFormattedSt
ring(TAG_COLOR+'1<'+IdIRC.Nick+'>
'+Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Le
ngth(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text)-i));

TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
        // Перемикаємося на нову вкладку
        Irc_PageControl.ActivePage:=Irc_PageControl.Pages[ip];
        end;
        end;

        // Якщо вкладка привата з такою назвою вже існує, то зупиняємо
процедуру

```

```

for np:=0 to Irc_PageControl.PageCount-1 do
begin
  if Irc_PageControl.Pages[np].Caption=Copy(s,0, c-1) then
  begin
    // Зупиняємо процедуру
    Exit;
  end;
end;

// Якщо вкладки немає, то відправляємо повідомлення й
показуємо його текст в активному вікні
IdIRC.WriteLine('PRIVMSG '+Copy(s,0, c-1)+'
:'+Copy(s,c+1,Length(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit'
)).Text)-c+1));

TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'5<'+IdIRC.Nick+'>
'+Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Le
ngth(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text)-i));

TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
end

else

// QUERY (Особисте повідомлення й створення вкладки для привата)
if
AnsiUpperCase(Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit'
)).Text,1, i-1))='/QUERY' then
begin
  // Робота з файлами лога

IRCLogs(Irc_PageControl.ActivePage.Caption,IdIRC.Nick,Copy(TEdit(FindComponent(I
rc_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Length(TEdit(FindComponent(Irc
_PageControl.ActivePage.Name+'_Edit')).Text)-i));

// Якщо вкладка привата з такою назвою вже існує, то
відкриваємо її й відправляємо повідомлення
for ip:=0 to Irc_PageControl.PageCount-1 do
begin
  if Irc_PageControl.Pages[ip].Caption=Copy(s,0, c-1) then
  begin
    // Відправляємо повідомлення
    IdIRC.WriteLine('PRIVMSG '+Copy(s,0, c-1)+'
:'+Copy(s,c+1,Length(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit'
)).Text)-c+1));

TIRCMemo(FindComponent(Irc_PageControl.Pages[ip].Name+'_IRCMemo')).AddFormattedSt
ring(TAG_COLOR+'1<'+IdIRC.Nick+'>
'+Copy(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text,i+1,Le
ngth(TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text)-i));

TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
// Перемикаємося на нову вкладку
Irc_PageControl.ActivePage:=Irc_PageControl.Pages[ip];
end;
end;

// Якщо вкладка привата з такою назвою вже існує, то зупиняємо
процедуру
for np:=0 to Irc_PageControl.PageCount-1 do
begin
  if Irc_PageControl.Pages[np].Caption=Copy(s,0, c-1) then
  begin
    // Зупиняємо процедуру
    Exit;
  end;
end;
end;

```

```

        // Якщо вкладки немає, те створюємо її нову вкладку й додаємо
        в неї повідомлення
        IrcNewPrivatPage(Copy(s,0, c-
1), IdIRC.Nick, Copy(TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit'))
.Text, i+1, Length(TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).T
ext)-i), 7);

        // Відправляємо повідомлення
        IdIRC.WriteLine('PRIVMSG '+Copy(s,0, c-1)+'
:'+Copy(s, c+1, Length(TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit'
)).Text)-c+1));

TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
        // Перемикаємося на нову вкладку
        Irc_PageControl.ActivePageIndex:=Irc_PageControl.PageCount-1;
    end

    else

        // TOPIC
        if
AnsiUpperCase (Copy(TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit'))
.Text, 1, i-1))='/TOPIC' then
            begin
                IdIRC.WriteLine('TOPIC '+Copy(s,0, c-1)+'
:'+Copy(s, c+1, Length(TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit'
)).Text)-c+1));

TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
                end

            else

                // NOTICE
                if
AnsiUpperCase (Copy(TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit'))
.Text, 1, i-1))='/NOTICE' then
                    begin
                        IdIRC.WriteLine('NOTICE '+Copy(s,0, c-1)+'
:'+Copy(s, c+1, Length(TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit'
)).Text)-c+1));

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'7* - '+IdIRC.Nick+' -
'+Copy(TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text, i+1, Le
ngth(TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text)-i));

TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
                    end

                else

                    // CLEAR - очищення вікна
                    if
AnsiUpperCase (TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text
)='/CLEAR' then
                        begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).Clear;

TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
                        end

                    else

                        // HELP
                        if
AnsiUpperCase (TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text
)='/HELP' then
                            begin

```

```

Irc_Faq_SpeedButtonClick(nil);

TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
end

else

// будь-яка інша команда
begin

IdIRC.WriteLine (Copy (TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit'))
.Text, 2, Length (TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text)));

TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
end;
end

else

// повідомлення на канал і приват
begin
IdIRC.WriteLine ('PRIVMSG '+Irc_PageControl.ActivePage.Caption+
:'+TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text);

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'1<'+IdIRC.Nick+'>
'+TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text);

// Робота з файлами лога для каналу й привата

IRCLogs (Irc_PageControl.ActivePage.Caption, IdIRC.Nick, TEdit (FindComponent (Irc_Pa
geControl.ActivePage.Name+'_Edit')).Text);

TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
end;
except
on e:Exception do Reconnect (E.Message);
end;
end;
end;

end;

// Процедура вставки спецтегів у поле введення вікна сервера, каналу й привата
procedure TForm1.Irc_Channels_EditKeyDown (Sender: TObject;
var Key: Word; Shift: TShiftState);
var
C : Char;
begin
if IrcColorForm.Showing=true then IrcColorForm.Close;
C := UpCase (Char (Key));
if (C in ['B', 'U', 'R', 'K', #13]) and (ssCtrl in Shift) then
begin
case C of
'B' :
TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).SelText:=TAG_BOLD;
'U' :
TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).SelText:=TAG_UNDER
LINE;
'R' :
TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).SelText:=TAG_INVER
SE;
'K' : begin

IrcColorForm.Top:=Form1.Top+TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Na
me+'_IRCMemo')).Height+90;
IrcColorForm.Left:=Form1.Left+100;
IrcColorForm.Show;

```

```

TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).SelText :=
TAG_COLOR;
    end;
    #13 : begin
        if (Irc_PageControl.ActivePage.Caption[1]='#') and
(idIRC.Connected) then
            begin
                // me (Action)
                try
                    IdIRC.WriteLine ('PRIVMSG
'+Irc_PageControl.ActivePage.Caption+' : ACTION
'+TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text+' ');
                except
                    on e:Exception do Reconnect (E.Message);
                end;
            end;

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'6* '+IdIRC.Nick+'
'+TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text);

        // Робота з файлами лога

IRCLogs (Irc_PageControl.ActivePage.Caption, IdIRC.Nick, TEdit (FindComponent (Irc_Pa
geControl.ActivePage.Name+'_Edit')).Text);

TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text:='';
    end;
    end;
    Key := 0;
end;
end;

// Перехід на Edit при відпусканні миші
procedure TForm1.Irc_Server_IRCMemoMouseUp (Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
    Form1.ActiveControl:=TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit'
));

TEdit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).SelStart:=Length (T
Edit (FindComponent (Irc_PageControl.ActivePage.Name+'_Edit')).Text);
end;

// Процедура кліку за списком користувачів на каналі
procedure TForm1.Irc_Channels_ListBoxClick (Sender: TObject);
begin
    if
TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
1 then

TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Hint:=TListB
ox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings [TLis
tBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex];
end;

// Процедура показу кількості користувачів на каналу
procedure TForm1.Irc_Channel_ListBoxMouseMove (Sender: TObject;
    Shift: TShiftState; X, Y: Integer);
begin
    // Кількість користувачів у каналі

TSplitter (FindComponent (Irc_PageControl.ActivePage.Name+'_Splitter')).Hint:='Коп
истувачів на каналі
'+IntToStr (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).I
tems.Count);

```

```

end;

// PopOp меню списку користувачів на каналі

// Додати нік у поле уведення
procedure TForm1.Irc_Channels_ListBox_PopupMenu_AddNickClick(
  Sender: TObject);
begin
  if
  TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
  1 then
  begin
    // Якщо перший символ у ніку рівний + % @ те додаємо нік бех першого символу -
    інакше просто додаємо нік
    if
    (TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
    gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
    ][1]='+') or

    (TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
    gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
    ][1]='%') or

    (TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
    gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
    ][1]='@') then

    TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text:=TEdit(FindCo
    mponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text+Copy(TListBox(FindCompone
    nt(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindCompo
    nent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex],2,Length(TListBox(F
    indComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox
    (FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]))+' '
    else

    TEdit(FindComponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text:=TEdit(FindCo
    mponent(Irc_PageControl.ActivePage.Name+'_Edit')).Text+TListBox(FindComponent(Ir
    c_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(
    Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]+' ';
    end
  else
  TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormatedS
  tring(TAG_COLOR+'4Виберіть користувача зі списку');
  end;

// Whois
procedure TForm1.Irc_Channels_ListBox_PopupMenu_WhoisClick(
  Sender: TObject);
begin
  if
  TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
  1 then
  begin
    try
    if
    (TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
    gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
    ][1]='+') or

    (TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
    gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
    ][1]='%') or

    (TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
    gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
    ][1]='@') then
      IdIRC.WriteLine('WHOIS
'+Copy(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Ite

```

```

mIndex], 2, Length (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
x')).Items.Strings[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_List
Box')).ItemIndex]))
    else
        IdIRC.WriteLine ('WHOIS
'+TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Stri
ngs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x]);
    except
        on e:Exception do Reconnect (E.Message);
    end;
end
else
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Приват
procedure TForm1.Irc_Channels_ListBox_PopupMenu_PrivatClick(
    Sender: TObject);
var
    i      : Integer;
    s      : String;
begin
    if
TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
1 then
        begin
            // Одержуємо назву вкладки привата
            if
(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='+') or

(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='%') or

(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='@') then

s:=Copy (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Item
s.Strings[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).It
emIndex], 2, Length (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_Lis
tBox')).Items.Strings[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_Lis
tBox')).ItemIndex]))
            else

s:=TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Str
ings[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemInd
ex];

            // Якщо вкладка привата з такою назвою вже існує, то відкриваємо її й
зупиняємо процедуру
            for i:=0 to Irc_PageControl.PageCount-1 do
                begin
                    if Irc_PageControl.Pages[i].Caption=s then
                        begin
                            // Перемикаємося на нову вкладку
                            Irc_PageControl.ActivePage:=Irc_PageControl.Pages[i];
                            // Зупиняємо процедуру
                            Exit;
                        end;
                    end;
                end;

            // Якщо вкладки немає, те створюємо її нову вкладку
            IrcNewPrivatPage (s, '', '', 7);

```

```

// Перемикаємося на нову вкладку
Irc_PageControl.ActivePageIndex:=Irc_PageControl.PageCount-1;
end
else

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Оп
procedure TForm1.Irc_Channels_ListBox_PopupMenu_Control_OpClick(
  Sender: TObject);
begin
  if
  TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
  1 then
  begin
  try
  if
  (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
  gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
  ][1]='+') or

  (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
  gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
  ][1]='%') or

  (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
  gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
  ][1]='@') then
    IdIRC.WriteLine ('MODE '+Irc_PageControl.ActivePage.Caption+' +o
  '+Copy (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items
  .Strings [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Ite
  mIndex], 2, Length (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
  x')).Items.Strings [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
  x')).ItemIndex]))
    else
    IdIRC.WriteLine ('MODE '+Irc_PageControl.ActivePage.Caption+' +o
  '+TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
  gs [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemInde
  x]);
  except
  on e:Exception do Reconnect (E.Message);
  end;
  end
  else
  TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
  tring (TAG_COLOR+'4Виберіть користувача зі списку');
  end;

// Керування . Деоп
procedure TForm1.Irc_Channels_ListBox_PopupMenu_Control_DeOpClick (Sender:
  TObject);
begin
  if
  TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
  1 then
  begin
  try
  if
  (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
  gs [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
  ][1]='+') or

  (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
  gs [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
  ][1]='%') or

```

```

(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='@') then
    IdIRC.WriteLine('MODE '+Irc_PageControl.ActivePage.Caption+' -o
'+Copy(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex],2,Length(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]))
    else
        IdIRC.WriteLine('MODE '+Irc_PageControl.ActivePage.Caption+' -o
'+TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]);
except
    on e:Exception do Reconnect(E.Message);
end;
end
else
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Полуоп
procedure TForm1.Irc_Channels_ListBox_PopupMenu_Control_PoluOpClick(
    Sender: TObject);
begin
if
TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-1 then
begin
try
    if
(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='+') or

(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='%') or

(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex][1]='@') then
        IdIRC.WriteLine('MODE '+Irc_PageControl.ActivePage.Caption+' +h
'+Copy(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex],2,Length(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]))
        else
            IdIRC.WriteLine('MODE '+Irc_PageControl.ActivePage.Caption+' +h
'+TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex]);
except
    on e:Exception do Reconnect(E.Message);
end;
end
else
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedString(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Забрати полуопа
procedure TForm1.Irc_Channels_ListBox_PopupMenu_Control_DePoluOpClick(
    Sender: TObject);
begin

```

```

if
TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
1 then
begin
try
if
(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='+') or

(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='%') or

(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='@') then
    IdIRC.WriteLine ('MODE '+Irc_PageControl.ActivePage.Caption+' -h
'+Copy (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Ite
mIndex], 2, Length (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
x')).Items.Strings[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_List
Box')).ItemIndex]))
    else
    IdIRC.WriteLine ('MODE '+Irc_PageControl.ActivePage.Caption+' -h
'+TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Stri
ngs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x]);
except
on e:Exception do Reconnect (E.Message);
end;
end
else
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Войс
procedure TForm1.Irc_Channels_ListBox_PopupMenu_Control_VoiceClick (Sender:
TObject);
begin
if
TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
1 then
begin
try
if
(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='+') or

(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='%') or

(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='@') then
    IdIRC.WriteLine ('MODE '+Irc_PageControl.ActivePage.Caption+' +v
'+Copy (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Ite
mIndex], 2, Length (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
x')).Items.Strings[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_List
Box')).ItemIndex]))
    else
    IdIRC.WriteLine ('MODE '+Irc_PageControl.ActivePage.Caption+' +v
'+TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Stri
ngs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x]);

```

```

except
  on e:Exception do Reconnect(E.Message);
end;
end
else
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Девойс
procedure TForm1.Irc_Channels_ListBox_PopupMenu_Control_DeVoiceClick(Sender:
TObject);
begin
if
TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
1 then
begin
try
if
(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='+') or

(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='%') or

(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='@') then
  IdIRC.WriteLine ('MODE '+Irc_PageControl.ActivePage.Caption+' -v
'+Copy (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Ite
mIndex], 2, Length (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
x')).Items.Strings[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex]))
  else
  IdIRC.WriteLine ('MODE '+Irc_PageControl.ActivePage.Caption+' -v
'+TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
]);
except
  on e:Exception do Reconnect (E.Message);
end;
end
else
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Керування . Бан
procedure TForm1.Irc_Channels_ListBox_PopupMenu_Control_KickClick(Sender:
TObject);
begin
if
TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
1 then
begin
try
if
(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='+') or

(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='%') or

(TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin

```

```

gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='@') then
    IdIRC.WriteLine('KICK '+Irc_PageControl.ActivePage.Caption+
'+Copy(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Ite
mIndex],2,Length(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBo
x')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_List
Box')).ItemIndex]))
    else
        IdIRC.WriteLine('KICK '+Irc_PageControl.ActivePage.Caption+
'+TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Stri
ngs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x]);
except
    on e:Exception do Reconnect(E.Message);
end;
end
else
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormatedS
tring(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Бан
procedure TForm1.Irc_Channels_ListBox_PopupMenu_Control_BanClick(
    Sender: TObject);
begin
if
TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
1 then
begin
try
if
(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='+') or

(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='%') or

(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='@') then
    IdIRC.WriteLine('MODE '+Irc_PageControl.ActivePage.Caption+ ' +b
'+Copy(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Ite
mIndex],2,Length(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBo
x')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_List
Box')).ItemIndex]))
    else
        IdIRC.WriteLine('MODE '+Irc_PageControl.ActivePage.Caption+ ' +b
'+TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Stri
ngs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x]);
except
    on e:Exception do Reconnect(E.Message);
end;
end
else
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormatedS
tring(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// CTCP запити
procedure TForm1.Irc_Channels_ListBox_PopupMenu_CTCP_Click(
    Sender: TObject);
var
    s : String;
begin

```

```

with sender as TMenuItem do s:=Caption;
if
TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
1 then
begin
try
if
(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='+') or

(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='%') or

(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='@') then
    IdIRC.WriteLine('PRIVMSG
'+Copy(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Ite
mIndex],2,Length(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBo
x')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_List
Box')).ItemIndex]))+' :[] '+AnsiUpperCase(s)+'[] ')
    else
    IdIRC.WriteLine('PRIVMSG
'+TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Stri
ngs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x]+' :[] '+AnsiUpperCase(s)+'[] ');
except
on e:Exception do Reconnect(E.Message);
end;
end
else
TIRCMemo(FindComponent(Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormatedS
tring(TAG_COLOR+'4Виберіть користувача зі списку');
end;

// Action
procedure TForm1.Irc_Channels_ListBox_PopupMenu_Action_Click(Sender: TObject);
var
s : String;
begin
with sender as TMenuItem do s:=Caption;
if
TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex<>-
1 then
begin
try
if
(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='+') or

(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='%') or

(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Strin
gs[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).ItemIndex
][1]='@') then
    begin
    IdIRC.WriteLine('PRIVMSG '+Irc_PageControl.ActivePage.Caption+' :[] ACTION
'+s+'
'+Copy(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBox')).Ite
mIndex],2,Length(TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_ListBo
x')).Items.Strings[TListBox(FindComponent(Irc_PageControl.ActivePage.Name+'_List
Box')).ItemIndex]))+'[] ');

```

```

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'6* '+IdIRC.Nick+' '+s+'
'+Copy (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
x')).Items.Strings [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_List
Box')).ItemIndex])));
    // Робота з файлами лога
    IRCLogs (Irc_PageControl.ActivePage.Caption, IdIRC.Nick, s+'
'+Copy (TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items
.Strings [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
x')).Items.Strings [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_List
Box')).ItemIndex])));
    end
    else
    begin
        IdIRC.WriteLine ('PRIVMSG '+Irc_PageControl.ActivePage.Caption+' :[] ACTION
'+s+'
'+TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Stri
ngs [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).ItemInde
x]+'[] ');

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'6* '+IdIRC.Nick+' '+s+'
'+TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Stri
ngs [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex]);
    // Робота з файлами лога
    IRCLogs (Irc_PageControl.ActivePage.Caption, IdIRC.Nick, s+'
'+TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBox')).Items.Stri
ngs [TListBox (FindComponent (Irc_PageControl.ActivePage.Name+'_ListBo
x')).ItemIndex]);
    end;
except
    on e:Exception do Reconnect (E.Message);
end;
end
else
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'4Виберіть користувача зі списку');
end;

// PopUp меню списку каналів
// Увійти на канал
procedure TForm1.Go_To_ChannelClick (Sender: TObject);
var
    i : Integer;
begin
    if Channels_ListBox.ItemIndex=-1 then exit;
    i:=Pos (' ', Channels_ListBox.Items.Strings [Channels_ListBox.ItemIndex]);
    try
        IdIRC.Join (Copy (Channels_ListBox.Items.Strings [Channels_ListBox.ItemIndex], 1,
i-1));
    except
        // Помилка
        on e:Exception do
TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring (TAG_COLOR+'4'+E.Message);
end;
end;

// Клік за списком каналів
procedure TForm1.Channels_ListBoxClick (Sender: TObject);
begin
    if Channels_ListBox.ItemIndex<>-1 then

```

```

Channels_ListBox.Hint:=ParseColorString(Channels_ListBox.Items.Strings[Channels_
ListBox.ItemIndex]);
end;

// DCC
// Підтвердження докачки файлу
procedure TForm1.IdIRCDCCAccept(Sender: TObject; ANick, APort, AFileName,
  APosition: String);
begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'1IdIRCDCCAccept '+'ANick='+ANick+' APort='+APort+'
AFileName='+AFileName+' APosition='+APosition);
end;

// Користувач хоче поспілкуватися з Вами
procedure TForm1.IdIRCDCCChat(Sender: TObject; ANick, AIP, APort: String);
var
  // Переводимо IP з DWORD у нормальний вид
  ip1,ip2,ip3,ip4 : Integer;
  ip                : String;
begin
  // Переводимо IP з DWORD у нормальний вид
  ip4:=StrToInt(AIP) div 16777216;
  ip3:=StrToInt(AIP) mod 16777216 div 65536;
  ip2:=StrToInt(AIP) mod 16777216 mod 65536 div 256;
  ip1:=StrToInt(AIP) mod 16777216 mod 65536 mod 256;
  ip:=IntToStr(ip4)+'.'+IntToStr(ip3)+'.'+IntToStr(ip2)+'.'+IntToStr(ip1);

  IrcDCCForm.Caption:='DCC чат';
  IrcDCCForm.IrcDCC_Label.Caption:='Користувач хоче поспілкуватися з вами через
DCC чат';
  IrcDCCForm.IrcDCC_User_Label_.Caption:=ANick;
  IrcDCCForm.IrcDCC_IP_Label_.Caption:=ip;
  IrcDCCForm.IrcDCC_IP_Label_.Hint:=AIP;
  IrcDCCForm.IrcDCC_Port_Label_.Caption:=APort;
  IrcDCCForm.IrcDCC_FileName_Label.Visible:=false;
  IrcDCCForm.IrcDCC_FileName_Label_.Visible:=false;
  IrcDCCForm.IrcDCC_FileSize_Label.Visible:=false;
  IrcDCCForm.IrcDCC_FileSizeLabel.Visible:=false;
  IrcDCCForm.IrcDCC_FileSize_Label_.Visible:=false;
  IrcDCCForm.IrcDCC_Gauge.Visible:=false;
  IrcDCCForm.IrcDCC_OK_Button.Top:=80;
  IrcDCCForm.IrcDCC_Close_Button.Top:=80;
  IrcDCCForm.Height:=163;
  IrcDCCForm.IrcDCC_OK_Button.Enabled:=true;
  IrcDCCForm.IrcDCC_Close_Button.Caption:='Скасування';
  IrcDCCForm.IrcDCC_StatusBar.Panels[0].Text:='';
  IrcDCCForm.Show;
end;

// Запит на докачку файлу
procedure TForm1.IdIRCDCCResume(Sender: TObject; ANick, APort, AFileName,
  APosition: String);
begin

TIRCMemo (FindComponent (Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFormattedS
tring(TAG_COLOR+'1IdIRCDCCResume '+'ANick='+ANick+' APort='+APort+'
AFileName='+AFileName+' APosition='+APosition);
end;

// Користувач намагається послати вам файл
// Розмір файлу рядком
function BytesToStr(const i64Size: Int64): string;
const
  i64GB = 1024 * 1024 * 1024;
  i64MB = 1024 * 1024;
  i64KB = 1024;

```

```

begin
  if i64Size div i64GB > 0 then
    Result := Format('%.2f GB', [i64Size / i64GB])
  else if i64Size div i64MB > 0 then
    Result := Format('%.2f MB', [i64Size / i64MB])
  else if i64Size div i64KB > 0 then
    Result := Format('%.2f KB', [i64Size / i64KB])
  else
    Result := IntToStr(i64Size) + ' Byte(s)';
end;

procedure TForm1.IdIRCDCCSend(Sender: TObject; ANick, AIP, APort,
  AFileName, AFileSize: String);
var
  // Переводимо IP з DWORD у нормальний вид
  ip1, ip2, ip3, ip4 : Integer;
  ip : String;
begin
  // Переводимо IP з DWORD у нормальний вид
  ip4:=StrToInt(AIP) div 16777216;
  ip3:=StrToInt(AIP) mod 16777216 div 65536;
  ip2:=StrToInt(AIP) mod 16777216 mod 65536 div 256;
  ip1:=StrToInt(AIP) mod 16777216 mod 65536 mod 256;
  ip:=IntToStr(ip4)+'.'+IntToStr(ip3)+'.'+IntToStr(ip2)+'.'+IntToStr(ip1);

  IrcDCCForm.Caption:='DCC прийом файлу';
  IrcDCCForm.IrcDCC_Label.Caption:='Користувач хоче відправити вам файл через
DCC';
  IrcDCCForm.IrcDCC_User_Label.Caption:=ANick;
  IrcDCCForm.IrcDCC_IP_Label.Caption:=ip;
  IrcDCCForm.IrcDCC_IP_Label.Hint:=AIP;
  IrcDCCForm.IrcDCC_Port_Label.Caption:=APort;
  IrcDCCForm.IrcDCC_FileName_Label.Caption:=AFileName;
  IrcDCCForm.IrcDCC_FileName_Label.Hint:=AFileName;
  IrcDCCForm.IrcDCC_FileSize_Label.Caption:=AFileSize;

  IrcDCCForm.IrcDCC_FileSizeLabel.Left:=136+IrcDCCForm.IrcDCC_FileSize_Label_.Width+10;
  IrcDCCForm.IrcDCC_FileSizeLabel.Caption:='(
'+BytesToStr(StrToInt(AFileSize))+' )';
  IrcDCCForm.IrcDCC_FileName_Label.Visible:=true;
  IrcDCCForm.IrcDCC_FileName_Label_.Visible:=true;
  IrcDCCForm.IrcDCC_FileSize_Label.Visible:=true;
  IrcDCCForm.IrcDCC_FileSizeLabel.Visible:=true;
  IrcDCCForm.IrcDCC_FileSize_Label_.Visible:=true;
  IrcDCCForm.IrcDCC_Gauge.Visible:=true;
  IrcDCCForm.IrcDCC_OK_Button.Top:=152;
  IrcDCCForm.IrcDCC_Close_Button.Top:=152;
  IrcDCCForm.Height:=237;
  IrcDCCForm.IrcDCC_Gauge.Progress:=0;
  IrcDCCForm.IrcDCC_OK_Button.Enabled:=true;
  IrcDCCForm.IrcDCC_Close_Button.Caption:='Скасування';
  IrcDCCForm.IrcDCC_StatusBar.Panels[0].Text:='';
  IrcDCCForm.Show;
end;

// Команди
// Включити режим AWAY
procedure TForm1.Irc_Commands_PopupMenu_Away_OnClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Включити режим AWAY';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть текст повідомлення';
  IrcCommandsForm.Irc_Commands_Edit.Text:='';
  IrcCommandsForm.ShowModal;
end;

// Виключити режим AWAY
procedure TForm1.Irc_Commands_PopupMenu_Away_OffClick(Sender: TObject);
begin

```

```

try
  IdIRC.WriteLine('AWAY');
except
  on e:Exception do Reconnect(E.Message);
end;
end;

// Зайти в канал
procedure TForm1.Irc_Commands_PopupMenu_JoinClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Зайти в канал';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть ім'я каналу (і пароль якщо необхідно)';
  IrcCommandsForm.Irc_Commands_Edit.Text:='#';
  IrcCommandsForm.ShowModal;
end;

// Покинути канал
procedure TForm1.Irc_Commands_PopupMenu_PartClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Покинути канал';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть ім'я каналу';
  if Irc_PageControl.ActivePage.Caption[1]='#' then
    IrcCommandsForm.Irc_Commands_Edit.Text:=Irc_PageControl.ActivePage.Caption
  else
    IrcCommandsForm.Irc_Commands_Edit.Text:='#';
  IrcCommandsForm.ShowModal;
end;

// Послати приват
procedure TForm1.Irc_Commands_PopupMenu_MsgClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Послати приват';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть нік користувача й текст повідомлення';
  IrcCommandsForm.Irc_Commands_Edit.Text:='';
  IrcCommandsForm.ShowModal;
end;

procedure TForm1.Irc_Commands_PopupMenu_NoticeClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Послати Notice';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть нік користувача або ім'я каналу й текст повідомлення';
  IrcCommandsForm.Irc_Commands_Edit.Text:='';
  IrcCommandsForm.ShowModal;
end;

// Послати повідомлення від третьої особи
procedure TForm1.Irc_Commands_PopupMenu_MeClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Послати Action';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть ім'я каналу й текст повідомлення';
  if Irc_PageControl.ActivePage.Caption[1]='#' then
    IrcCommandsForm.Irc_Commands_Edit.Text:=Irc_PageControl.ActivePage.Caption+' '
  else
    IrcCommandsForm.Irc_Commands_Edit.Text:='#';
  IrcCommandsForm.ShowModal;
end;

// Whois користувача (online)
procedure TForm1.Irc_Commands_PopupMenu_WhoisClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Whois користувача (online)';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть нік користувача';
  IrcCommandsForm.Irc_Commands_Edit.Text:='';
  IrcCommandsForm.ShowModal;
end;

```

```

// Whowas користувача (offline)
procedure TForm1.Irc_Commands_PopupMenu_WhowasClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Whowas користувача (offline)';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть нік користувача';
  IrcCommandsForm.Irc_Commands_Edit.Text:='';
  IrcCommandsForm.ShowModal;
end;

// Перемінити нік
procedure TForm1.Irc_Commands_PopupMenu_NickClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Перемінити нік';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть новий нік';
  IrcCommandsForm.Irc_Commands_Edit.Text:='';
  IrcCommandsForm.ShowModal;
end;

// Послати CTCP запити
procedure TForm1.Irc_Commands_PopupMenu_CTCP_Click(Sender: TObject);
var
  s : String;
begin
  with sender as TMenuItem do s:=Caption;
  IrcCommandsForm.Caption:='Послати CTCP запит '+s;
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть нік користувача або ім'я каналу';
  IrcCommandsForm.Irc_Commands_Edit.Text:='';
  IrcCommandsForm.ShowModal;
end;

// Одержати інформацію про канал
procedure TForm1.Irc_Commands_PopupMenu_ChannelInfoClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Одержати інформацію про канал';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть ім'я каналу';
  if Irc_PageControl.ActivePage.Caption[1]='#' then
    IrcCommandsForm.Irc_Commands_Edit.Text:=Irc_PageControl.ActivePage.Caption
  else
    IrcCommandsForm.Irc_Commands_Edit.Text:='#';
  IrcCommandsForm.ShowModal;
end;

// Одержати список серверів мережі
procedure TForm1.Irc_Commands_PopupMenu_LinksClick(Sender: TObject);
begin
  try
    IdIRC.WriteLine('LINKS');
  except
    on e:Exception do Reconnect(E.Message);
  end;
end;

// Одержати дані админа сервера
procedure TForm1.Irc_Commands_PopupMenu_AdminClick(Sender: TObject);
begin
  try
    IdIRC.WriteLine('ADMIN');
  except
    on e:Exception do Reconnect(E.Message);
  end;
end;

// Зареєструвати нік
procedure TForm1.Irc_Commands_PopupMenu_RegNickClick(Sender: TObject);
begin
  IrcCommandsForm.Caption:='Зареєструвати нік';
  IrcCommandsForm.Irc_Commands_GroupBox.Caption:='Уведіть пароль і E-mail';

```



```

    then begin
        Inc(Current, 3);
    end else begin
        Inc(Current, 2);
    end;
end else begin
    Inc(Current);
end;
if Phrase[Current] = ','
then begin
    if Phrase[Current + 1] in ['0'..'9']
    then begin
        if Phrase[Current + 2] in ['0'..'9']
        then begin
            Inc(Current, 3);
        end else begin
            Inc(Current, 2);
        end;
    end else begin
        Inc(Current);
    end;
end;
Continue;
end;
#15:
begin
    Channels_ListBox.Canvas.Font := Channels_ListBox.Font;
    Inc(Current);
    Continue;
end;
#22:
begin
    Toggle(fsItalic);
    Inc(Current);
    Continue;
end;
#31:
begin
    Toggle(fsUnderline);
    Inc(Current);
    Continue;
end;
#32:    begin
    NextSpace := Current;
end;
end;
Inc(Width, Channels_ListBox.Canvas.TextWidth(Phrase[Current]));
if Width > WidthMax
then begin
    if LastSpace > 0
    then begin
        Current := LastSpace;
    end else begin
        Dec(Current);
    end;
    NextSpace := -1;
    Inc(Result, 17);
    x := 3;
    Width := x;
end;
LastSpace := NextSpace;
Inc(Current);
end;
end;

procedure TForm1.Channels_ListBoxMeasureItem(Control: TWinControl;
    Index: Integer; var Height: Integer);
begin
    Height := CheckItemSize(Index);

```

```

end;

procedure TForm1.Channels_ListBoxDrawItem(Control: TWinControl;
  Index: Integer; Rect: TRect; State: TOwnerDrawState);
procedure Toggle(FontStyle: TFontStyle);
begin
  if FontStyle in Channels_ListBox.Canvas.Font.Style
  then begin
    Channels_ListBox.Canvas.Font.Style := Channels_ListBox.Canvas.Font.Style -
[FontStyle];
  end else begin
    Channels_ListBox.Canvas.Font.Style := Channels_ListBox.Canvas.Font.Style +
[FontStyle];
  end;
end;
end;
var
  y: Integer;
  x: Integer;
  DefaultColor: TColor;
  Phrase: String;
  Current: Integer;
  NextSpace: Integer;
  LastSpace: Integer;
  LastPrint: Integer;
  WidthMax: Integer;
  Width: Integer;
  Size: Integer;
procedure Flush(Skip: Integer = 0);
var
  Count: Integer;
  Text: String;
begin
  Count := Current - LastPrint + 1 - Skip;
  if Count > 0
  then begin
    Text := Copy(Phrase, LastPrint, Count);
    Channels_ListBox.Canvas.TextOut(x, y, Text);
    Inc(x, Channels_ListBox.Canvas.TextWidth(Text));
    Inc(LastPrint, Count);
  end;
end;
end;
begin
  Size := CheckItemSize(Index);
  if Size <> Channels_ListBox.Perform(LB_GETITEMHEIGHT, Index, 0)
  then begin
    Channels_ListBox.Perform(LB_SETITEMHEIGHT, Index, Size);
    Channels_ListBox.Invalidate;
    Exit;
  end;
  y := Rect.Top + 2;
  x := Rect.Left + 3;
  if odSelected in State
  then begin
    DefaultColor := $00FFCECE;
  end else begin
    DefaultColor :=Channels_ListBox.Color;
  end;
  Channels_ListBox.Canvas.Font :=Channels_ListBox.Font;
  Channels_ListBox.Canvas.Brush.Color := DefaultColor;
  Channels_ListBox.Canvas.FillRect(Rect);
  Phrase :=Channels_ListBox.Items[Index];
  Current := 1;
  NextSpace := -1;
  LastSpace := -1;
  LastPrint := 1;
  WidthMax :=Channels_ListBox.ClientWidth - GetSystemMetrics(SM_CXVSCROLL);
  Width := x;
  Size := Length(Phrase);
  while Current <= Size

```

```

do begin
  case Phrase[Current] of
    #02:
    begin
      Flush(1);
      Toggle(fsBold);
      Inc(Current);
      LastPrint := Current;
      Continue;
    end;
    #03:
    begin
      Flush(1);
      if Phrase[Current + 1] in ['0'..'9']
      then begin
        if Phrase[Current + 2] in ['0'..'9']
        then begin
          Channels_ListBox.Canvas.Font.Color := Colors[StrToInt(Copy(Phrase,
Current + 1, 2)) mod 16];
          Inc(Current, 3);
          LastPrint := Current;
        end else begin
          Channels_ListBox.Canvas.Font.Color := Colors[StrToInt(Phrase[Current
+ 1]) mod 16];
          Inc(Current, 2);
          LastPrint := Current;
        end;
      end else begin
        Channels_ListBox.Canvas.Font.Color :=Channels_ListBox.Font.Color;
        Channels_ListBox.Canvas.Brush.Color := DefaultColor;
        Inc(Current);
        LastPrint := Current;
      end;
      if Phrase[Current] = ','
      then begin
        if Phrase[Current + 1] in ['0'..'9']
        then begin
          if Phrase[Current + 2] in ['0'..'9']
          then begin
            Channels_ListBox.Canvas.Brush.Color :=
Colors[StrToInt(Copy(Phrase, Current + 1, 2)) mod 16];
            Inc(Current, 3);
            LastPrint := Current;
          end else begin
            Channels_ListBox.Canvas.Brush.Color :=
Colors[StrToInt(Phrase[Current + 1]) mod 16];
            Inc(Current, 2);
            LastPrint := Current;
          end;
        end else begin
          Channels_ListBox.Canvas.Brush.Color := DefaultColor;
          Inc(Current);
          LastPrint := Current;
        end;
      end;
      Continue;
    end;
    #15:
    begin
      Flush(1);
      Channels_ListBox.Canvas.Font :=Channels_ListBox.Font;
      Channels_ListBox.Canvas.Brush.Color := DefaultColor;
      Inc(Current);
      LastPrint := Current;
      Continue;
    end;
    #22:
    begin
      Flush(1);

```

```

Toggle(fsItalic);
Inc(Current);
LastPrint := Current;
Continue;
end;
#31:
begin
Flush(1);
Toggle(fsUnderline);
Inc(Current);
LastPrint := Current;
Continue;
end;
#32:
begin
NextSpace := Current;
end;
end;
Inc(Width, Channels_ListBox.Canvas.TextWidth(Phrase[Current]));
if Width > WidthMax
then begin
if LastSpace > 0
then begin
Current := LastSpace;
end else begin
Dec(Current);
end;
Flush;
NextSpace := -1;
Inc(y, Channels_ListBox.ItemHeight);
x := Rect.Left + 3;
Width := x;
end;
LastSpace := NextSpace;
Inc(Current);
end;
Flush;
if odFocused in State
then begin
Channels_ListBox.Canvas.DrawFocusRect(Rect);
end;
end;
//////////
// Дії при відкритті програми
procedure TForm1.FormCreate(Sender: TObject);
var
// Читання налаштувань із ini файлу
Ini : Tinifile;
// Розшифровка пароля IRC
s : String[255];
c : array[0..255] of Byte absolute s;
i : Integer;
// Говорилка
NumFound : DWord;
ModeInfo : TTSModeInfo;
begin
//////////
// Заносимо дані у вікно сервера IRC
Irc_Server_IRCMemo.AddFormattedString('
8,8_____ 0,0_ /join #имя_канала - Зайти
в канал');
Irc_Server_IRCMemo.AddFormattedString('
8,8_ 0,0_ 1,1_ 0,0_ 1,1_
0,0_ 1,1_ 0,0_ 1,1_ 0,0_ 1,1_ 0,0_ 8,8_ 0,0_
/nick ваш_новий_нік - Перемінити нік');
Irc_Server_IRCMemo.AddFormattedString('
8,8_ 0,0_ 4,4_ 1,1_ 0,0_ 4,4_
1,1_ 0,0_ 4,4_ 0,0_ 4,4_ 1,1_ 0,0_ 4,4_ 1,1_ 0,0_
8,8_ 0,0_ /list - Показати список каналів');
Irc_Server_IRCMemo.AddFormattedString('
8,8_ 0,0_ 4,4_ 1,1_ 4,4_
1,1_ 0,0_ 4,4_ 1,1_ 0,0_ 4,4_ 1,1_ 0,0_ 4,4_ 1,1_ 4,4_

```

```

1,1_ 0,0_ 8,8_ 0,0_ /msg нік_користувача текст - Відправити особисте
повідомлення');
Irc_Server_IRCMemo.AddFormattedString(' 8,8_ 0,0_ 4,4_ 1,1_ 0,0_
4,4_ 0,0_ 4,4_ 1,1_ 0,0_ 4,4_ 0,0_ 8,8_ 0,0_ 6/me
текст або наберіть текст і натисніть Ctrl+Enter -');
Irc_Server_IRCMemo.AddFormattedString(' 8,8_ 0,0_ 4,4_ 1,1_ 0,0_ 4,4_
1,1_ 0,0_ 4,4_ 1,1_ 0,0_ 4,4_ 1,1_ 0,0_ 4,4_ 1,1_
0,0_ 8,8_ 0,0_ 6Відправити Action на канал');
Irc_Server_IRCMemo.AddFormattedString(' 8,8_ 0,0_ 4,4_ 0,0_ 4,4_
0,0_ 4,4_ 0,0_ 4,4_ 0,0_ 4,4_ 0,0_ 8,8_ 0,0_
7/notice нік_користувач або #ім'я_канала и текст -');
Irc_Server_IRCMemo.AddFormattedString('
8,8_ 0,0_ 7Відправити Notice');

Irc_Server_IRCMemo.AddFormattedString('/away текст - Включити автовідповідь на
особисті повідомлення (/away - Виключити)');
Irc_Server_IRCMemo.AddFormattedString('/clear - Очистити вікно показу
повідомлень');
Irc_Server_IRCMemo.AddFormattedString('/help - Показати довідник по IRC
командах');
Irc_Server_IRCMemo.AddFormattedString('Ctrl+B - Жирний текст');
Irc_Server_IRCMemo.AddFormattedString('Ctrl+U - Підкреслений текст');
Irc_Server_IRCMemo.AddFormattedString('Ctrl+R - Інвертований текст');
Irc_Server_IRCMemo.AddFormattedString('Ctrl+K+номер кольору - 1ц 2в 3е 4т
5н 6про 7й текст');
Irc_Server_IRCMemo.AddFormattedString('Ctrl+K+номер кольору+,+номер кольору тла -
1,1ц 2в 3е 4т 5н 6про 7й текст на кольоровому тлі');

// Говорилка
try
    // Ініціалізація аудіопристрою
    CoCreateInstance(CLSID_MMAudioDest, Nil, CLSCTX_ALL,
IID_IAudioMultiMediaDevice, fIAMM);

    // Створення об'єкта, що перераховується, для перебору всіх движків у
системі
    // за допомогою інтерфейсу ITTSEnum
    CoCreateInstance(CLSID_TTSEnumerator, Nil, CLSCTX_ALL, IID_ITTSEnum,
aTTSEnum);
    // Скидаємо на перший
aTTSEnum.Reset;
    // Одержуємо перший движок
aTTSEnum.Next(1, ModeInfo, @NumFound);
    While NumFound > 0 do
        begin
            Speak_Combobox.Items.Add(String(ModeInfo.szModeName));
            // Одержуємо інші
aTTSEnum.Next(1, ModeInfo, @NumFound);
        end;
    if Speak_Combobox.Items.Count=-1 then
        begin
            Speak_Combobox.Enabled:=false;
        end;
except
    Speak_Combobox.Enabled:=false;
end;

// Читання настроювань із ini файлу й розшифровка пароля IRC
Ini:=TiniFile.Create(ExtractFilePath(ParamStr(0))+ 'Config.ini');
// Положення форми на робочому столі
Form1.Left:=Ini.ReadInteger('Position', 'Left', Form1.Left);
Form1.Top:=Ini.ReadInteger('Position', 'Top', Form1.Top);
// Speak
Speak_Combobox.ItemIndex:=Ini.ReadInteger('Speak', 'Speak_Combobox', Speak_Combobox.ItemIndex);
//if (Speak_Combobox.ItemIndex=-1) and (Speak_Combobox.Items.Count>0) then
Speak_Combobox.ItemIndex:=0;
//Speak_ComboboxChange(nil);

```

```

// IRC
Irc_Nik_Edit.Text:=Ini.ReadString('IRC','Nick',Irc_Nik_Edit.Text);
Irc_Server_Address_ComboBox.Text:=Ini.ReadString('IRC','Server',Irc_Server_Address_ComboBox.Text);
Irc_Password_Edit.Text:=Ini.ReadString('IRC','Password',Irc_Password_Edit.Text);
// Розшифровка пароля IRC
s:=Irc_Password_Edit.Text;
for i := 1 to Length(s) do s[i] := Char(23 xor Ord(c[i]));
Irc_Password_Edit.Text:=s;
Ini.Free;
end;

// Збереження налаштувань в ini файлі й шифрування пароля IRC
procedure TForm1.FormDestroy(Sender: TObject);
var
  Ini : TIniFile; //необхідно створити об'єкт, щоб потім з ним працювати
  // Шифрування пароля IRC
  s   : String[255];
  c   : array[0..255] of Byte absolute s;
  i   : Integer;
begin
  // Шифрування пароля IRC
  s:=Irc_Password_Edit.Text;
  for i := 1 to Ord(s[0]) do c[i] := 23 xor c[i];
  Irc_Password_Edit.Text:=s;

  // Запис налаштувань в ini файл і розшифровка пароля IRC
  Ini:=TIniFile.Create(extractfilepath(paramstr(0))+ 'Config.ini');
  // Положення форми на робочому столі
  Ini.WriteInteger('Position','Left',Form1.Left);
  Ini.WriteInteger('Position','Top',Form1.Top);
  //Speak
  Ini.WriteInteger('Speak','Speak_ComboBox',Speak_ComboBox.ItemIndex);
  // IRC
  Ini.WriteString('IRC','Nick',Irc_Nik_Edit.Text);
  Ini.WriteString('IRC','Server',Irc_Server_Address_ComboBox.Text);
  Ini.WriteString('IRC','Password',Irc_Password_Edit.Text);

  Ini.Free;

  // Обриваємо поточне з'єднання
  try
    if IdIRC.Connected then IdIRC.Disconnect;
  except
  end;

end;
////////////////////////////////////

// розтягуємо Edit
procedure TForm1.FormResize(Sender: TObject);
var
  i : Integer;
begin
  Irc_Server_Edit.Width:=Form1.Width;
  if Irc_PageControl.PageCount>1 then
    begin
      for i:=2 to Irc_PageControl.PageCount-1 do
        begin
          TEdit(FindComponent(Irc_PageControl.Pages[i].Name+'_Edit')).Width:=Form1.Width;
        end;
      Irc_Close_SpeedButton.Left:=Form1.Width-60;
      Irc_Faq_SpeedButton.Left:=Form1.Width-110;
      Irc_Log_SpeedButton.Left:=Form1.Width-160;
      Irc_Settings_SpeedButton.Left:=Form1.Width-210;
      Irc_Bot_SpeedButton.Left:=Form1.Width-260;
    end;
end;

```

```

Irc_Commands_SpeedButton.Left:=Form1.Width-310;
Irc_Channels_SpeedButton.Left:=Form1.Width-360;
Irc_ChannelSettings_SpeedButton.Left:=Form1.Width-410;
Irc_Favorite_SpeedButton.Left:=Form1.Width-460;
Irc_Clear_SpeedButton.Left:=Form1.Width-510;

Irc_Connected_Button.Left:=Form1.Width-110;
end;

procedure TForm1.Speak_ComboBoxChange(Sender: TObject);
var
  NumFound : DWord;
  // Для зберігання інформації про поточний движок
  ModeInfo : TTSMModeInfo;
begin
  try
    CoCreateInstance(CLSID_MMAudioDest, nil, CLSCTX_ALL,
IID_IAudioMultiMediaDevice, fIAMM);
    CoCreateInstance(CLSID_TTSEnumerator, nil, CLSCTX_ALL, IID_ITTSEnum,
aTTSEnum);
    aTTSEnum.Reset;
    // Перескакуємо на потрібний движок
    Form1.aTTSEnum.skip(Speak_ComboBox.ItemIndex);
    aTTSEnum.Next(1, ModeInfo, @NumFound);
    if assigned(fpModeInfo) then
      // якщо fpModeInfo не дорівнює nil
      dispose(fpModeInfo);
      new(fpModeInfo);
      fpModeInfo^:=ModeInfo;
      // завантажуюмо движок по його GUID
      aTTSEnum.Select(fpModeInfo^.gModeID, fITTSCentral, IUnknown(fIAMM));
    except
      end;
  end;
end.

```

```
// IrcCommandsUnit.pas - модуль обработки команд
```

```
unit IrcCommandsUnit;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, IRCMemo;
```

```
type
```

```
TIrcCommandsForm = class(TForm)  
  Irc_Commands_Label: TLabel;  
  Irc_Commands_OK_Button: TButton;  
  Irc_Commands_Close_Button: TButton;  
  Irc_Commands_GroupBox: TGroupBox;  
  Irc_Commands_Edit: TEdit;  
  procedure Irc_Commands_OK_ButtonClick(Sender: TObject);  
  procedure Irc_Commands_Close_ButtonClick(Sender: TObject);  
  procedure Irc_Commands_EditKeyDown(Sender: TObject; var Key: Word;  
    Shift: TShiftState);  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;
```

```
var
```

```
IrcCommandsForm: TIrcCommandsForm;
```

```
implementation
```

```
uses InstrumentsUnit;
```

```
{$R *.dfm}
```

```
procedure TIrcCommandsForm.Irc_Commands_OK_ButtonClick(  
  Sender: TObject);
```

```
var
```

```
i:integer;
```

```
begin
```

```
if (Irc_Commands_Edit.Text='') or (Irc_Commands_Edit.Text='#') then close  
  else
```

```
  begin
```

```
    i:=Pos(' ',Irc_Commands_Edit.Text);
```

```
    try
```

```
      // Включити режим AWAY
```

```
      if IrcCommandsForm.Caption='Включити режим AWAY' then
```

```
        begin
```

```
          Form1.IdIRC.WriteLine('AWAY :'+Irc_Commands_Edit.Text);
```

```
          away_message:=Irc_Commands_Edit.Text;
```

```
        end;
```

```
      // Зайти в канал
```

```
      if IrcCommandsForm.Caption='Зайти в канал' then
```

```
Form1.IdIRC.Join(Irc_Commands_Edit.Text);
```

```
      // Покинути канал
```

```
      if IrcCommandsForm.Caption='Покинути канал' then
```

```
Form1.IdIRC.Part(Irc_Commands_Edit.Text);
```

```
      // Послати приват
```

```
      if IrcCommandsForm.Caption='Послати приват' then
```

```
        begin
```

```
          Form1.IdIRC.WriteLine('PRIVMSG '+Copy(Irc_Commands_Edit.Text,1, i-1)+'  
:'+Copy(Irc_Commands_Edit.Text,i+1,Length(Irc_Commands_Edit.Text)-i));
```

```
          With Form1 Do
```

```
TIRCMemo (FindComponent (Form1.Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFor
```

```

matedString(TAG_COLOR+'1<'+Copy(Irc_Commands_Edit.Text,1, i-1)+'>
'+Copy(Irc_Commands_Edit.Text,i+1,Length(Irc_Commands_Edit.Text)-i));
    // Робота з файлами лога
    Form1.IRCLogs(Copy(Irc_Commands_Edit.Text,1, i-
1),Copy(Irc_Commands_Edit.Text,1, i-
1),Copy(Irc_Commands_Edit.Text,i+1,Length(Irc_Commands_Edit.Text)-i));
    end;

    // Послати Notice
    if IrcCommandsForm.Caption='Послати Notice' then
    begin
        Form1.IdIRC.WriteLine('NOTICE '+Copy(Irc_Commands_Edit.Text,1, i-1)+'
:'+Copy(Irc_Commands_Edit.Text,i+1,Length(Irc_Commands_Edit.Text)-i));
        With Form1 Do
TIRCMemo(FindComponent(Form1.Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFor
matedString(TAG_COLOR+'7* - '+Copy(Irc_Commands_Edit.Text,1, i-1)+' -
'+Copy(Irc_Commands_Edit.Text,i+1,Length(Irc_Commands_Edit.Text)-i));
        end;

    // Послати Action
    if IrcCommandsForm.Caption='Послати Action' then
    begin
        Form1.IdIRC.WriteLine('PRIVMSG '+Copy(Irc_Commands_Edit.Text,1, i-1)+'
:[] ACTION '+Copy(Irc_Commands_Edit.Text,i+1,Length(Irc_Commands_Edit.Text)-i)+'[]
');
        With Form1 Do
TIRCMemo(FindComponent(Form1.Irc_PageControl.ActivePage.Name+'_IRCMemo')).AddFor
matedString(TAG_COLOR+'6* '+Form1.IdIRC.Nick+'
'+Copy(Irc_Commands_Edit.Text,i+1,Length(Irc_Commands_Edit.Text)-i));
        // Робота з файлами лога
        Form1.IRCLogs(Copy(Irc_Commands_Edit.Text,1, i-
1),Form1.IdIRC.Nick,Copy(Irc_Commands_Edit.Text,i+1,Length(Irc_Commands_Edit.Tex
t)-i));
        end;

    // Whois користувача (online)
    if IrcCommandsForm.Caption='Whois користувача (online)' then
Form1.IdIRC.WriteLine('WHOIS '+Irc_Commands_Edit.Text);

    // Whowas користувача (offline)
    if IrcCommandsForm.Caption='Whowas користувача (offline)' then
Form1.IdIRC.WriteLine('HOWAS '+Irc_Commands_Edit.Text);

    // Перемінити нік
    if IrcCommandsForm.Caption='Перемінити нік' then Form1.IdIRC.WriteLine('NICK
'+Irc_Commands_Edit.Text);

    // Послати STCP запит Ping
    if IrcCommandsForm.Caption='Послати STCP запит Ping' then
Form1.IdIRC.WriteLine('PRIVMSG '+Irc_Commands_Edit.Text+' :[] PING[] ');

    // Послати STCP запит Time
    if IrcCommandsForm.Caption='Послати STCP запит Time' then
Form1.IdIRC.WriteLine('PRIVMSG '+Irc_Commands_Edit.Text+' :[] TIME[] ');

    // Послати STCP запит Version
    if IrcCommandsForm.Caption='Послати STCP запит Version' then
Form1.IdIRC.WriteLine('PRIVMSG '+Irc_Commands_Edit.Text+' :[] VERSION[] ');

    // Послати STCP запит Finfer
    if IrcCommandsForm.Caption='Послати STCP запит Finger' then
Form1.IdIRC.WriteLine('PRIVMSG '+Irc_Commands_Edit.Text+' :[] FINGER[] ');

    // Послати STCP запит UserInfo
    if IrcCommandsForm.Caption='Послати STCP запит UserInfo' then
Form1.IdIRC.WriteLine('PRIVMSG '+Irc_Commands_Edit.Text+' :[] USERINFO[] ');

    // Послати STCP запит ClientInfo

```

```

    if IrcCommandsForm.Caption='Послати CTCP запит ClientInfo' then
Form1.IdIRC.WriteLine('PRIVMSG '+Irc_Commands_Edit.Text+' :[] CLIENTINFO ');

    // Послати CTCP запит Source
    if IrcCommandsForm.Caption='Послати CTCP запит Source' then
Form1.IdIRC.WriteLine('PRIVMSG '+Irc_Commands_Edit.Text+' :[] SOURCE ');

    // Послати CTCP запит Sound
    if IrcCommandsForm.Caption='Послати CTCP запит Sound' then
Form1.IdIRC.WriteLine('PRIVMSG '+Irc_Commands_Edit.Text+' :[] SOUND ');

    // Одержати інформацію про канал
    if IrcCommandsForm.Caption='Одержати інформацію про канал' then
Form1.IdIRC.WriteLine('CS INFO '+Irc_Commands_Edit.Text+' ALL');

    // Зареєструвати нік
    if IrcCommandsForm.Caption='Зареєструвати нік' then
Form1.IdIRC.WriteLine('NS REGISTER '+Irc_Commands_Edit.Text);

    // Зареєструвати канал
    if IrcCommandsForm.Caption='Зареєструвати канал' then
Form1.IdIRC.WriteLine('CS REGISTER '+Irc_Commands_Edit.Text);

    Close;
except
    // Помилка
    on e:Exception do
        begin
            Form1.Reconnect(E.Message);
            Close;
        end;
end;
end;
end;

procedure TIrcCommandsForm.Irc_Commands_Close_ButtonClick(Sender: TObject);
begin
    Close;
end;

// Вставка спецтегів
procedure TIrcCommandsForm.Irc_Commands_EditKeyDown(Sender: TObject;
    var Key: Word; Shift: TShiftState);
begin
    Form1.IrcColor(nil, IrcCommandsForm.Irc_Commands_Edit, Key, Shift);
end;

end.
```