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 клієнтського модуля Інтернет-радіо”

КБПЗ – 2023

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-22М-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Білоконенко В.Ю.
«____» _____ 2023 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Кислун О.А.
«____» _____ 2023 р.
Рецензент _____
_____

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань _12 _____*"Інформаційні технології"*_____

Спеціальність 123 *"Комп'ютерна інженерія"*

Освітньо-професійна (освітньо-наукова) програма _*"Комп'ютерна інженерія"*_

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

Білоконенку Валентину Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи клієнтського модуля Інтернет-радіо*

2. Керівник роботи *Кислун Олег Андрійович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 35-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту *10.12.2023 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою розробки є дослідження та програмна реалізація системи клієнтського модуля Інтернет-радіо*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.	*6. Наукова новизна.*
2.Перегляд аналогічних існуючих систем.	*7. Економічна ефективність розробленої програми.*
3. Опис і обґрунтування проектних рішень.	*8. Заходи з охорони праці та техніки безпеки.*
4. Етапи програмування системи.	*9. Висновки.*
5. Впровадження системи в промислову експлуатацію	

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна	*1 аркуш*
Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*
Показники економічної ефективності	*1 аркуш*

КОПІЯ 2023

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)

КБТЗ - 2023

АНОТАЦІЯ

Білоконенко В.Ю. Дослідження та програмна реалізація системи клієнтського модуля Інтернет-радіо. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи клієнтського модуля Інтернет-радіо.

Метою розробки є дослідження та програмна реалізація системи клієнтського модуля Інтернет-радіо.

Об'єктом дослідження є процес клієнтського модуля Інтернет-радіо.

Предметом дослідження є методи клієнтського модуля Інтернет-радіо.

Методи дослідження базуються на методах теорії телетрафіку, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи клієнтського модуля Інтернет-радіо.

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 RAD Studio Delphi 10.

Ключові слова: комп'ютерна інженерія, Інтернет-радіо

КПЗ – 2023

ABSTRACT

Bilokonenko V.Yu. Research and software implementation of the Internet radio client module system. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this final qualification work for the second (master's) level of higher education, software is developed, which is intended for the Internet radio client module system.

The goal of the development is research and software implementation of the Internet radio client module system.

The object of the study is the process of the Internet radio client module.

The subject of the study is the methods of the Internet radio client module.

Research methods are based on teletraffic theory methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the Internet radio client module 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 RAD Studio Delphi 10 environment.

Keywords: computer engineering, Internet radio

КОПЗ – 2023

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 3
ВСТУП..... 5
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 7
1.1 Призначення системи..... 7
1.2 Область застосування..... 8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування..... 25
2.3 Розгорнута постановка завдання 31
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 32
3.1 Опис функціонування системи 32
3.2 Розробка структурної схеми..... 39
3.3 Розробка функціональної схеми 43
3.4 Розробка діаграми процесів..... 46
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ..... 48
4.1 Розробка блок-схем та опис алгоритмів функціонування системи..... 48
4.2 Захист розробленого програмного забезпечення..... 58
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 63
6 НАУКОВА НОВИЗНА 65

КОПІЯ – 2023

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ 66
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти 66
7.2 Розрахунок трудомісткості розробки програмної продукції 68
7.3 Визначення чисельності виконавців і планового фонду зарплати 70
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника 74
7.5 Визначення собівартості розробки та ціни програмної продукції. 79
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції. 82
7.7 Визначення експлуатаційних витрат 82
7.8 Визначення економічної ефективності програмної продукції. 84
7.9 Висновок 86
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 87
8.1 Вступ 87
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером 88
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ... 89
8.4 Розробка заходів з умов поліпшення охорони праці 92
8.5 Розрахункова частина 93
9 ОСНОВНІ ВИСНОВКИ 97
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 99

КОПІЯ 2023

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БД – база даних

КМЗ – корпоративна мережа зв’язку

ЛОМ – локальна обчислювальна мережа

ОЗП – оперативно-запам’ятовувальний пристрій

ПАТМ – пристрій автоматизованої телефонної мережі

ПЗ – програмне забезпечення

ПП – програмний продукт

СПД – система передачі даних

СУБД – система управління БД

AVVID – архітектурна модель фирмы Cisco Systems

CNG – Comfort Noise Generator. Генератор комфортного шуму

DSP – Digital Signal Processor. Процесор цифрової обробки сигналів

DTMF – Dual Tone Multi-Freqency. Багаточастотна система кодування цифр номера

GK – Gatekeeper. Гейткіпер . Виконує функції керування зоною мережі H.323

GW – Gateway. Шлюз. Апаратно-програмний комплекс, що забезпечує обмін даними між мережами різних типів

H.323 – Рекомендація ITU-T, що визначає системи мультимедійного зв'язку в мережах з пакетною комутацією

H.248 – Протокол керування транспортним шлюзом

IP – Internet Protocol. Протокол міжмережної взаємодії

LPC – Linear Prediction Coding. Кодування з лінійним пророкуванням

MCU – Multipoint Control Unit. Пристрій керування конференцією

MG – Media Gateway. Транспортний шлюз

MGCP – Media Gateway Control Protocol. Протокол керування шлюзами

КПЗ – 2023

MP – Multipoint processor. Процесор для обробки інформації користувачів при централізованих конференціях

OSI – Open System Interconnection. Взаємодія відкритих систем

PPP – Point-to-Point Protocol. Протокол двостороннього зв'язку

RADIUS – Remote Authentication Dial-In User Service. Протокол автентифікації й авторизації абонентів, а також обліку обсягу наданних їм послуг

RAS – Registration Admission and Status. Протокол взаємодії термінального встаткування з gatekeeper. Входить у сімейство протоколів H.323

RSVP – Resource Reservation Protocol. Протокол резервування ресурсів

RTCP – Real-time Transport Control Protocol. Протокол контролю транспортування інформації в реальному часі

SIP – Session Initiation Protocol. Протокол ініціювання сеансів зв'язку

TCP – Transmission Control Protocol. Протокол керування передачею (даних) Основний транспортний протокол у стеці протоколів TCP/IP.

TCP/IP – Transmission Control Protocol/Internet Protocol. Стек протоколів, що забезпечують організацію зв'язку між комп'ютерами в мережі Інтернет

UDP – User Datagram Protocol. Протокол передачі дейтаграмм користувача. Подібно TCP, використовує для доставки даних протокол IP. На відміну від TCP/IP, передбачає обмін дейтаграммами без підтвердження

VoIP – Voice over Internet Protocol. Технологія, що дозволяє використовувати IP-мережу для передачі мовної інформації

КПІ – 2023

ВСТУП

Актуальність теми. Бурхливий розвиток інтернет-радіо обумовлено широкими можливостями технології, що дозволяє широкомовним радіостанціям не тільки істотно розширити свою аудиторію, але і якісно поліпшити надавані слухачам послуги. Інтернет-радіо має наступні основні переваги:

– Глобальний охват. Зона віщання не обмежується, як у випадку з передачею по ефірі, радіусом дії ретранслятора. Доступ до трансльованої передачі може одержати користувач, підключений до мережі Інтернет у будь-якій крапці земної кулі.

– Висока якість звукового сигналу. Застосовувані програмно-технічні засоби дозволяють надійно доставляти сигнал без втрат у якості й завмирань (завмирання, фединг – зміна амплітуди й фази сигналу через переміщення передавача або приймача в системі радіозв'язку й/або поширення сигналу через неоднорідне середовище).

– Мультимедійний супровід. Трансляція звукового сигналу (наприклад, музичної композиції) може супроводжуватися передачею алфавітно-цифрової або графічної інформації.

– Зворотний зв'язок зі слухачем. Простий і зручний зв'язок може бути організована по Інтернету у вигляді голосового спілкування або обміну повідомленнями в процесі віщання.

Перераховані можливості дозволяють організувати віщання з урахуванням переваг слухачів (сформувати постійну аудиторію по інтересах), підсилити сприйняття трансльованих звукових програм за рахунок мультимедійного супроводу, забезпечити необхідна якість трансляції незалежно від місця прийому сигналу.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи клієнтського модуля Інтернет-радіо.

КПЗ – 2023

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

– Огляд існуючих систем клієнтського модуля Інтернет-радіо.

– Дослідження системи клієнтського модуля Інтернет-радіо.

– Програмна реалізація системи клієнтського модуля Інтернет-радіо.

Об'єктом дослідження є процес клієнтського модуля Інтернет-радіо.

Предметом дослідження є методи клієнтського модуля Інтернет-радіо.

Методи дослідження базуються на методах теорії телетрафіку, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод клієнтського модуля Інтернет-радіо.

– Розроблено вітчизняний продукт клієнтського модуля Інтернет-радіо, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі клієнтського модуля Інтернет-радіо.

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 клієнтського модуля Інтернет-радіо, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБГВ–2023

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Інтернет-радіо полегшує надання всілякої інформації – від детальних відомостей про трансльовану музичну композицію до досить різноманітних знань у всіляких областях. З огляду на, що значне число слухачів є постійними користувачами Інтернету, доступ до звукової інформації, включаючи музику (до речі, без порушення прав інтелектуальної власності), за допомогою інтернет-радіо виявляється для них найбільш зручним і природним.

Інтернет-радіо дозволяє не тільки заздалегідь познайомити слухачів із програмою трансляції, але й гнучко міняти її з урахуванням переваг аудиторії. Для цього є можливості інтерактивної взаємодії в реальному масштабі часу: за допомогою голосових повідомлень, живого спілкування в ефірі й обміну цифровою інформацією у вигляді електронної пошти або коротких повідомлень.

Обмін цифровими повідомленнями може вестися в рамках окремих цільових груп і бути пов'язаним з певною темою, обговорюваної в трансльованій програмі. Можливість у реальному часі організувати голосування по заданій темі й об'єктивно оцінити його результати дозволяє гнучко формувати зміст музичних програм і актуальних рубрик.

Завдяки інтерактивній взаємодії з аудиторією, а також наявності засобів контролю за кількістю підключень, виконаних для прослуховування інтернет-радіо й участі в обговоренні трансльованих програм, можна об'єктивно визначати рейтинг радіостанції й рівень інтересу до її окремих трансляцій. Аналіз цих даних допоможе скоординувати зусилля для подальшого розширення аудиторії й підвищення комерційної привабливості радіостанції для рекламодавців і різних організацій, зацікавлених у поширенні тої або іншої інформації.

КПЗ – 2023

1.2 Область застосування

Якщо ви діджей на радіо або відкриваєте власну онлайн-радіостанцію, дуже важливо вибрати правильне програмне забезпечення для радіо, щоб втілити своє бачення в життя. Зі свого минулого досвіду роботи з різними програмними платформами онлайн-радіо я зрозумів, що для успішної онлайн-станції є кілька ключових функцій.

Надійність: Перш за все, важлива надійність програмного забезпечення. Без надійної системи ніщо інше не має значення. Ваша онлайн-радіостанція має працювати цілодобово, без вихідних, тому вкрай важливо вибрати програмне забезпечення, яке є надійним і має солідну репутацію. Немає нічого гіршого, ніж ваша станція вимикається посеред шоу або втрачає слухачів через буферизацію чи проблеми з підключенням. Знайдіть час, щоб дослідити та прочитати відгуки користувачів на таких платформах, як Capterra, Trustpilot і G2, перш ніж перейти на платформу.

Автоматизація: Ще однією важливою особливістю, яку слід враховувати, є автоматизація. До недавньої появи ефективних опцій я годинами налаштовував свої шоу вручну. Наявність можливості автоматизувати певні аспекти вашої онлайн-радіостанції може стати порятунком. Шукайте програмне забезпечення, яке має надійні функції автоматизації, такі як планування списків відтворення, повторювані шоу календаря, ротації, автоматичний діджей і автоматичне перемикання після збою у разі проблем із з'єднанням.

Налаштування: Налаштування сторінки станції – ще одна важлива функція, яку слід враховувати. Зробивши свою онлайн-радіостанцію унікальною, ви виділите її серед конкурентів. Шукайте програмне забезпечення, яке дає вам можливість налаштувати тему та інтерфейс вашої сторінки, а також додати логотип і брендинг. Ваші слухачі оцінять особистий підхід. Не забудьте також налаштувати свої канали соціальних мереж, щоб забезпечити послідовність бренду.

КОПЗ – 2023

Інтерфейс: Також важливо мати зручний інтерфейс. Керування онлайн-радіостанцією не повинно завдавати вам головного болю. Тому вибирайте таке програмне забезпечення, як Airtime Pro, яке має інтуїтивно зрозумілий і зручний інтерфейс, що дозволяє легко завантажувати музику, планувати списки відтворення та керувати вашою станцією з одного централізованого місця.

Інтеграція: Нарешті, інтеграція з іншими платформами допоможе охопити якомога більше слухачів. Шукайте програмне забезпечення, яке інтегрується з такими популярними платформами, як iTunes, TuneIn і Spotify, що дозволяє вашим слухачам легко знаходити та налаштовуватися на вашу станцію. Крім того, якщо ви збираєтеся вести пряму трансляцію, переконайтеся, що ваше програмне забезпечення можна легко інтегрувати зі сторонніми програмами для трансляції, такими як MIXX, BUTT, SAM Broadcaster і Nicecast.

Інші речі, які слід враховувати: Окрім цих ключових функцій, є ще кілька факторів, про які слід пам’ятати, вибираючи програмне забезпечення для онлайн-радіо. Наприклад, важливо враховувати тип підтримки та ресурси, які пропонує постачальник програмного забезпечення. Шукайте платформу, яка надає корисну документацію, навчальні посібники та варіанти підтримки, які допоможуть вам отримати максимальну віддачу від програмного забезпечення. Нарешті (і це, мабуть, само собою зрозуміло), подумайте про те, що ви отримуєте за свої гроші. Хоча існує багато дешевих варіантів, часто варто інвестувати в програмну платформу, яка надає більш надійні функції та підтримку. Тільки не забудьте врахувати вартість програмного забезпечення під час планування свого бюджету.

Зосередившись на цих функціях і факторах, ви будете на шляху до створення успішної та унікальної онлайн-радіостанції, яка сподобається вашим слухачам.

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

КПІ – 2023

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

AudioJack

Головна панель програми представлена у вигляді вікна, у якому є перелік доступних станцій. Ледве вище цього списку ліворуч праворуч перебувають кнопки: play/stop/start recording, кнопка додавання радіо через URL, видалення, налаштування станцій, пошук серед наявних, пошук у Мережі (дозволяє відшукати потрібну композицію, і якщо така є у відтворенні, то скачати її не важко буде). Крім того, можна записати дані про пісню, поки її ні, і якщо в майбутньому її будуть програвати, програма автоматично запише композицію на комп'ютер.

Програма може записувати радіо у форматі MP3, WAV, OGG, може обновляти список радіостанцій, становити списки вибраного. AudioJack надзвичайно функціональний, до всього іншого вміщає в себе конвертор аудіо (WAV/WMA/MP3/MP2/OGG/), редактор медіафайлів (з додатковими налаштуваннями (посилення, тиша, інвертування, відтворення навпаки й т.д.), може створювати CD, створювати мультимедіа CD/DVD, копіювати CD і створювати для них обкладинки.

КПЗ – 2023

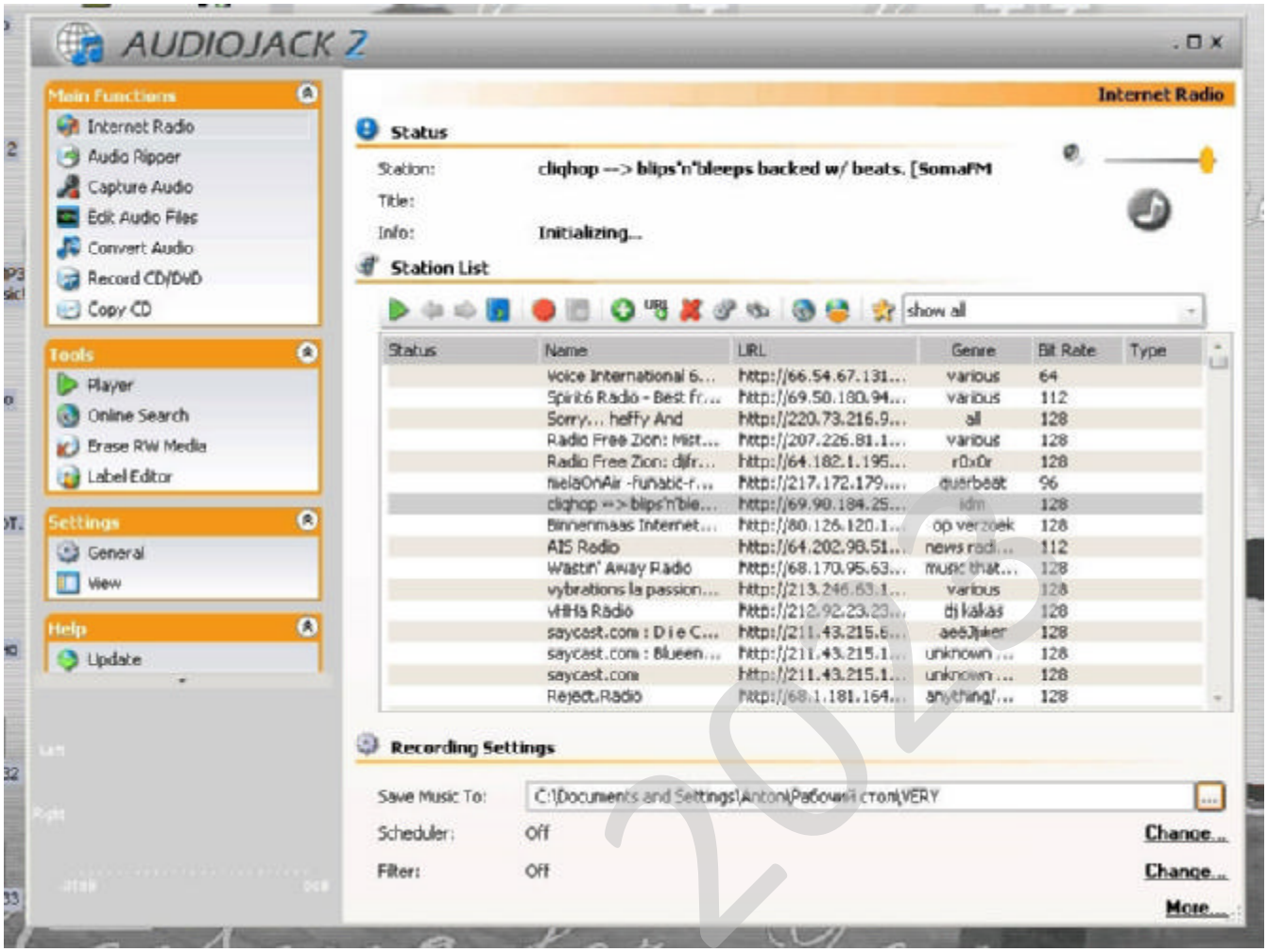

Рисунок 2.1 – Інтерфейс користувача AudioJack

SopCast

SopCast – це однорангова технологія, що забезпечує можливість перегляду відео через Інтернет.

Весь принцип роботи системи дуже схожий на роботу технології bit-torrent, коли передача інформації децентралізована.

Під час перегляду ТБ або прослуховування Інтернет-радіо ви також можете здійснювати одночасну трансляцію цього матеріалу в Інтернеті.

Окрім прослуховування інтернет-радіо та перегляду ТБ, SopCast дозволяє створити власний канал мовлення з необмеженою кількістю каналів. Щоб організувати свій канал мовлення, не потрібно купувати дороге і потужне

КБПЗ – 2025

обладнання, оскільки мовлення буде здійснюватися за тією ж системою, що і прослуховування інтернет-радіо або перегляд телепередач.

Після запуску програми в певній вкладці ви можете побачити весь список доступних для відтворення каналів.

Також у цій програмі є додаткова функція для перегляду відео на зручному для вас відеоплеєрі, наприклад KMplayer, VLC або будь-якому іншому, який підтримує цей вид відтворення.

Якщо доступних для перегляду програм вам не вистачає, ви можете без проблем знайти велику кількість трансляцій у форматі додатка і забути про зависання, які трапляються при перегляді трансляцій онлайн.

Серед основних функцій програми можна виділити наступні:

- стабільність і доступність мовлення, завдяки використанню технології P2P;
- висока швидкість кешування;
- є можливість створити власний канал мовлення без використання потужного обладнання;
- програма використовує невелику кількість програмних ресурсів;
- можливість запису онлайн-трансляції на жорсткий диск ПК;
- є можливість транслювати трансляційний канал на будь-якому плеєрі в системі;
- додаток безкоштовний і в ньому відсутня реклама;
- існує підтримка великої різноманітності платформ і операційних систем.

YourFlavor Radio

Як би вам не подобалася улюблена мелодія, рано чи пізно вона почне вас дратувати. А щоб прослухати всі мелодії улюбленого музичного жанру, знадобляться роки, а то й десятиліття, враховуючи те, що репертуар постійно поповнюється новими піснями.

YourFlavor Radio – це унікальне потокове радіо, яке дозволяє слухати не просто одного співака, а схожі композиції з музичного жанру. Вам потрібно лише

КПІ – 2023

вказати, яка музика вам подобається, а програма зробить все інше. Ознайомитися з новими і найпопулярнішими піснями популярних виконавців улюбленого жанру ви можете прямо зараз, для цього достатньо просто завантажити безкоштовно YourFlavor Radio.

Основні можливості програми:

– Створення списку відтворення на основі певної пісні, виконавця або жанру музики.

– Унікальний плейлист, який включає як сучасні, так і найпопулярніші пісні минулих років.

– Інноваційний підхід до створення персонального плейлиста на основі 450 параметрів.

– Музика відтворюється відповідно до ваших уподобань.

– Можливість збільшення різноманітності шляхом додавання кількох виконавців.

– При налаштуванні надаються детальні підказки, іноді досить їх просто дотримуватися.

– Можливість пропускати треки, які вам не подобаються.

– Вся музика, яку ви слухали, зберігається в історії, що дозволяє повторювати відтворення найулюбленішої.

– При відтворенні доступна коротка інформація про виконавця і список схожих композицій, яких немає в списку відтворення. При бажанні ви можете переключитися на цей список.

– Стабільне штатне мовлення.

V-Radio

V-Radio – це програма для прослуховування онлайн радіостанцій по всьому світу, яка відрізняється стабільною роботою, мінімальними вимогами і простим інтерфейсом.

Щоб знайти та додати нові радіостанції, вам потрібно лише вказати координати станції та її назву.

КОПІЯ – 2023

Переваги програми V-Radio

- Величезний вибір радіостанцій з будь-якої точки світу.
- Висока якість відтворення.
- Без реклами і оголошень.
- Простий і інтуїтивно зрозумілий інтерфейс.
- Список відтворення та списки улюблених каналів, якими можна поділитися з іншими користувачами.
- Під час відтворення пісні доступна така інформація: ім'я виконавця, його назва, частота і радіостанція.
- Зручна навігація між станціями зі збереженням історії подорожей.
- Пошук пісень за обраними параметрами.
- Автоматичне оновлення.
- Мінімальне використання системних ресурсів.

Maxuden Radio

В Інтернеті можна знайти велику різноманітність програмного забезпечення, призначеного для підключення та прослуховування онлайн-радіостанцій. Однак програма не часто позбавлена одного неприємного недоліку – вбудованої реклами, причому досить агресивної. Maxuden Radio – приклад приємного видатного програмного забезпечення з базовим набором станцій, доступних для прослуховування відразу після установки. Завантажити Maxuden Radio безкоштовно з нашого сайту не проблема, а якщо ви не знайшли нічого для себе в базовому наборі, то можете додати нові станції у вигляді URL в програмі.

Maxuden Radio характеризується простотою та легкістю використання для прослуховування потокового аудіо на вашому домашньому ПК. Програма відображає інформацію про виконавця або автора пісні електронного треку, є функціонал для веб-серфінгу, функція запису радіо та багато іншого.. Крім усього іншого, Maxuden Radio може складати списки музичних станцій за музичною формою, легко додавати нові посилання на аудіопотоки в програмі, створювати

КОПІЯ 2023

списки тих, які вам найбільше сподобалися. Однією з переваг програми є запис трансляції в популярному форматі mp3, який відкривається. Запис можна розбити на окремі файли. Maxuden Radio може працювати у фоновому режимі, що робить його майже непомітним для користувача, споживаючи при цьому мінімум апаратних ресурсів. Ви можете легко завантажити Maxuden Radio безкоштовно з нашого веб-сайту.

Коротко про основні переваги:

- Швидкий пошук за списком відтворення радіостанцій;
- Розробники мають власні списки відтворення та підтримуються популярні;
- Ви можете додавати та редагувати радіостанції;
- Пошук в Інтернеті за інформацією про автора треку;
- Низькі вимоги до обладнання;
- Повна відсутність реклами.

Тепер ми перерахуємо основні нововведення, представлені у версії 4.0:

- Новий код, написаний з нуля;
- Відсутність підводних каменів попередніх версій;
- Новий движок для скінів;
- Вікно може бути компактно представлено в щільному вигляді або бібліотечному стилі;
- Виправлена помилка, яка виникає під час перемикання станцій;
- Ви можете редагувати список пошукових систем;
- Додано більше налаштувань підключення;
- Трансляцію тепер можна записувати у форматі OGG Vorbis;
- Програма інтегрована з панеллю завдань Windows 7, 8.

KLBPleer

KLBPleer – плеєр для прослуховування аудіофайлів, а також радіостанцій. Ця програма працює за допомогою безкоштовної бібліотеки bass.dll, завдяки якій

КПЗ – 2023

програму можна підключати до аудіопристроїв, а також відтворювати аудіофайли та записувати потокове передавання з Інтернету.

Цей програвач також містить арсенал додаткових цікавих функцій, а саме:

– перемикання станції, якщо вона в несправному стані, плеєр дозволяє шукати робочу станцію;

– є можливість переглянути назви раніше відтворених треків, але тільки якщо станція підтримує цю функцію;

– є можливість одночасного відтворення окремих потоків, як для радіо, так і для плеєра;

– є можливість окремо регулювати гучність плеєра і радіо;

– плеєр дозволяє швидко змінювати рівень гучності, наприклад, натиснувши «Пробіл» один раз, гучність збільшиться на 50%, при другому натисканні – на 100%, а натиснувши «Esc», рівень звуку буде 0%.

– Є можливість підрахувати загальний рівень всього трафіку, а також загальний час прослуховування всіх станцій.

– Плеєр дозволяє відхиляти рекламні фрази, які вже внесені в текстовий файл garbage.txt, які з'являються в рядку «Назва треку» і передаються станцією.

– Корисною функцією також є те, що плеєр автоматично вимикає звук на кількох станціях під час реклами, що робиться через теги мовника. Щоб включити цю функцію в плеєрі, потрібно поставити галочку в AdsCut.

– KLBPleer також дозволяє регулювати розмір мережевого буфера, який не дозволяє перезавантажуватися.

– Керувати плеєром можна за допомогою медіа-клавіш на клавіатурі.

– Ви можете вимкнути лише плеєр за допомогою кнопки на медіа-клавіатурі.

Ви можете завантажити KLBPleer безкоштовно з нашого веб-сайту.

Плеєр KLBPleer має наступні функції:

– відтворення у випадковому порядку можливо двома способами;

КПЗ-2023

– плеєр дозволяє повторювати поточний трек, зчитуючи його з оперативної пам'яті ПК;

– можливий повтор декількох треків як з диска, так і з оперативної пам'яті ПК;

– плеєр дозволяє повторювати частини треку між точками AB;

– можна повторити весь список.

Radiocent

Безкоштовна програма Radiocent доступна в трьох версіях для різних операційних систем: Windows, Android та iOS. Це не що інше, як одна з найбільших колекцій радіостанцій з усього світу. Список налічує кілька десятків тисяч станцій, і доступ до кожної з них можна отримати миттєво і безкоштовно.

Ключовою особливістю цієї програми є не тільки наявність безлічі станцій, розділених за жанрами, але і можливість записувати улюблені треки для подальшого прослуховування їх на комп'ютері. І навіть якщо запис було розпочато не на початку треку, він буде записаний повністю.

Основні характеристики Radiocent

– простий програвач онлайн-радіостанцій і MP3-файлів;

– сортування станцій за жанром, країною, якістю та назвою;

– Розділ «Вибране» для збереження улюблених станцій;

– функція онлайн ТБ;

– зручний список із сортуванням за жанрами;

– Запис музики в MP3 в режимі реального часу;

– регулярне оновлення каталогу радіостанцій (оновлення відбувається автоматично при запуску програми);

– управління програмою з системного трея і можливість відтворення музики при згорнутому вікні програми.

Перевагами програми перед іншими аналогічними базами радіостанцій можна назвати зручний і мінімалістичний інтерфейс без зайвих кнопок і

КОПІЯ 2023

елементів, наявність української мови, збереження історії прослуховування і відсутність системних ресурсів.

Screamer Radio

Screamer Radio – безкоштовне програмне забезпечення для прослуховування інтернет-радіостанцій, що працює під управлінням сімейства операційних систем Windows. Потокове радіо можна відтворювати у форматах MP3, AAC, WMA або Ogg Vorbis.

Також можна записати потік у файл, а потім прослухати, отримати ID3-теги з сервера трансляції. За бажанням користувача інформація, яку несуть теги, може бути відображена під час поточного відтворення та записана у структурі файлу під час запису в медіа-потік.

Описаний додаток є безкоштовним пропрієтарним програмним забезпеченням, тому ви можете завантажити Screamer Radio безкоштовно, почати слухати радіо та багато іншого.

Screamer Radio простий у використанні. Основні переваги наступні:

- програма повністю безкоштовна;
- інтерфейс не набридає рекламою, наявність якої характерна для більшості подібних програм;
- достатня кількість інтернет-радіостанцій на вибір;
- можна додавати радіостанції в закладки, зберігається історія відтворення;
- гнучкість налаштувань;
- потоки легко записуються і можуть бути легко розбиті на окремі доріжки, а потім записані ID3-теги;
- Screamer Radio майже не споживає системні ресурси.

Вищезазначені переваги будуть у вашому розпорядженні, вам просто потрібно безкоштовно завантажити Screamer Radio і почати ним користуватися, якщо ви хочете слухати улюблену музику.

Недоліки програми в тому, що інтерфейс не продуманий, і навігація по радіостанціях дещо незручна.

КБПЗ – 2023

RadioClicker

RadioClicker – це менеджер якості радіостанцій і телеканалів. Він має низькі вимоги і працює практично на будь-якому комп'ютері. З його допомогою ви можете переглянути потрібну програму онлайн.

Список доступних ресурсів вражає – це понад сотня найменувань, які транслюються різними мовами. Легко знайти все: спорт, розваги, нон-фікшн матеріали.

RadioClicker може записати всю прослухану музику в звуковий (wav, mp3) файл. Вона може автоматично поставити назву і виконавця композиції.

Переваги RadioClicker:

- RadioClicker містить тільки якісні станції;
- безкоштовно завантажити RadioClicker – значить отримати можливість записувати улюблені твори на льоту;
- відсутність реклами, непотрібних панелей інструментів;
- ергономічний інтерфейс;
- Утиліта не збирає інформацію про користувачів.

Відомі проблеми RadioClicker:

- Програма відмовляється запускатися, виводячи в інформаційному вікні повідомлення «доступ заборонено». Це проблема з вашим антивірусом. Додайте RadioClicker до довіреної зони.
- Ви стали свідком повідомлення «станція не відповідає». Варто подивитися на ваш брандмауер. Переконайтеся, що він не заважає програмі.
- «Заморозити» аудіо або відео. Найпоширенішою причиною є недостатня швидкість інтернет-з'єднання. Можна спробувати змінити «буфер відтворення». Або перезапустіть запущені станції.

Xstar Radio

Оновлена версія Xstar Radio – це програма в portable-версії, яка призначена для прослуховування онлайн-радіостанцій.

КПЗ – 2023

Сьогодні постійно оновлюваний і зростаючий список радіостанцій налічує 2000 каналів, що транслюють потокове аудіо зі швидкістю 96-128 Кб/с.

Xstar Radio містить меншу кількість станцій, ніж деякі конкуренти, але програма не має мертвих лінків радіоканалів, тому нова версія Xstar Radio є сучасним якісним продуктом.

Ви можете слухати радіо за допомогою програми, не заплативши ні копійки. На нашому сайті є посилання для безкоштовного скачування Xstar Radio.

Вам доведеться заплатити тільки за можливість запису і потокової передачі в mp3 і за зміну візуального оформлення. Щоб зробити всі функції програми доступними, необхідно завантажити реєстраційний додаток і скористатися одним із двох способів оплати: відправити смс або оплатити кредитною карткою.

Розблокування платних можливостей програми прив'язане до комп'ютера, на якому воно було здійснено, тобто «розблокована» програма не працюватиме на іншому комп'ютері (нагадаємо, що завантаження Xstar Radio на нашому сайті є безкоштовним). Ціна раніше заблокованих можливостей програми становить близько 10 доларів.

Запис музики, що транслюється інтернет-радіо, здійснюється за допомогою спеціального плагіна. Плагін має можливість зберігати файли в різні файли окремих композицій, які розпізнаються в аудіопотоці. Шлях до папки завантажених треків – «C: Record». Після натискання клавіші зі зірочкою відкривається меню «Вибране», де ви можете зберегти улюблені радіостанції.

BooRadio

BooRadio – це програма, яка дозволяє слухати радіо і дивитися телеканали через Інтернет. Ви можете слухати музику і дивитися шоу онлайн абсолютно безкоштовно. Додаток має платну та безкоштовну версії.

За допомогою цього безкоштовного програмного забезпечення ви можете дивитися улюблені телеканали. Список каналів доступний на сервері. Він з'являється у вигляді готових списків відтворення. Кожен користувач може створити власний список каналів із представлених і додавати до нього нові

КОПІЯ – 2023

канали. Для більш комфортного перегляду програма BooRadio може працювати в повноекранному режимі.

BooRadio має доступ до величезної кількості радіостанцій. Програма автоматично налаштує сортування по країнах. Є підтримка скінів.

Єдине, що вам знадобиться для запуску програми, це встановлений Microsoft.NET Framework 2.0. Для встановлення програми та її подальшої роботи не потрібно реєструватися. До того ж програма легко встановлюється – просто завантаживши її на комп'ютер, ви можете відкрити та використовувати. Ця програма не займає багато місця, її розмір становить всього 655 Кб. Інтерфейс BooRadio дуже простий і зручний.

Додаток працює на всіх версіях операційної системи Windows.

Єдиний його недолік – інтерфейс тільки англійською мовою. Але це легко компенсується простотою використання.

RadioSure

RadioSure – простий в установці, дуже надійний і простий плеєр, що дозволяє слухати радіо через інтернет, який має зручний інтерфейс з дуже великою базою радіостанцій.

На даний момент база плеєра становить понад 20 тисяч радіостанцій, велика кількість з яких українською мовою. Всі радіостанції записуються у вигляді таблиці, яка містить інформацію про якість трансляції. Список згруповано за жанрами та країнами, вказано мову трансляції.

Натиснувши на назву категорії, ви можете відсортувати таблицю, наприклад, за країною чи стилем.

Всі радіостанції поділяються на жанри:

- 70-ті / 80-ті / 90-ті;
- попси;
- танець.
- реп;
- хіп хоп;

КБПЗ – 2023

– R & B;

– рок (поділяється на підкатегорії: важкий, класичний, готичний, альтернативний);

– електронна музика (розділена на підкатегорії: транс, техно, лаунж, хаус, драм і бас);

– блюз;

– хітове радіо;

– латинь.

– джаз;

– класичний;

– дитячий.

– християнський.

– фольклор.

Переваги RadioSure:

– може розпізнавати та показувати назви пісень;

– приймає інтернет-радіо різних форматів: mp3, ogg vorbis, wma, aac +;

– швидко знаходити для вас улюблені радіостанції;

– плавно та зручно перемикати станції за допомогою ефекту перехресного переходу;

– запише ефір на диск, розділивши записи по треках;

– може одночасно записувати декілька радіостанцій (безкоштовна версія записує дві радіостанції);

– програма, а також база радіостанцій оновлюються автоматично;

– плеєр повністю підтримує UNICODE – це відображає всі ваші місцеві станції та шукає їх;

– RadioSure легко налаштувати та має функцію запису.

– економне використання трафіку.

КОПЗ – 2023

All-Radio

All-Radio – це багатофункціональна програма, яка дозволяє слухати радіостанції і дивитися телепередачі онлайн. Програма All-Radio завдяки своїй простоті у використанні автоматично шукає телеканали та радіостанції під час запуску.

Функції ALL-RADIO

- 5 підключених серверів баз даних, що містять теле- та радіоканали на будь-який смак користувача;
- понад 9000 радіостанцій і 1700 телеканалів, перелік яких постійно оновлюється;
- пошук потрібного каналу по країні та регіону;
- можливість запису трансляцій в аудіофайли і потокове відео різних форматів;
- сумісний з будь-якою версією Windows.

Переваги ALL-RADIO

- простий і зручний інтерфейс програми, що не викликає труднощів у використанні;
- широкий вибір теле- та радіоканалів різної спрямованості;
- можливість роботи у фоновому режимі;
- багатомовна підтримка;
- обмеження запису аудіо- та відеоматеріалів за гучністю;
- додаткові функції (захист паролем, додавання улюблених каналів, підтримка проксі-серверів і деякі інші).

All-Radio – це найзручніший і найшвидший спосіб переглядати популярні телеканали та слухати улюблені радіостанції через Інтернет. При цьому програма поширюється абсолютно безкоштовно і безкоштовно; не потребує реєстрації та оновлень.

КОПЗ – 2023

PC-RADIO

Це один з найпопулярніших плеєрів для прослуховування радіо в інтернеті. Плеєр відмінно працює не тільки на високих швидкостях. Користувачі з інтернетом 3G, EDGE 2G, а також супутниковим інтернетом (з прискорювачами globax, slonax) на тарифах анлім і серфінг також можуть відчути всю принадність цієї програми. Якщо завантажити PC-RADIO безкоштовно, то можна слухати радіо навіть на низькій швидкості інтернету. При цьому трафік майже не витрачається, а якість звуку ідеальна. Це стало можливим завдяки розробникам, які застосували спеціальне кодування потоку.

При запуску програма автоматично підключається до сервера і отримує всі необхідні оновлення. У цьому випадку список радіостанцій у списку відтворення оновлюється. Тепер за допомогою PC-RADIO ви можете слухати рок радіо, клубне радіо, ретро, поп, джаз, класику, дитяче, гумор, релакс і т.д. При цьому канали є інтерактивними, а це означає, що будь-який трек з бази надані на сайті, можна замовити в ефірі.

Також користувачі, які завантажили PC-RADIO, можуть переглядати відео та грати в популярні онлайн-ігри.

Інтерфейс PC-RADIO є різними мовами, кришка яскрава та інтуїтивно зрозуміла. Праворуч перераховані найпопулярніші радіостанції з невеликим описом стилю відтворюваних треків. Зліва – короткий список додаткових функцій.

Особливості нової версії:

- Яскравий зовнішній вигляд.
- Можливість запису потокової передачі.
- Можливість роботи через проксі-сервер.
- Покращений список уподобань.
- Можливість перемикати якість потоків.
- Таймер вимкнення.

PCRADIO сумісний з Windows 10/11.

КОПІЯ 2023

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 – використовучи 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.

КОПЗ-2023

− 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 TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

− 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 стосуться, 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.

КОПЗ 2023

– 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.

КОПІЯ 2023

Це розширює потужність конструкцій records в Delphi, які використовуться 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 – використовучи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існучим 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 відладочними 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 відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

КПЗ-2023

Підвищення якості й швидкодії інструментів

– Велика кількість поліпшень STL від Dinkumware.

– Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.

– Поліпшена підтримка Cmake.

– Велика кількість виправлень для підвищення стабільності і якості.

– Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.

– Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуться під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентів на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовуть стиль.

КПЗ-2023

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 TMemo 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.

КПЗ - 2023

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи клієнтського модуля Інтернет-радіо.

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) сформувати висновки про виконаний обсяг робіт та одержані результати.

КПЗ – 2023

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Протоколи Інтернет-радіо

Стек протоколів H.323 є одним з найпоширеніших на сьогодні. Це найстарший і найбільш стабільний із всіх використовуваних зараз протоколів, тому він вимагає особливої уваги.

Протокол ініціації сесій (SIP) – це відносно новий протокол, що одержує широке поширення. Він є значно більше молодим відносно H.323 і тому поки не одержав такого ж масштабного поширення.

Протокол MGCP – це керуючий Інтернет-радіо-протокол, що найбільше часто використовується для керування шлюзами в Інтернет-радіо-мережі.

Відносно новий протокол MGCP одержав широке поширення як частина архітектури Cisco AVVID.

AVVID звичайно використовує саме MGCP у зв'язуванні з CCM для керування шлюзами.

H.323

Як уже говорилося, H.323 є набором протоколів. Всі пристрої, використовувані H.323, можна поділити на чотири категорії: термінали, шлюзи, гейткіпери (Gatekeeper – гейткіпер) і точки багатопунктового контролю (Multipoint Control Unit – MTU).

Термінали, також називані кінцевими точками (endpoints), надають користувальницький інтерфейс до протоколу H.323 і забезпечують двосторонній мультимедійний зв'язок реального часу. Шлюзи виконують роль "перекладачів" для забезпечення взаємодії між H.323 і не-H.323 сутностями. Шлюзи, так само як і термінали, розглядаються як кінцеві точки. Гейткіпери виконують функції контролю викликів, такі як трансляція адрес і керування займаною смугою

КПЗ – 2023

пропущення. Гейткіпери можна вважати найбільш важливим компонентом у стеці H.323. MCU забезпечують можливість конференцій.

Стек протоколів H.323

IP, TCP, UDP

Протоколи IP, TCP і UDP безсумнівно є протоколами стека TCP/IP, але вони тут розглядаються тому, що надають транспортний сервіс для стека протоколів H.323.

Кожний термінал, шлюз, гейткіпер і MCU повинен мати свій унікальний IP-адресу. Це також ставиться й до ПК із додатками, які використовують H.323. IP надає кожній точці H.323-адресу й забезпечує механізм маршрутизації H.323-пакетів у мережі. TCP використовується для встановлення початкового з'єднання між терміналами H.323 і шлюзами/гейткіперуми. Протокол UDP використовується для передачі безпосередньо голосу через мережу.

H.225

H.225 забезпечує встановлення й контроль викликів з усією необхідною сигналізацією для здійснення з'єднання між двома кінцевими точками.

H.245

Керуюча сигналізація H.245 застосовується для узгодження використання каналу й можливостей. Керуючі повідомлення несуть інформацію, що ставиться до наступних моментів:

– обмін інформацією про доступні можливості;

– відкривання й закривання логічного каналу, використовуваного для медійного потоку;

– повідомлення керування потоком;

– загальні команди.

Після встановлення виклику всі процеси передачі інформації проходять по логічних каналах.

RAS

RAS – це протокол, що використовується між кінцевими точками

КПІ – 2023

(терміналами й шлюзами) і гейткіперуми. Він застосовується для здійснення реєстрації, контролю доступу, статусу й змін доступної смуги пропущення, а також для відключення кінцевих точок від гейткіперу. RAS використовує порт UDP 1719.

RTP

RTP надає наскрізний мережний транспорт для додатків, що передають дані реального часу. Він використовує для передачі даних протокол UDP. Передача даних супроводжується керуючим протоколом (RTCP) для моніторингу доставки даних.

Кодеки

Кодеки використовуються не тільки протоколом H.323, а всіма протоколами Інтернет-радіо для визначення алгоритмів компресії й декомпресії, застосовуваних для передачі аудіо/відео по мережі. H.323 підтримує більшість стандартів кодування аудіо й відео, включаючи G.7XX для аудіо й H.26X для відео.

Етапи з'єднання

Процедури з'єднання за протоколом H.323 можуть бути згруповані в п'ять етапів:

1. Виявлення й реєстрація.
2. Установлення виклику.
3. Сигнальний потік.
4. Медійний потік і потік керування.
5. Завершення виклику.

Виявлення й реєстрація пристроїв

На стадії виявлення й реєстрації гейткіпер ініціює процес "розвідування" для визначення гейткіперу, з яким кінцева точка повинна взаємодіяти. Це може бути зконфігуровано статично, або процес виявлення може відбуватися за допомогою багатоадресних повідомлень. Після виявлення термінал або шлюз реєструється на заданому гейткіпері.

КПЗ – 2023

Реєстрація використовується кінцевою точкою для визначення зони, з якої вона може бути асоційована. Зона – це набір компонентів, керованих одним гейткіпером. Після визначення зони гейткіпер інформується про адресу зони.

Внутрізонові виклики

Допустимо, шлюзи (термінали) уже зареєстровані, і шлюз X хоче зробити виклик на термінал, підключений до шлюзу Y. Шлюз X посилає ARQ (Admission Request) повідомлення гейткіперу, запитуючи дозвіл на встановлення виклику на телефон, що обслуговується шлюзом Y. Гейткіпер дозволяє виклик з вимогою сигналізації (дозвону) прямо, посилаючи повідомлення ACF (Admission Confirmation).

Міжзонові виклики

Допустимо, гейткіпер A контролює зону A, а гейткіпер B – зону B, шлюз X зареєстрований на гейткіпері A, а шлюз Y – на гейткіпері B, і шлюз A хоче встановити з'єднання з терміналом, підключеним до шлюзу Y.

Процес установлення виклику містить наступні етапи (рисунок 2.8).

1. Шлюз X запитує з'єднання зі шлюзом Y у свого локального гейткіперу.

2. Запит місця розташування (LRQ – Location request). Гейткіпер шлюзу X не знає IP-адресу шлюзу Y і запитує адресу в гейткіперу шлюзу Y.

3. Місце розташування підтверджене (LCF – Location confirm). Гейткіпер шлюзу Y відповідає IP-адресою шлюзу Y.

4. Гейткіпер шлюзу X підтверджує його запит і надає йому IP-адресу шлюзу Y.

5. Установлення з'єднання між шлюзами.

Установлення з'єднання

На етапі встановлення з'єднання шлюзи взаємодіють прямо. Альтернативний метод установлення з'єднання – це гейткіпер-маршрутизуєма сигналізація, при якій всі повідомлення по установці з'єднання проходять через гейткіпер.

Установлення з'єднання засноване на протоколі ITU-Q.931 (H.225 є

КОПІЯ 2023

підмножиною Q.931), що визначає метод установлення, обробки й завершення мережного з'єднання по цифровій мережі ISDN. Процес складається із шести фаз:

1. Шлюз X посилає H.225 повідомлення встановлення дозвону для запиту з'єднання.

2. Шлюз Y посилає обернено H.225 повідомлення, заявляючи про можливість продовження процесу.

3. Шлюз Y запитує в гейткіперу правомірність дзвінка, посилаючи йому RAS-повідомлення (ARQ) по каналі RAS.

4. Гейткіпер підтверджує, що дзвінок правомірний, посилаючи шлюзу Y ACF-повідомлення.

5. Шлюз Y посилає H.225-повідомлення шлюзу X, сповіщаючи його, що з'єднання встановлене.

6. Шлюз Y посилає H.225-повідомлення шлюзу X, сповіщаючи його, що виклик установлений.

Установлення логічних каналів

Після того як з'єднання встановлене, взаємодія відбувається по логічних каналах. H.245 використовується для визначення процесу керування цими каналами. На один виклик може доводитися кілька каналів для різних типів трафіку (відео, аудіо, дані). H.245 LCSE (Local Channel Signaling Entity) відкриває логічний канал для кожного потоку. Канали можуть бути як односпрямованими, так і двонаправленими.

Установлення логічних каналів відбувається в такий спосіб:

1. Шлюз X повідомляє шлюз Y, які можливості він підтримує, посилаючи H.245 Terminal Capability Set повідомлення.

2. Шлюз Y підтверджує запит, посилаючи H.245 Terminal Capability Set Acknowledge повідомлення.

3. Аналогічний п.1, але тільки у зворотному напрямку.

4. Аналогічний п.2, але тільки у зворотному напрямку.

5. Шлюз X відкриває медіаканал зі шлюзом Y, посилаючи H.245

КОПЗ-2023

повідомлення Open Logical Channel, включаючи адресу RTCP каналу.

6. Шлюз Y підтверджує встановлення логічного каналу зі шлюзом X, посилаючи H.245-повідомлення Open Logical Channel Acknowledge, включаючи RTP-адресу, виділена шлюзом Y, і RTCP-адресу, отримана від шлюзу X.

7. Аналогічний п.5, але тільки у зворотному напрямку.

8. Аналогічний п.6, але тільки у зворотному напрямку.

Медійний потік і потік керування

Медійний потік управляється RTCP. RTCP використовує виділений логічний канал для кожного RTP-потоку. Кінцеві точки можуть спробувати змінити виділену смугу пропущення, що вони споконвічно запросили. Для збільшення виділеної смуги пропущення кінцеві точки повинні запросити на цей дозвіл у гейткіперу.

Завершення виклику зупиняє медіапоток і закриває логічні канали. Воно може бути запитане як кінцевими точками, так і гейткіпером. Завершення виклику також завершує H.245-сесію, звільняє H.225/Q.931 з'єднання й надає гейткіперу підтвердження про роз'єднання по RAS.

Сигналізація між кінцевими точками без посередника в H.323

Якщо шлюзи знають IP-адреси один одного, то можливо їхня взаємодія без гейткіперу. Цей процес можна описати наступними кроками:

1. Шлюз ініціює H.225.0-сесію зі шлюзом призначення.

2. Процедура встановлення виклику, що базується на Q.931, створює сигнальний канал між кінцевими точками.

3. Кінцеві точки відкривають канал для функцій керування H.245. Відбувається обмін можливостями й дескрипторами логічних каналів.

4. Відкривається RTP-сесія.

MGCP

Протокол MGCP являє собою приклад моделі із централізованим керуванням викликами. Він визначає керування телефонними шлюзами із центрального керуючого компонента, називаного телефонним агентом (Call

КОПІЯ 2023

Agent). Шлюзи взаємодіють із агентами, які здійснюють сигналізацію й обробку викликів.

Компоненти MGCP

В MGCP-оточенні використовуються наступні компоненти:

– кінцеві точки;

– шлюзи;

– агент.

Кінцеві точки – це точки з'єднання пакетної мережі. Вони можуть бути фізичними й логічними. Шлюзи – це вузли об'єднання кінцевих точок.

Агент MGC (Media Gateway Controller) являє собою центральний керуючий елемент в MGCP-оточенні. MGC здійснює керування діяльністю шлюзів у припущенні, що шлюзи фіксують події й доповідають про їх. Агент, ґрунтуючись на подіях, інструктує шлюзи про дії, які необхідно вживати. Він також ініціює всі Інтернет-радіо-етапи з'єднання.

Поняття MGCP

Базові поняття MGCP:

– Виклики й з'єднання. Дозволяють установлювати наскрізні з'єднання двох і більше кінцевих точок.

– Події й сигнали. Дозволяють телефонним агентам інструктувати шлюзи.

– Цифрові карти й пакети. Дозволяють шлюзам визначати пункт призначення викликів.

Взаємодія агентів і шлюзів

Процес взаємодії телефонного агента зі шлюзами для забезпечення телефонного виклику можна описати наступною послідовністю дій.

1. Агент направляє повідомлення RQNT (Request Notification) кожному шлюзу. Цей запит дає інструкцію шлюзам чекати події off-hook і дати гудок, коли така подія відбудеться. Агент також повідомляє про необхідність моніторингу інших подій. Надаючи цифрову карту в запиті, агент дозволяє шлюзам зібрати цифри перед тим як інформувати про подію агента.

КПЗ – 2023

2. Шлюз відповідає на запит. Із цього моменту агент і шлюзи чекають подій.

3. Шлюз А посилає оповіщення (NTFY) агентові, повідомляючи його, що необхідна подія відбулася.

4. Після підтвердження можливості агент інструктує шлюз А створити з'єднання (CRCX) з його кінцевою точкою.

5. Шлюз відповідає дескриптором сесії. Дескриптор визначає, як мінімум, IP-адресу й UDP-порт для наступної RTP-сесії. Шлюз не має дескриптора сесії віддаленої сторони, і з'єднання переходить у режим очікування.

6. Агент відправляє запит на з'єднання шлюзу В. У запиті агент надає дескриптор сесії, що він одержав від шлюзу А. Агент також посилає інструкції про те, які в цей момент події важливі і які сигнали шлюзу генерувати. У цьому випадку такою подією є off-hook, сигналом – дзвінок.

7. Шлюз В відповідає на запит і повідомляє свій дескриптор сесії.

8. Агент передає дескриптор сесії шлюзу А в запиті MDCX (Modify Connection). Тепер шлюзи можуть установити RTP-сесії для передачі голосу.

9. Наприкінці виклику одна з кінцевих точок розпізнає перехід у стан on-hook. Допустимо, це трапилося на шлюзі А. Тому що агент проінструктував повідомити про таку подію, шлюз А посилає агентові повідомлення.

10. Агент розсилає повідомлення DLCX (Delete Connection) кожному шлюзу.

11. Шлюзи видаляють з'єднання й відповідають.

3.2 Розробка структурної схеми

Типова система передачі потокових аудіоданих через Інтернет складається із трьох базових елементів:

– станції – пристрої, що генерує звуковий потік (у відповідності зі списком звукових файлів або шляхом прямій оцифровки аналогового потоку від аудіокарти або мікрофона) і напрямного його на адресу сервера;

КПЗ – 2023

– сервера (повторювача) – пристрою, що приймає звуковий потік від станції й перенапрямної його копії всім підключеним до нього користувачам Інтернету;

– клієнта – пристрою, що приймає звуковий потік від сервера й перетворюючого його в аудіосигнал, доступний слухачеві інтернет-радіостанції.

Структурна схема інтернет-радіо представлена на рисунку 3.1.

Функціонування сервера й клієнта інтернет-радіо забезпечується типовими програмно-технічними засобами, тому нижче розглядаються тільки питання організації роботи станції інтернет-радіо як найбільш складного й відповідального компонента. Крім потоку звукових даних, станція передає текстові метадані – наприклад, інформацію про себе саму й про поточну композицію, що пропонується для прослуховування клієнтові. Як станція можуть виступати комп'ютер із установленої на ньому програмою-аудіоплеєром і плагіном-кодеком (або спеціалізованою програмою) або професійний апаратний пристрій, що перетворить аналоговий звуковий потік у цифровий сигнал.

З огляду на вимоги до надійності роботи станції, описана нижче технічне рішення припускає використання спеціалізованого професійного встаткування, вартість якого цілком порівнянна з вартістю сучасної комп'ютерної системи. Разом з тим комп'ютер може застосовуватися в складі станції як засіб формування призначених для трансляції матеріалів, а також інтерактивної взаємодії зі слухачами. Дане рішення забезпечує висока якість віщання по відносно ненадійних каналах зв'язку й розраховано на професійні студії, використовувані широкомовними радіостанціями, хоча з його допомогою організувати трансляцію можуть і індивідуальні користувачі.

Використовуване професійне встаткування призначене для організації обміну звуковою інформацією з мереж передачі даних з використанням протоколу IP. Перетворення аудіосигналу для його передачі на адресу сервера по IP-Каналі виробляється за допомогою аудіокодеку Instreamer, до якого можуть підключатися різні джерела сигналу, включаючи мікрофон і аудіопрогравач.

КПЗ – 2023

Аудіодані, що надходять від сервера, перетворяться за допомогою аудіокодеку Exstreamer і виводяться на навушники або гучномовець. Паралельно вони можуть записуватися в цифровому або аналоговому виді на відповідний пристрій.

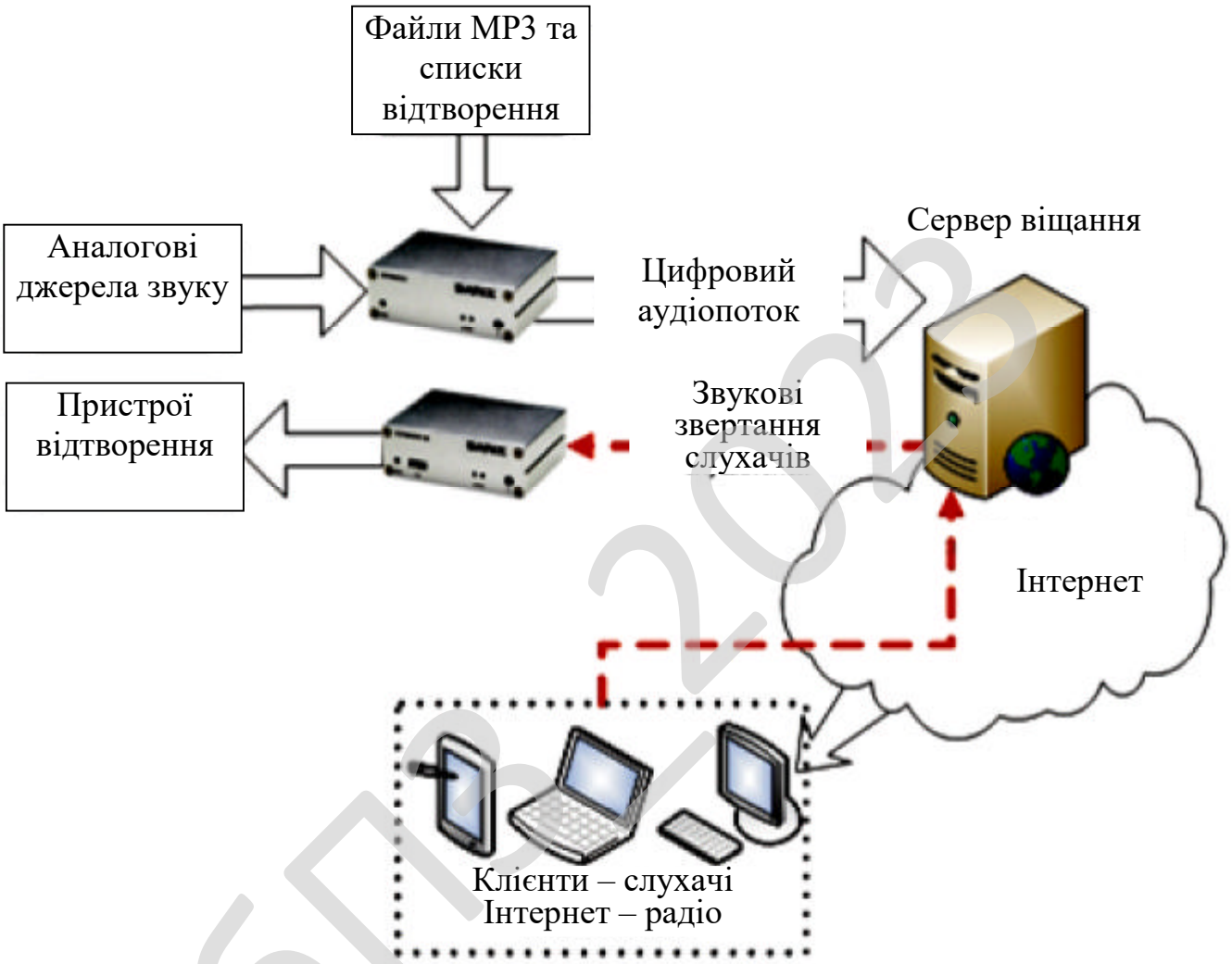

Рисунок 3.1 – Структурна схема системи

У розглянутій конфігурації станція інтернет-радіо дозволяє реалізувати наступні основні функції:

– підготовку на комп'ютері звукових матеріалів;

– їхню реєстрацію й зберігання в енергонезалежній пам'яті передавального пристрою;

КОПІЯ 2023

– трансляцію з високою якістю заздалегідь підготовлених звукових матеріалів у заданій послідовності з дотриманням часової сітки віщання;

– обмін звуковими повідомленнями, включаючи повідомлення з мікрофона, з використанням протоколу IP у реальному часі;

– реєстрацію й зберігання цифрових повідомлень, що надходять від сервера.

Статистичний аналіз роботи інтернет-радіо здійснюється з використанням відповідних програм на вхідному в його состав комп'ютері, а стандартні можливості застосовуваних пристроїв можна розширити за допомогою високорівневого мови програмування Audio Barix Control Language, що дозволяє писати додатка для різних аудіокодеків.

Аудіокодек Instreamer являє собою багатопротокольний перетворювач високоякісних звукових стереосигналів у потік цифрових даних, переданих по мережах IP. Він підтримує формати аудіоданих MP3, PCM, G.711 і G.722, які транслюються в цифрових форматах TCP, UDP, Shoutcast/Icecast і Multicast RTP; забезпечує передачу потокових звукових сигналів з мінімальною затримкою, що гарантує високу якість звуку; має функцію трансляції музики у фоновому режимі й тимчасовій зупинці музичної трансляції для реалізації додатків IP-телефонії.

Аудіокодек Exstreamer являє собою багатопротокольний перетворювач IP-потоку аудіоданих у форматах TCP, UDP і Multicast RTP у звуковий сигнал для його виводу на навушники, гучномовець або підсилювач у форматах AACplus, MP3, Ogg Vorbis, G.711, PCM (лінійне декодування). Перетворювачі, що випускаються, адаптовані для роботи з різними зовнішніми пристроями в заданих конфігураціях. Вони можуть використовуватися як приймач і програвача інтернет-радіо, мати рознімання для карти пам'яті стандарту MicroSD або інтерфейс USB для підключення зовнішньої флеш-пам’яті.

КПІ – 2023

3.3 Розробка функціональної схеми

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

Використання з'єднання в режимі пакетної передачі даних для прослуховування Інтернет-радіостанцій може привести до істотного збільшення обсягу даних, переданих через мережу постачальника послуг. Додаткову інформацію про вартість послуг передачі даних можна одержати в постачальника послуг.

Прослуховування станцій Інтернет-радіо

Для прослуховування станцій Інтернет-радіо виконаєте наступні дії:

1. Виберіть станцію зі списку вибраних або з каталогу радіостанцій або виконаєте пошук станцій за назвою в службі Інтернет-радіо.

Для додавання радіостанції вручну виберіть Функції > Додати станції вручну. Можна шукати посилання на радіостанцію за допомогою Інтернет-браузера. Сумісні посилання автоматично відкриваються в додатку Інтернет-радіо.

2. Виберіть Функції > Слухати.

Відкривається екран "Поточний список" з інформацією про радіостанцію, що прослуховується, і композиції.

Для припинення відтворення натисніть клавішу прокручування, для поновлення відтворення натисніть клавішу прокручування ще раз.

Для перегляду інформації про радіостанцію виберіть Функції > Інформація про станцію (дана функція недоступна, якщо станція була збережена вручну).

При прослуховуванні станції, збереженої в списку "Вибране", виконаєте прокручування вліво або вправо для переходу до попередньої або наступної збереженої станції.

КПЗ_2023

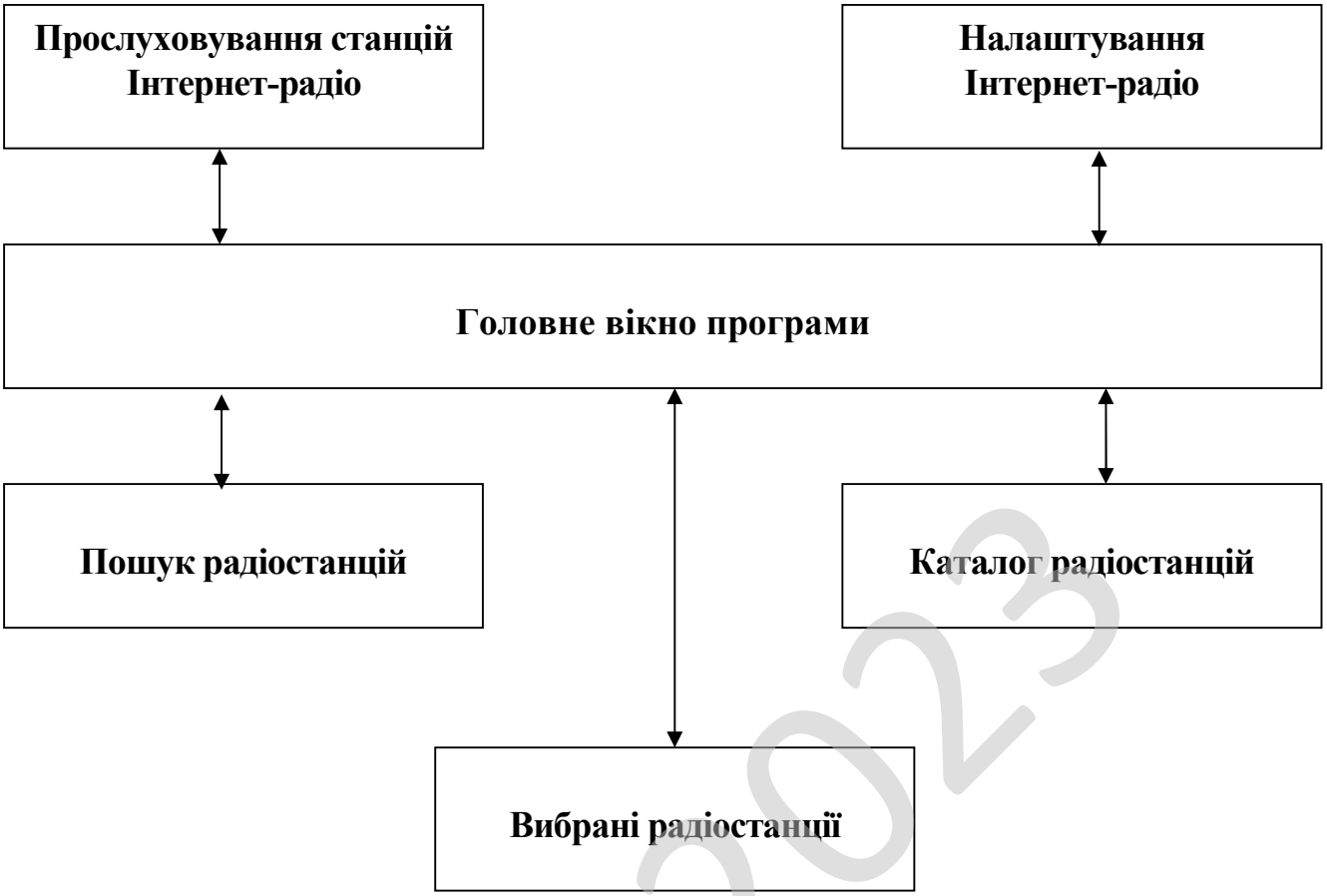

Рисунок 3.2 – Функціональна схема системи

Каталог радіостанцій

Для прослуховування радіостанцій Інтернет-радіо, що не входять у каталог, додайте вручну інформацію про радіостанцію або знайдіть посилання на радіостанцію в Інтернеті за допомогою Інтернет-браузера.

Виберіть одну з наступних функцій:

– Пошук за жанром – перегляд жанрів доступних радіостанцій.

– Пошук за мовою – перегляд мов віщання радіостанцій.

– Пошук за країною/регіоном – перегляд країн, у яких ведеться віщання радіостанцій.

– Кращі станції – перегляд найбільш популярних радіостанцій у каталозі.

КПЗ – 2023

Пошук радіостанцій

Для пошуку радіостанцій у службі Інтернет-радіо за назвою, необхідно виконати наступні дії:

1. Виберіть Пошук.

2. У поле пошуку введіть назву радіостанції або перші букви назви й виберіть Функції > Пошук.

На дисплеї з'явиться список відповідних радіостанцій.

Для прослуховування радіостанції виділите її й виберіть Функції > Слухати.

Для збереження радіостанції в списку вибраного виділите її й виберіть Функції > Додати в "Вибране".

Для виконання ще одного пошуку виберіть Функції > Повторити пошук.

Вибрані радіостанції

Для перегляду й прослуховування вибраних станцій виберіть Вибране.

Для додавання радіостанції в список вибраного вручну виберіть Функції > Додати станцію вручну. Уведіть Інтернет-адресу радіостанції й назва, що буде відображатися в списку вибраних.

Для додавання поточної відтвореної радіостанції в список вибраного виберіть Функції > Додати в "Вибране".

Для перегляду інформації про радіостанції, переміщення радіостанції в списку нагору або долілиць або видалення радіостанції зі списку вибраного виберіть Функції > Станція й необхідна функція.

Для перегляду тільки радіостанцій, назва яких починається з певних букв або цифр, починайте вводити символи. На дисплеї з'явиться список відповідних радіостанцій.

Налаштування Інтернет-радіо

Щоб вибрати крапку доступу за замовчуванням для з'єднання з мережею, виберіть Крапка доступу за замовчуванням і доступні функції. Виберіть Завжди запитувати, якщо необхідно, щоб крапка доступу запитувалася щораз при відкритті додатка.

КПЗ 2023

Якість радіомовлення залежить від обраної швидкості з'єднання. Чим вище швидкість, тим краще якість. Щоб уникнути буферизації використовуйте найвищу якість тільки для високошвидкісних з'єднань.

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

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3.

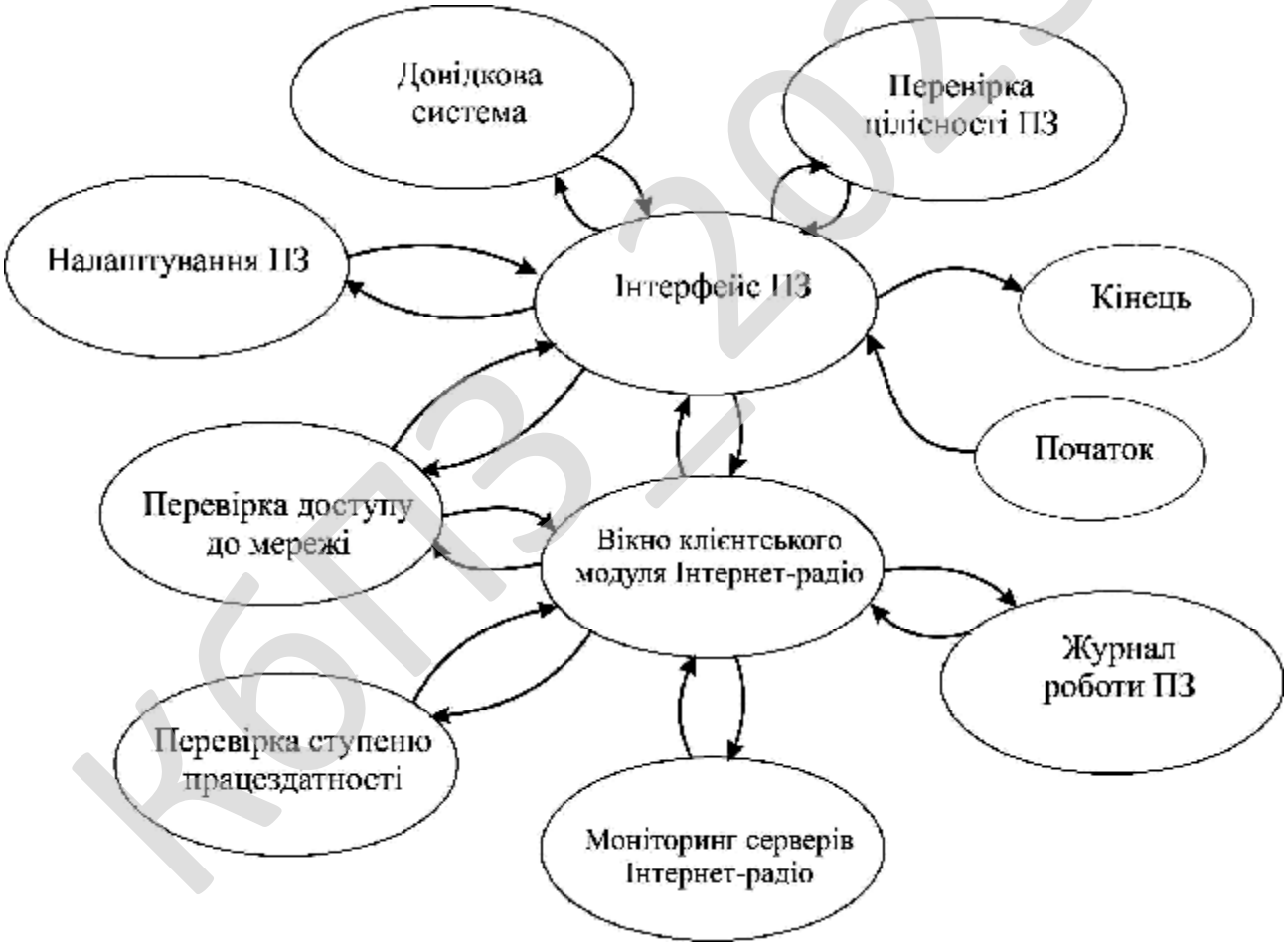

Рисунок 3.3 – Діаграма взаємодії процесів

КПЗ-2023

Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

– Інтерфейс ПЗ.

– Перевірка цілісності ПЗ.

– Довідкова система.

– Налаштування ПЗ.

– Перевірка доступу до мережі.

– Вікно клієнтського модуля Інтернет-радіо.

– Журнал роботи ПЗ.

– Моніторинг серверів Інтернет-радіо.

– Перевірка ступеню працездатності.

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.

КбПЗ_2023

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми. З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем я враховував, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю клієнтського Інтернет-радіо.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

КПЗ-2023

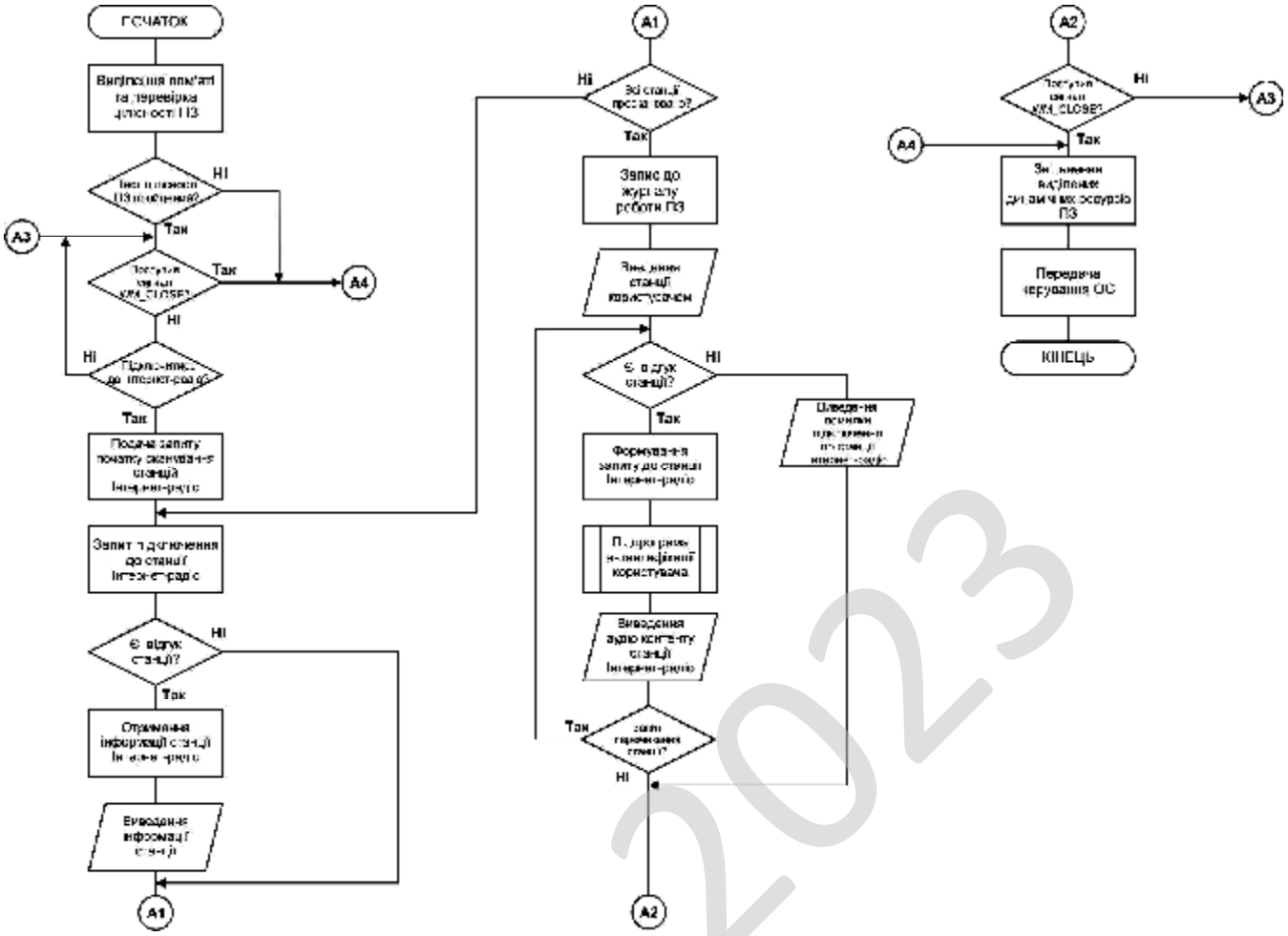

Рисунок 4.1 – Блок схема основної програми

Опис алгоритмів функціонування системи

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

КПЗ - 2023

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм.

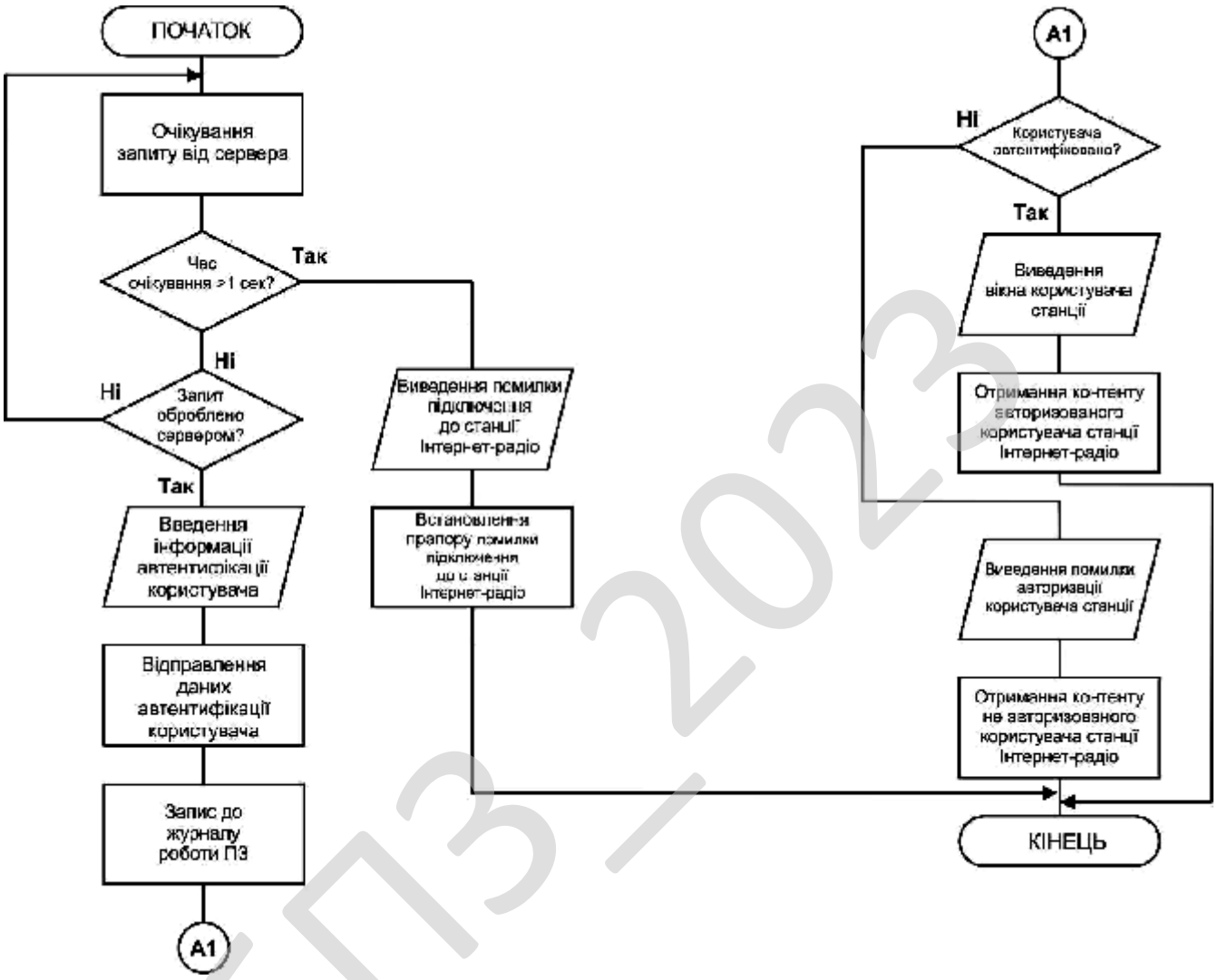

Рисунок 4.2 – Блок схема підпрограми

Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

КПЗ – 2023

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об’єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

– Керівникам проектів, які керують розподілом завдань і контролем за проектом.

– Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.

– Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.

– Програмістам які реалізовують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

Обґрунтування інформаційного забезпечення та перелік первісних даних

Під вхідною інформацією розуміється вся інформація, необхідна для вирішення задачі і розташована на різних носіях: первинних документах,

КПЗ – 2023

машинних носіях, у пам'яті персонального комп'ютера. Вхідною інформацією для розроблювальної в дипломному проекті ПЗ клієнтського модуля Інтернет-радіо є мова або аудіодані.

Щоб передати мову через мережу, мовну інформацію потрібно перетворити в аналоговий електричний сигнал. При переході до цифрових мереж зв'язку виникла необхідність перетворити аналоговий електричний сигнал у цифровий формат на передавальній стороні, тобто закодувати, і перевести назад в аналогову форму, тобто декодувати, на прийомній стороні.

Процес перетворення аналогового мовного сигналу в цифрову форму називають аналізом або цифровим кодуванням мови, а зворотний процес відновлення аналогової форми мовного сигналу – синтезом або декодуванням мови.

Ціль будь-якої схеми кодування – одержати таку цифрову послідовність, що вимагає мінімальної швидкості передачі й з якої декодер може відновити вихідний мовний сигнал з мінімальними змінами.

При перетворенні мовного сигналу в цифрову форму, мають місце два процеси – дискретизація, тобто формування дискретних у часі відрахунків амплітуди сигналу, і квантування, тобто дискретизація отриманих значень за амплітудою. Ці дві функції виконуються т.зв. аналого-цифровими перетворювачами (АЦП), які розміщаються в сучасних АТС на платі абонентських комплектів, а у випадку передачі мови по IP-мережах – у терміналі користувача (комп'ютері або IP-телефоні).

Процес аналого-цифрового перетворення одержав, стосовно до систем зв'язку, назву імпульсно-кодової модуляції (ІКМ).

Щоб знизити необхідну швидкість передачі біт, застосовують нелінійний (логарифмічний) закон квантування, тобто квантуванню піддається не амплітуда сигналу, а її логарифм. У цьому випадку має місце процес «стиску» динамічного діапазону сигналу, а при відновленні сигналу відбувається зворотний процес.

Сьогодні застосовуються два основні різновиди ІКМ: з кодуванням по m-

КПЗ-2023

закону й по А-закону. У результаті стиску сигнал з амплітудою, що кодується 12-13 бітами, описується всього вісьма бітами.

Розрізняються ці різновиди ІКМ деталями процесу стиску (m-закон кодування переважніше використовувати при малій амплітуді сигналу й при малому відношенні сигнал/шум). У Північній Америці використовується кодування по m-закону, а в Європі – по А-закону. Тому при міжнародному зв'язку в багатьох випадках потрібне перетворення m-закону в А-закон, відповідальність за яке несе країна, у якій використовується m-закон кодування. В обох випадках кожний відлік кодується 8 бітами, або одним байтом, який можна вважати звуковим фрагментом.

Для передачі послідовності таких фрагментів необхідна пропускна здатність каналу, рівна 64 Кбіт/с. Оскільки ІКМ була першою стандартною технологією, що одержала широке застосування в цифрових системах передачі, пропускна здатність каналу, рівна 64 Кбіт/с, стала всесвітнім стандартом для цифрових мереж всіх видів, причому – стандартом, що забезпечує передачу мови з дуже гарною якістю.

Однак така висока якість передачі мовного сигналу (що є еталоном при оцінці якості інших схем кодування) досягнута в системах ІКМ за рахунок явно надлишкова, при сучасному рівні технології, швидкості передачі інформації.

Щоб зменшити властиву ІКМ надмірність і знизити вимоги до смуги пропущення, послідовність чисел, отримана в результаті перетворення мовного аналогового сигналу в цифрову форму, піддається математичним перетворенням, що дозволяють зменшити необхідну швидкість передачі. Ці перетворення «сирого» цифрового потоку в потік меншої швидкості називають «стиском» (а часто – кодуванням, розглядаючи ІКМ як якусь відправну точку для подальшої обробки інформації). Існує безліч підходів до «стиску» мовної інформації; всі їх можна розділити на три категорії: кодування форми сигналу (waveform coding), кодування вихідної інформації (source coding) і гібридне кодування, що представляє собою сполучення двох підходів.

КПЗ – 2023

Опис кодування форми сигналу

Імпульсно-кодова модуляція, по суті, і являє собою схему кодування форми сигналу. Однак цікавлять більш складні алгоритми, що дозволяють знизити вимоги до смуги пропущення.

Розглянуті методи кодування форми сигналу використовують ту обставину, що між випадковими значеннями декількох послідовних обчислень існує деяка залежність. Це дозволяє з досить високою точністю пророчити значення будь-якого відліку на основі значень декількох попередніх йому обчислень.

При побудові алгоритмів кодування названа закономірність використовується двома способами. По-перше, є можливість змінювати параметри квантування залежно від характеру сигналу. У цьому випадку крок квантування може змінюватися, що дозволяє певною мірою згладити протиріччя між зменшенням числа біт, необхідних для кодування величини відліку при збільшенні кроку квантування, і звуженням динамічного діапазону кодера, неминучим без адаптації.

Деякі алгоритми передбачають зміну параметрів квантування приблизно в рамках вимовних складів, а деякі змінюють крок квантування на основі аналізу статистичних даних про амплітуду сигналу, отриманих за відносно короткий проміжок часу. По-друге, існує підхід, називаний диференціальним кодуванням або лінійним пророкуванням. Замість того, щоб кодувати вхідний сигнал безпосередньо, кодують різницю між вхідним сигналом і «передвіщеною» величиною, обчисленою на основі декількох попередніх значень сигналу. Описаний метод називається лінійним пророкуванням, тому що він використовує тільки лінійні функції попередніх обчислень.

Найпростішою реалізацією останнього підходу є так звана дельта-модуляція (ДМ), алгоритм якої передбачає кодування різниці між сусідніми обчисленнями сигналу тільки одним інформаційним бітом, забезпечуючи передачу, по суті, тільки знака різниці.

КПІ – 2023

Алгоритмом, побудованим на описані вище принципах, є алгоритм адаптивної диференціальної імпульсно-кодової модуляції (АДІКМ) (G.726). Алгоритм передбачає формування сигналу помилки пророкування і його наступне адаптивне квантування. При досить гарних характеристиках алгоритму, АДІКМ практично не застосовується для передачі мови по мережах з комутацією пакетів, тому що цей алгоритм дуже чутливий до втрат цілих блоків відліку, що відбуваються при втратах пакетів у мережі. У таких випадках порушується синхронізація кодера й декодера, що приводить до катастрофічного погіршення якості відтворення мови навіть при малій імовірності втрат.

Перелік вихідних даних

У ході розробки ПЗ клієнтського модуля Інтернет-радіо визначено, що вихідною інформацією є мова на пункті отримання інформації. Як може здатися на перший погляд, вузькополосне кодування мови, що вимагає обчислювальної потужності, є самим складним завданням, виконуваної устаткуванням Інтернет-радіо.

Однак це не так: алгоритми кодування мови стандартизовані й відмінно документовані, більше того, на ринку доступні досить ефективні їхні реалізації для всіх популярних DSP-платформ. З іншого боку, в устаткуванні Інтернет-радіо повинні бути реалізовані багато інших функцій, спосіб реалізації яких не є об'єктом стандартизації.

На передавальній стороні устаткування Інтернет-радіо працює за принципом «закодував, передав і забув». На прийомній стороні все набагато складніше. Пакети приходять із мережі із затримкою, що міняється за випадковим законом. Більше того, пакети можуть прийти не в тій послідовності, у якій були передані, а деякі пакети можуть взагалі бути загублені. Приймач повинен справлятися з усіма цими труднощами, забезпечуючи на виході нормальний звуковий потік з тактовою синхронізацією, або генерованим на основі прийнятого потоку даних, або одержуваним по каналах E1. Прив'язка мовних потоків до місцевого тактового сінхросигналу здійснюється шляхом непомітної на слух

КПЗ-2023

деформації періодів мовчання у відтвореному сигналі. До цього залишається додати необхідність передачі факсимільної інформації в реальному часі з автоматичним розпізнаванням сигналів факсимільних апаратів і передачу DTMF-сигналів з коректним їхнім відновленням у приймачі.

Сигнали багаточастотного набору номера (DTMF) – просто звукові сигнали, передані по телефонному каналі. При передачі їх по цифровій телефонній мережі не виникає ніяких проблем, тому що кодування за допомогою алгоритму G.711 не накладає ніяких обмежень на вид звукових сигналів – це може бути мова, сигнали модему, або тональні сигнали – всі будуть успішно відтворені на приймачі.

Вузькополосні кодеки, щоб досягти низьких швидкостей передачі, використовують той факт, що сигнал, який вони кодують, представляє саме мову. Сигнали DTMF при проходженні через такі кодеки спотворюються й не можуть бути успішно розпізнані приймачем на прийомній стороні.

Коли користувачеві потрібно ввести якусь додаткову інформацію у віддалену систему при вже встановленому з'єднанні, необхідно забезпечити можливість надійної передачі DTMF-сигналів через мережу Інтернет-радіо. У випадках, коли система, взаємодіюча з користувачем, просто ставить запитання й чекає введення, тривалість і момент передачі сигналу не важливі. В інших випадках система видає користувачеві список і просить його нажати, наприклад, кнопку «#», як тільки він почує потрібну інформацію; тут ситуація більше складна, і необхідна більше точна прив'язка вчасно.

Існуючі методи передачі сигналів DTMF по мережах Інтернет-радіо.

– Обов'язковий метод. Спеціальне повідомлення протоколу H.245 може містити символи цифр і «*», «#». У цьому випадку використовується надійне TCP-з'єднання, так що інформація не може бути загублена. Однак через особливості TCP можуть мати місце значні затримки;

– Нестандартний метод. Він може бути застосований у терміналах H.323v2 при використанні процедури fastStart і відсутності каналу H.245. Для передачі

КПІ – 2023

сигналів DTMF відкривається спеціальна RTP-сесія, у якій передаються кодовані значення прийнятих цифр, а також дані про амплітуду й тривалість сигналів. Може бути використана та ж сесія, що й для мови, але зі спеціальним типом корисного навантаження. Використання RTP дозволяє прив'язати DTMF– сигнали до реального часу, що є важливою перевагою даного методу.

У принципі, перший метод може бути більше кращим, однак у випадку міжнародних викликів і при використанні віддалених систем, що вимагають твердої прив'язки введення користувача до часу, може виявитися необхідним застосувати другий метод. Шлюзи Інтернет-радіо повинні обов'язково придушувати перекручені сигнали DTMF, що пройшли через основний мовний канал. У противному випадку, при відновленні сигналів, про які була прийнята інформація, можуть виникнути неприємні ефекти накладення й розмноження сигналів. На основі даного огляду функцій устаткування Інтернет-радіо можна зробити вивід, про тім що, незважаючи на існування стандартних алгоритмів кодування мови, у розроблювачів є величезний простір для діяльності, спрямованої на подальше вдосконалювання технології Інтернет-радіо.

Опис технологічного процесу обробки даних

Технологічний процес складається з двох основних етапів – збір і облік даних по IP-адресам й формування відповідної легенди та власне сама робота програми у режимі Інтернет-радіо та селективного зв’язку. Вони можуть виконуватися в будь-який календарний момент часу і включають операції введення, з’єднання і ін. Операції мають програмне виконання, підлегле єдиній алгоритмічній схемі.

Програма реалізована в середовищі Delphi XE7. Робота з програмою починається з виведення інформаційного вікна й активізації системи меню й здійснюється в діалоговому і по-дійному режиму.

При цьому під діалогом розуміється надання користувачу декількох альтернатив і обробка його вибору. У діалогову систему входять головне меню з відповідними спливаючими підменю а також діалогові вікна. Під подіями

КПЗ-2023

розуміються процеси, що активізуються користувачем (наприклад – натискання функціональних клавіш), а також програмні події – з'єднання по IP-технології.

Програма складається з наступних основних модулів. Основна процедура – конфігурація середовища оточення, формування основного екрана програми клієнтського модуля Інтернет-радіо, створення системи головного меню і відповідних підміню, активізація меню. Процедура обробки головного меню – запуск відповідної процедури. Процедура введення даних – забезпечення введення інформації станцій Інтернет-радіо у базу даних, контроль за допустимістю цих значень, забезпечення введення даних шляхом вибору зі списку. Допоміжні процедури і функції – реалізація запитів, повідомлень, формування списків вибору станції Інтернет-радіо, а також контроль за даними, що вводяться. Усі модулі в програмі зв'язані між собою за даними, що аналізуються на вході і виробляються на виході. Дані в модулі надходять через діалог з користувачем, параметри і документи інформаційної бази. Передача даних від одного модуля до іншого здійснюється тільки через збережені документи. Дані через діалог можуть бути отримані прямим і непрямим способом. Прямий спосіб реалізується шляхом їхнього введення за шаблоном чи по запиту конкретних значень реквізитів. Непрямий спосіб – шляхом чи меню логічних (альтернативних) запитів – «так», «ні». При непрямому способі дані, що надходять у модуль, заздалегідь передбачені алгоритмом, але зовні виглядають в обліку відомими фразами. Параметри (мова) – вхідні дані, отримані у виді конкретних значень, переданих в оперативній пам'яті суміжним модулям (функціям).

КБПЗ-2023

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою IDEA – симетричний блоковий алгоритм шифрування даних, запатентований швейцарською фірмою Ascom. Відомий тим, що застосовувався в

пакеті програм шифрування PGP. У листопаді 2000 року IDEA був представлений як кандидат у проекті NESSIE в рамках програми Європейської комісії IST (англ. Information Societes Technology, інформаційні громадські технології).

Першу версію алгоритму розробили в 1990 році Лай Сюецзя (Xuejia Lai) і Джеймс Мессі (James Massey) зі Швейцарського інституту ETH Zürich (за контрактом з Hasler Foundation, яка пізніше влилася в Ascom-Tech AG) як заміна DES (англ. Data Encryption Standard, стандарт шифрування даних) і назвали її PES (англ. Proposed Encryption Standard, запропонований стандарт шифрування). Потім, після публікації робіт Біхамом і Шаміра по диференціальному криптоанализу PES, алгоритм був поліпшений з метою посилення криптостійкості і названий IPES (англ. Improved Proposed Encryption Standard, покращений запропонований стандарт шифрування). Через рік його перейменували в IDEA (англ. International Data Encryption Algorythm).

Так як IDEA використовує 128-бітний ключ і 64-бітний розмір блоку, відкритий текст розбивається на блоки по 64 біт. Якщо таке розбиття неможливо, останній блок доповнюється різними способами певною послідовністю біт. Для уникнення витоку інформації про кожному окремому блоці використовуються різні режими шифрування. Кожен вихідний незашифрований 64 – біт ний блок ділиться на чотири підблока по 16 біт кожен, так як всі алгебраїчні операції, що використовуються в процесі шифрування, відбуваються над 16-бітними числами. Для шифрування і розшифрування IDEA використовує один і той же алгоритм.

Позначення операцій:

- $\boxplus$ Додавання за модулем 2^{16}.
- $\odot$ Множення за модулем $2^{16}+1$.
- $\oplus$ Побітова виключна диз'юнкція.

Фундаментальним нововведенням в алгоритмі є використання операцій з різних алгебраїчних груп, а саме:

Додавання за модулем 2^{16}.

Множення за модулем $2^{16}+1$.

КОПЗ – 2023

Побітова виключна диз'юнкція (XOR).

Ці три операції несумісні в тому сенсі, що ніякі дві з них не задовольняють дистрибутивному закону, тобто:

$$a \odot (b \oplus c) \neq (a \odot b) \oplus (a \odot c).$$

Застосування цих трьох операцій ускладнює криптоаналіз IDEA в порівнянні з DES, який базується виключно на операції виключає АБО, а також дозволяє відмовитися від використання S-блоків і таблиць заміни. IDEA є модифікацією мережі Фейстеля.

Генерація ключів

З 128-бітного ключа для кожного з восьми раундів шифрування генерується по шість 16-бітних подключів, а для вихідного перетворення генерується чотири 16-бітних підключа. Всього буде потрібно 52 = 8 x 6 + 4 різних подключів по 16 біт кожен. Процес генерації п'ятдесяти двох 16-бітних ключів полягає в наступному:

Насамперед, 128-бітний ключ розбивається на вісім 16-бітних блоків. Це будуть перші вісім подключів по 16 біт кожен – $(K_1^{(1)}K_2^{(1)}K_3^{(1)}K_4^{(1)}K_5^{(1)}K_6^{(1)}K_1^{(2)}K_2^{(2)})$

Потім цей 128-бітний ключ циклічно зсувається вліво на 25 позицій, після чого новий 128-бітний блок знову розбивається на вісім 16-бітних блоків. Це вже наступні вісім подключів по 16 біт кожен – $(K_3^{(2)}K_4^{(2)}K_5^{(2)}K_6^{(2)}K_1^{(3)}K_2^{(3)}K_3^{(3)}K_4^{(3)})$

Процедура циклічного зсуву і розбивки на блоки триває до тих пір, поки не будуть згенеровані всі 52 16-бітних підключа.

Шифрування

Структура алгоритму IDEA показана на рисунку 4.3.

КОПЗ – 2023

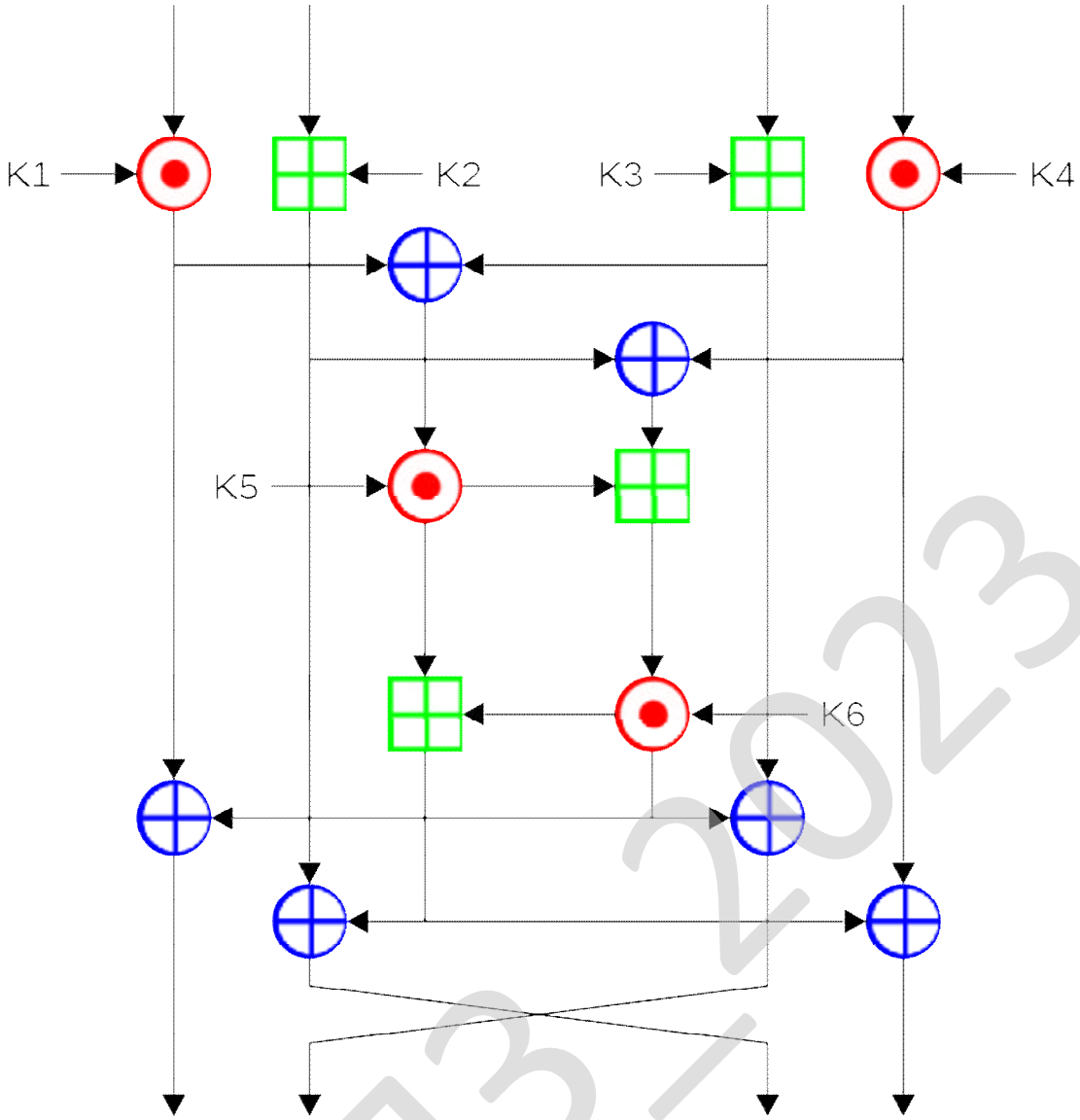

Рисунок 4.3 – Структура алгоритму IDEA

Процес шифрування складається з восьми однакових раундів шифрування і одного вихідного перетворення. Вихідний незашифрований текст ділиться на блоки по 64 біта. Кожен такий блок ділиться на чотири підблока по 16 біт кожен. На рисунку ці підблоки позначені D_1, D_2, D_3, D_4. У кожному раунді використовуються свої підключі згідно з таблицею підключів. Над 16-бітними підключами і підблока незашифрованого тексту проводяться наступні операції:

– Множення за модулем $2^{16}+1 = 65537$, причому замість нуля використовується 2^{16}.

КПЗ 2023

– Додавання за модулем 2^{16}.

– Побітове виключне АБО.

В кінці кожного раунду шифрування є чотири 16-бітних підблоки, які потім використовуються як вхідні підблоки для наступного раунду шифрування. Вихідна перетворення являє собою скорочений раунд, а саме, чотири 16-бітних підблоки на виході восьмого раунду і чотири відповідних підключа піддаються операціям:

– Множення за модулем $2^{16}+1$.

– Додавання за модулем 2^{16}.

Після виконання вихідного перетворення конкатенація підблоків D_1', D_2', D_3' і D_4' являє собою зашифрований текст. Потім береться наступний 64-бітний блок незашифрованого тексту і алгоритм шифрування повторюється. Так продовжується до тих пір, поки не зашифрують всі 64-бітові блоки вихідного тексту.

КбПЗ_2023

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні розділи:

– Меню яке складається з пунктів: Файл; Налаштування; Станції Інтернет-радіо; Довідка.

– Вкладки форм які розділяються на: Інтернет станції; Інструкція користувача; Довідка автора.

– Функції обрання серверу та станції Інтернет-радіо.

– Функціональних кнопок: Сканувати; Налаштувати; Дані авторизації; Наступна Інтернет-станція; Оновлення станцій; Авторська довідка.

Рисунок 5.1 – Головне вікно ПЗ

КБЗІ 2023

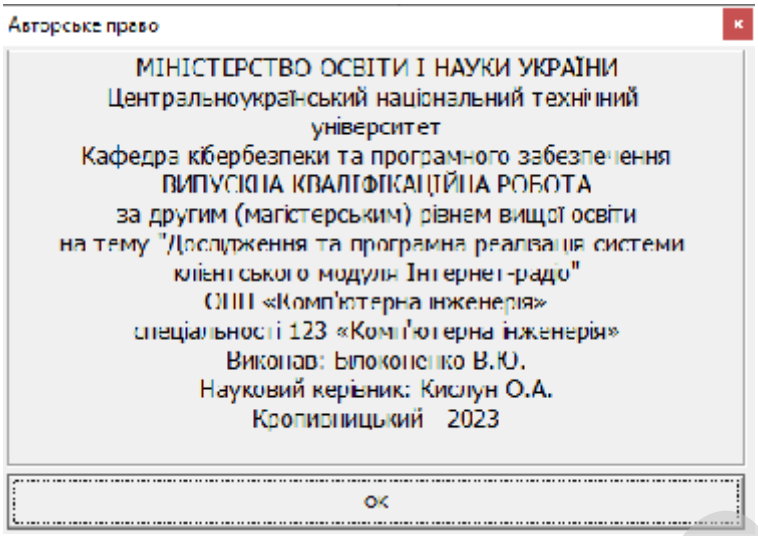

Рисунок 5.2 – Авторське право

Обрано умови розповсюдження – Shareware. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми. В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (зареєструватися), заплативши авторові певну суму. В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

КОПІЯ 2023

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи клієнтського модуля Інтернет-радіо.

Метою розробки є дослідження та програмна реалізація системи клієнтського модуля Інтернет-радіо.

Об'єктом дослідження є процес клієнтського модуля Інтернет-радіо.

Предметом дослідження є методи клієнтського модуля Інтернет-радіо.

Методи дослідження базуються на методах теорії телетрафіку, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод клієнтського модуля Інтернет-радіо.

– Розроблено вітчизняний продукт клієнтського модуля Інтернет-радіо, який має більш широкі можливості, на відміну від існуючих аналогів.

КБПЗ – 2023

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 48 днів (два місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи клієнтського модуля Інтернет-радіо.

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	17
3. Запланований термін розробки, днів	Fpq	48 (2 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

КОПІЯ 2023

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	7
8. Кількість форм вихідної інформації.	–	8
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	2
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	1
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)	–	3
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	1
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	1
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)	–	1
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	3
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	3
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

КОПЗ – 2023

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	1
29. Досвід роботи з програмними інструментами розробки (1-6)	–	2
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	3
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	17000
33. Норматив додаткової зарплати, % :	$Н_д$	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	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 є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де: A – коефіцієнт Боема, A = 2,45;

КПЗ – 2023

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

$$B = 1{,}01+0{,}001(3{,}24+3{,}64+3{,}38+4{,}94+2{,}73) = 1{,}028.$$

$$T_{ном} = 2{,}45 \cdot 2{,}7^{1{,}028} = 6{,}8 \text{ люд-міс.}$$

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

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

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

$$T_{уточн} = 6{,}8 \cdot (0{,}88 \cdot 1 \cdot 0{,}88 \cdot 0{,}91 \cdot 0{,}89 \cdot 1 \cdot 1 \cdot 0{,}87 \cdot 1{,}22 \cdot 1 \cdot 1{,}1 \cdot 1 \cdot$$
$$\cdot 1{,}12 \cdot 1{,}22 \cdot 1{,}12 \cdot 1 \cdot 1{,}1) = 8{,}38 \text{ люд-міс.}$$

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). Для мови програмування Delphi – 2,66;

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

$$T_{РП} = 0{,}3 \cdot 2{,}66 \cdot 8{,}38^{0{,}33+0{,}2(1{,}028-1{,}01)} \cdot 100 = 197 \text{ люд/день.}$$

КПЗ-2023

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

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	15	Д7
Робочий проект	147	Ф 7.1-7.4
Впровадження	17	Д13
Всього	198	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{198 \cdot 1}{48\text{-}5} = 4,6 \text{ ставки.}$$

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.

КПЗ_2023

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 облад-нання	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	3	360	6
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор– маршрутизатор	30	3	90	1,5
Кабельні господарства ЛВС на 1 м. п.	2,5	450	1125	18,75
Копіювальний апарат	140	2	280	4,67
Усього за рік:			$З_ч$	60,25

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$Ф_{др}^{с} = \frac{З_ч \cdot n_{тіс}}{1,2}, \quad (7.6)$$

$$Ф_{др}^{с} = \frac{60,25 \cdot 2}{1,2} = 100 \text{ год.}$$

КПІ 2023

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{Ф^{с}_{др}}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$Ч_{ел}$ = 100/(48·8) = 0,25 ставки.

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків. Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2022, серверу доступу ADSL (ОС Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпровідного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

КБПВ – 2023

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	

КПЗ – 2023

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	13348	26696
Продакт-менеджер	0,25	13138	6569
Інженер-програміст	4,6	15000	138000
Інженер-електронщик	0,25	10000	5000
Інженер-системотехнік	0,25	10000	5000
Адміністратор мережі	0,5	10000	10000
Системний програміст	0,25	10000	5000
Дизайнер WEB	0,25	10000	5000
Інженер-верстальник	0,25	10000	5000
Бухгалтер-економіст	0,25	12500	6250
Всього за період розробки	$R_{сп}$ = 7,85	-	$Ф_{роб}$ = 212515

Розрахуємо середньоденну зарплату одного виконавця:

$$З_{сд} = \frac{Ф_{роб}}{R_{сп} F_{рд}}, \quad (7.8)$$

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

$$З_{сд} = \frac{212515}{7,85 \cdot 48} = 564 \; грн.$$

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:

КОПІЯ 2023

$$Б_{уд} = R^1_{сп} S_у Ц_{пл}, \quad (7.9)$$

де: $R^1_{сп}$ – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

$S_у$ – питома площа на одне робоче місце, $м^2$;

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

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пекторaль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 400…1600 $у.о./м^2$. Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./м^2. На кожне робоче місце у середньому потрібно 8 $м^2$. З урахуванням цього:

$$Б_{уд} = 9 \cdot 8 \cdot 20000 = 1440000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 144000 грн.

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

$$І_{нв} = R^1_{сп} \cdot Ц_м, \quad (7.10)$$

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

$$І_{нв} = 9 \cdot 20500 = 184500 \text{ грн.}$$

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 фірми supercomp.kiev.ua за 29.10.23 – джерело http: //supercomp.kiev.ua.

КБПЗ – 2023

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп’ютер		12721
Системний блок		7721
Процесор	AMD Ryzen 5 4500 (100-000000644) AM4, 6 ядер, 12 потоків, 3.6 GHz, 4.1 GHz, TDP – 65 Вт, 7nm, L1: 384KB, L2: 3MB, L3: 8MB	-
Системна плата	ASUS PRIME A320M-K AM4, DDR4, 3200 MHz, LAN – 1 Гбіт/с, D-Sub (VGA), HDMI, 1 x M.2, 4 x Sata 6.0 Gb/s, Micro-ATX	-
Відеокарта	Radeon R7 350 2Gb Afox (AFR7350-2048D5H4-V3) PCI-Express 3.0, 2 ГБ, GDDR5, 128 Bit	-
Жорсткий диск	SSD 2.5" 256GB Mibrand (MI2.5SSD/CA256GBST) 256 GB, 3D TLC NAND, 2.5", SATA III (6Gb/s)	-
Оперативна пам’ять	DDR4 8GB 3200 MHz Dato (DT8G4DLDND32) DDR4, 8 ГБ, В наборі – 1 шт	-
DVD-привод	-	-
Корпус	Vinga CS105B-450W, Miditower, ATX, Micro – ATX, Mini – ITX, PSU – 450 Вт	
Кардридер внутрішній	Transcend TS-RDF8K USB 3.0	
інше	Клавіатура, мишка	Подаруно

КОПІЯ 2023

Продовження таблиці 7.5

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	Монітор BenQ GL2450HM Black	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37	2800
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійног живлення	Powercom BNT-600AP USB	1400

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

Для визначення необхідної кількості капітальних вкладень складем таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість ,шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Персональні комп’ютери	9	12721	11448,9	125937,9
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	147569,4

КОПІЯ 2023

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1440000	-	-
2. Передавальні пристрої	144000	-	-
Всього по групі	1584000	5	79200
Група 4			
3. Обчислювальна техніка	147570	-	-
Всього по групі	147570	40	59028
Нематеріальні активи			
4. Нематеріальні активи	17000	10	1700
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	121875	20	24375
7. Господарський інвентар	180500	25	45125
Всього по групі	311406	-	71757,75
Разом	$К_р$ = 2059976		$А_р$ = 211685,75

Примітка: вартість автомобіля взята по даним електронного ресурсу http://www.avtopoisk.ua, що враховуючи курс 37 складає 121875 грн.

КПІ 2023

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$З_о = \frac{З_{сд} \cdot Т_{пз}}{N_е}, \quad (7.11)$$

де: $N_е$ – кількість екземплярів програм, шт.

$$З_о = 564 \cdot 198/17 = 6579 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$З_д = З_о \cdot Н_q \cdot 0{,}01, \quad (7.12)$$

де: $Н_q$ – норматив додаткової зарплати, %.

$$З_д = 6579 \cdot 10 \cdot 0{,}01 = 658 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $Н_с = 22\%$ від суми основної та додаткової зарплати:

$$С_{оц} = 0{,}01 \cdot Н_с (З_о + З_д), \quad (7.13)$$

де: $Н_с$ – відрахування на соціальні потреби, %.

$$С_{оц} = 0{,}01 \cdot 22(6579+658) = 2678 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $Н_г = 15\%$ від основної зарплати:

$$Г_{осп} = З_о \cdot Н_г \cdot 0{,}01, \quad (7.14)$$

де: $Н_г$ – загальногосподарські витрати, %.

$$Г_{осп} = 6579 \cdot 15 \cdot 0{,}01 = 987 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$З_М = (З_{М1} + З_{М2} + З_{М3})/N_е, \quad (7.15)$$

де: $З_{М1}$ – вартість паперу, грн.; $З_{М2}$ – вартість запам'ятовуючих пристроїв, грн.; $З_{М3}$ – вартість фарби, картриджів, тонеру, грн.; $N_е$ – кількість екземплярів програм, шт.

КПЗ-2023

Згідно виданих викладачем норм приймаємо 0,5 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_п = 70$ грн., визначаємо вартість паперу за період розробки $N_м = 2$ міс:

$$З_{М1} = Ц_п \cdot N_м. \quad (7.16)$$

$$З_{М1} = 0{,}5 \cdot 70 \cdot 2 = 70 \text{ грн.}$$

Згідно норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків за потребою та одного DVD диска для збереження резервної копії програми:

$$З_{М2} = \sum Ц_д\ , \quad (7.17)$$

де: $Ц_д$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 22 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 36 грн./шт.

$$З_{М2} = 36 \cdot 1 = 36 \text{ грн.}$$

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

$$З_{М3} = \sum Ц_{з.}, \quad (7.18)$$

де: $Ц_з$ – 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.

$$З_{М3} = 500+508+1155 = 2163 \text{ грн.}$$

$$З_М = (70 + 36 + 2163)/17 = 133 \text{ грн.}$$

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

$$О_п = З_о \cdot Н_п \cdot 0{,}01, \quad (7.19)$$

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

$$О_п = 6579 \cdot 15 \cdot 0{,}01 = 987 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_е = 17$ прим.):

КОПІЯ 2023

$$A_м = \frac{A_р \cdot N_{міс}}{N_е \cdot 12}, \quad (7.20)$$

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

$$A_м = 211686 \cdot 2/(17 \cdot 12) = 2075 \text{ грн.}$$

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	$З_о$	6579
2. Додаткова зарплата виконавців	$З_д$	658
3. Відрахування на соціальні потреби	$С_{оц}$	2678
4. Загальногосподарські витрати	$Г_{осп}$	987
5. Витрати на матеріали	$З_М$	133
6. Освоєння нових операційних систем, мов програмування	$О_п$	987
7. Амортизація основних фондів	$А_м$	2075
8. Повна собівартість програмного забезпечення	$С_п$	14097
9. Плановий прибуток	$П_р$	7048,5
10.Ціна підприємства $Ц_п = С_п + П_р$	$Ц_п$	21145,5
11.Податок на додану вартість $ПДВ = 0.01 \cdot Н_{дв} \cdot Ц_п$	$ПДВ$	4229,1
12.Відпускна ціна програмної продукції $Ц = Ц_п + ПДВ$	$Ц$	25374,6

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

КПІ – 2023

$$C_п = З_о + З_д + С_{оц} + Г_{осп} + З_м + О_п + А_м. \quad (7.21)$$

$$C_п = 6579+658+2678+987+133+987+2075 = 14097 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$П_р = 0{,}01 \cdot P_п \cdot C_п, \quad (7.22)$$

де: $P_п$ – рівень рентабельності, %.

$$П_р = 0{,}01 \cdot 50 \cdot 14097 = 7048{,}5 \text{ грн.}$$

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 пусконалагоджувальних робіт, а також витрати на випробування у виробничих умовах.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	25375
Всього капітальних витрат	–	25375

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.

КПІ 2023

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування	$З_р$	35430	11810
2. Витрати на електроенергію	$З_{ел}$	205	137
3. Витрати на амортизацію	$З_{ам}$	0	6344
Всього витрат за рік	$І$	35635	18291

Витрати на обслуговування системи:

$$З_р = Т_р \cdot З_г \cdot (1 + 0{,}01 \cdot Н_q) \cdot (1 + 0{,}01 \cdot Н_c), \quad (7.23)$$

де: $Т_р$ – кількість годин обслуговування за рік, *год.*;

$З_г$ – заробітна плата обслуговуючого персоналу, *грн / год*.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 80 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$З_{р\,баз} = 240 \cdot 110 \cdot 1{,}1 \cdot 1{,}22 = 35430 \text{ грн},$$

до:

$$З_{р\,нов} = 80 \cdot 110 \cdot 1{,}1 \cdot 1{,}22 = 11810 \text{ грн.}$$

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

$$З_{ел} = П_{ел} \cdot Т_р \cdot Ц_{ел}. \quad (7.24)$$

$$З_{ел\,баз} = 0{,}45 \cdot 120 \cdot 3{,}8 = 205 \text{ грн.}$$

$$З_{ел\,нов} = 0{,}45 \cdot 80 \cdot 3{,}8 = 137 \text{ грн.}$$

КПІ – 2023

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	–	25375	–	6343,75
Всього відрахувань	-	–	25375	–	6343,75

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:

$$E_в = (Ц_п - C_п) \cdot N_е - \sum_{i=1}^{m} E_{p_m} \cdot K_{p_m}, \quad (7.24)$$

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

$$Ев = (21145,5 - 14097) \cdot 17 - (0,05 \cdot 1584000 + 0,4 \cdot 147570 + \\ + 0,2 \cdot 121875 + 0,25 \cdot 189531 + 0,1 \cdot 17000) \cdot 2/12 = 84544 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_в = \frac{K_p^*}{(Ц_п - C_п) \cdot N_е}, \quad (7.25)$$

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.

КБПЗ-2023

$$T_в = \frac{2059976}{(21145,5 - 14097) \cdot 17 \cdot 12 / 2} = 2,9 \text{ років}.$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	17
2. Повна собівартість розробленої програми	Грн.	14097
3. Ціна розробленої програми	Грн.	21145,5
4. Плановий прибуток від реалізації розробленої програми	Грн.	7048,5
5. Рентабельність програмної продукції	%	50
6. Об’єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2059976
7. Загальний прибуток від реалізації програмної продукції	Грн.	119824,5
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	84544
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	2,9
10.Об’єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	25375
11.Величина економічного ефекту у користувача програмної продукції	Грн.	11000
12.Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	1,5

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

КОПІЯ 2023

$$E_{сп} = (I_б - I_н) - E_н(K_н - K_б), \quad (7.27)$$

де: $I_б$, $I_н$ – величина експлуатаційних витрат за базовим и новим варіантом відповідно; $K_б$, $K_н$ – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{сп} = (35635\text{-}18291)\text{-}0{,}25 \cdot 25375 = 11000 \text{ грн.}$$

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

$$T_{сп} = \frac{K_н - K_б}{I_б - I_н}, \quad (7.28)$$

$$T_{сп} = \frac{25375}{35635 - 18291} = 1{,}5 \text{ року}.$$

7.9 Висновки

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

КбПЗ_2023

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Законом України “Про охорону праці” [1] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207 [5], який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби польців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням [2]. До шкідливих факторів, які впливають на робитників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2], та «Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

КПІ - 2023

Умови праці програміста вулючають наступні фактори:

– параметри повітряного середовища в приміщенні;

– вентиляція приміщення;

– освітлення приміщення;

– параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення впливу комп'ютера на організм програміста визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста,

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Це важливе питання, адже робота з комп'ютером може впливати на здоров'я людини. За ДСанПіН 3.3.2.007-98 шкідливі і небезпечні фактори при роботі з комп'ютером можуть бути такі:

- Електромагнітне випромінювання – це випромінювання, яке створюється комп'ютером і його периферійними пристроями, такими як монітор, принтер, сканер тощо. Це випромінювання може викликати головний біль, запаморчення, розлади сну, зниження імунітету та інші негативні ефекти.
- Електростатичне поле – це поле, яке утворюється внаслідок накопичення електричних зарядів на поверхні комп'ютера і його частин. Це поле може спричиняти сухість шкіри, свербіж, подразнення очей, алергічні реакції та інші проблеми.
- Нервово-емоційне напруження – це напруження, яке виникає внаслідок трвлої концентрації уваги, високої вимогливості до результату роботи, нестабільності програмного забезпечення, конфліктних ситуацій тощо. Це напруження може призводити до стресу, депресії, роздратування, погіршення пам'яті та інших порушень.
- Навантаження на органи зору – це навантаження, яке виникає внаслідок тривалого перебування перед монітором, низької якості зображення, недостнго

КПЗ-2023

освітлення приміщення, поганої ергономіки робочого місця тощо. Це навантаження може спричиняти зниження гостроти зору, появу блимавок і кругів перед очима, синдром сухого ока та інші захворювання.

– Дрібні стереостатичні рухи кінцівок – це рухи, які пов'язані з керуванням клавіатурою і мишею. Ці рухи можуть призводити до перевантаження і запалення суглобів і сухожиль рук і пальців, а також до тендовагінітів і синдрому зап'ясткового каналу.

Таким чином, робота з комп'ютером не така безпечна, як може здатися на перший погляд. Тому дуже важливо дотримуватися правил охорони праці і гігієни при роботі з комп'ютером.

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	5,4
Довжина	6
Висота	3

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

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	$м^2$	не менше 6.0	6,5
Обсяг, V	$м^3$	не менше 20.0	19,4

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин) [2].

КОПЗ – 2023

У зазначеному приміщенні працє 5 осіб. За даними, які наведено у табл. 8.1 та табл. 8.2, можна зробити висновок, що площа приміщення у розрахунку на одно робоче місце програміста 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», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2], а об'єм – НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

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 Головного державного санітарного лікаря України [6], робота, яка виконується в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря у 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.

КПІ – 2023

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	22-23,5	40-60	0,1
Тепла	23-25	50-70	0,1	23-25	52-70	0,11

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 HP Laser 107a, 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.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5 – 28 – 2006 р 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. Крім того все поле зору повинне бути

КПЗ-2023

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:

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

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значеннь;

Таким чином можна припустити, що основною причиною можнивого 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’язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору).

КПІ – 2023

8.5 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток *63·63·6 мм*., (згідно з ДСТУ 2251-93 «Кутки сталеві гарячекатані рівнополичні. Сортамент») довжиною *L=2 м.,* та горизонтальний електрод – металева полоса з перетином *60·5 мм*. Напруга – *220/380 В*. Розрахункова схема розташування заземлюючих електродів – по контуру (прямокутником).

Розрахунок проведем за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару грунта – чорнозем, нижнього шару грунта – глина (питомий опір $\rho_2 = 40$ *Ом·м)*. Умовна товщина верхнього шару грунта: *H=0,5 м*. Відстань між вертикальними заземлювачами (електродами) *A=2 м*. Глибина закладення горизонтального контура заземлення *t=0,6 м*. Опір заземлювача, який нормується: $R_{3H} = 4$ *Ом*. Необхідно визначити необхідну кількість вертикальних заземлювачів та длину полоси (горизонтального заземлювача).

Розрахунок

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,6+2/2=1,6 \text{ м.}$$

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

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

де $\psi = 1,36$ – табличне значення коефіцієнта сезонності для відповідной климатичной зони у багатошаровому грунті [7];

$\rho_2 = 40$ *Ом·м.* – табличне значення питомого опору нижнього шару грунта (глина) [7].

Еквівалентний діаметр ветикальнього електрода (кутка) [7]:

$$Dв=0,95 \cdot K = 0,95 \cdot 63 = 59,85 \text{ мм.} = 0,0598 \text{ м.}$$

де $K = 63$ *мм.* – розмір метелевого кутка (задан).

КПІ – 2023

Відношення $A/L=2/2=1$

Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [7]:

$$R_O=0,366(\rho/L)[lg(2L/Dв)+(1/2)lg((4T+L)/(4T-L))]=$$

$$=0,366(54,5/2)[(lg(2\cdot2/0,0598)+(1/2\ )lg((4\cdot1,6+2)/(4\cdot1,6-2))]=$$

$$=19,6\ Ом.$$

Визначаємо коефіцієнт екранування ветикальньних електродів $Кев=0,62$ при орієтовній кількості ветикальньних електродів, яке дорівнює 5 [7].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без вхарування горизонтального заземлювача), при $R_{ЗН}=4\ Ом$:

$$N=R_O/(Кев\ R_{ЗН})=19,6/(0,62\cdot4)=7,8\approx8\ шт.$$

Визначаємо длину з'єднуючої полоси:

$$L_П=1,05\cdot A\cdot N=1,05\cdot2\cdot8=16,6\approx16\ м.$$

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору грунта $K_П$ [7]:

$$R_П=0,366(\rho\cdot K_П/L_П\ )lg(2\cdot L_П^{\,2}/(B\cdot t))=$$

$$=0,366(40\cdot5/16)\cdot lg((2\cdot16^{\,2}\ )/(0,06\cdot0,6\ ))=20,2\ Ом.$$

де $K_П=5$ – табличне значення кліматичного коефіцієнта питомого опору грунта для відповідной климатичной зони для з'єднуючої полоси [7]:

$B=60\ мм.=0,06\ м.$- ширина з'єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [7]:

$$R=(R_O\cdot R_П)/(R_O\cdot\eta_П+N\cdot R_П\cdot Кев)=$$

$$=(19,6\cdot20,2)/(19,6\cdot0,6+8\cdot20,2\cdot0,62)=3,53\ Ом.$$

де $\eta_П=0,6$ – табличне значення коефіцієнта екранування з'єднуючої полоси [7].

Умова $R\le R_{ЗН}$ виконується ($3,53\le4$).

Остаточно отримали: кількість вертикальних електродів дорівнює *8* при $R=3,53\ Ом.$

КПІ 2023

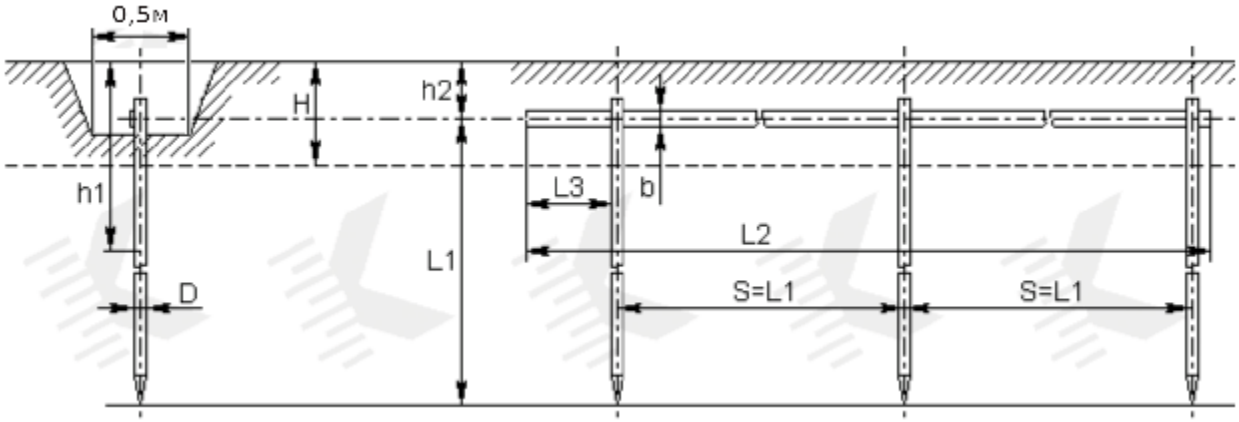

Рисунок 8.1 – Схема штучного заземлення

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.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів з питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

Список використаних джерел інформації

1. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-XII. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/2694-12 (дата звернення 19.10.23).

2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим

КОПІЯ 2023

доступу до ресурсу: https://zakon.rada.gov.ua/rada/show/v0007282-98 (дата звернення 19.10.23).

3. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: https://goo.su/9AkQ (дата звернення 19.10.23).

4. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/z0508 (дата звернення 19.10.23).

5. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/rada/show/va042282-99 (дата звернення 19.09.).

6. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: http://dspace.kntu.kr.ua/jspui/handle/123456789/12161 (дата звернення 19.09.).

7. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу: http://dspace.kntu.kr.ua/jspui/handle/123456789/12240 (дата звернення 19.09.).

8. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ ІВМ сумісного типу / Кіровоград. ін-т с.-г. машинобуд. ; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград: КІСМ, 1997. – 20 с. – Режим доступу: http://dspace.kntu.kr.ua/jspui/handle/123456789/4358 (дата звернення 19.09.).

КОПІЯ 2023

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи клієнтського модуля Інтернет-радіо.

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 клієнтського модуля Інтернет-радіо.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем клієнтського модуля Інтернет-радіо.

– Досліджена система клієнтського модуля Інтернет-радіо.

– На основі отриманих результатів досліджень створена програмна реалізація системи клієнтського модуля Інтернет-радіо.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання клієнтського модуля Інтернет-радіо.

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.

КПЗ-2023

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

Програма реалізована на мові високого рівня RAD Studio Delphi 10. 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 IDEA.

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 11000 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 1,5 роки.

КПЗ - 2023

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Білоконенко В.Ю. Дослідження та програмна реалізація системи клієнтського модуля Інтернет-радіо // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.

2. Cisco Voice Over IP. Student Guide – – Cisco Systems Inc, 2003

3. Cisco IP Telephony. Student Guide. Version 3.3. – – Cisco Systems Inc, 2002

4. Robert Padjen – – Cisco AVVID and IP Telephony. Design & Implementation – – Syngress Publishing Inc, 2001

5. Paul J. Fong – – Configuring Cisco Voice Over IP. Second Edition – – Syngress Publishing Inc, 2001

6. ITU-T Recommendation G.723.1. Dual Rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kit / sec. – 1996.

7. ITU-T Recommendation G.729. Speech codec for multimedia telecommunications transmitting at 8 / 13 kbit / s. – 1996.

8. ITU-T Recommendation H.225.0. Call signaling protocols and media stream packetization for packet-based multimedia communication systems. – Geneva, 1998.

9. ITU-T Recommendation H.245. Control protocol for multimedia communication. – Geneva, 1998

10. ITU-T Recommendation H.248. Gateway control protocol. – Geneva, 2000.

11. ITU-T Recommendation H.320. Narrow-band Visual Telephone Systems and Terminal Equipment. – 1996.

12. ITU-T Recommendation H.321. Adaptation of H.320 Visual Telephone Terminals to B-ISDN Environments. – 1996.

13. ITU-T Recommendation H.322. Visual Telephone Systems and Terminal Equipment for Local Area Networks which Provide a Guaranteed Quality of Service. –

КБПЗ-2023

1996.

14. ITU-T Recommendation H.323. Packet based multimedia communication systems. – Geneva, 1998.

15. ITU-T Recommendation H.324. Terminal for Low Bit Rate Multimedia Communications. – 1996.

16. ITU-T Recommendation Q.931. ISDN User-Network Interface Layer 3 Specification for Basic Call Control. – 1993.

17. RFC 2705. Media Gateway Control Protocol (MGCP) Version 1.0. M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett. October 1999.

18. RFC 2865. Remote Authentication Dial In User Service (RADIUS). C.Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.

19. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.

20. 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.

21. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022,

22. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.

23. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w

24. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE

КОПІЯ 2023

International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

25. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

26. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

27. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

28. 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.

29. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

30. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

31. 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

КОПЗ-2023

Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

32. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko 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.

33. 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.

34. 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.

35. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

36. Zhurakovskyi, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

37. Smirnov O. Kuznetsov A., Zaicenko 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.

38. 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.

КОПЗ – 2023

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. Smirnov A.A., Kuzne tsov 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.

45. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

КОПЗ 2023

46. Т.В. Смірнова, О.М. Дрєєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макрофотографій». IV міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

47. 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.

48. 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.

49. Смірнов О.А., Дрєєва Г.М., Дрєєв 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.

50. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» Комп’ютерні науки та кібербезпека. № 4. С. 30-37. 2019.

51. Смірнов О.А., Конопліцька-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.

КБПЗ – 2023

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

КПЗ-2023

					ВКРМ-123.23.0029.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив		Білоконенко В.Ю.			Дослідження та програмна реалізація системи клієнтського модуля Інтернет-радіо	Літ.	Аркуш 1 / Аркушів 6
Перевірив		Кислун О.А.				М	
Н. Контр.		Коваленко А.С.				ЦНТУ КІ-22М-2	
Затв.		Смірнов О.А.					

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи клієнтського модуля Інтернет-радіо.

2 Підстава для розробки

Підставою для розробки служить завдання на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 35-13 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 клієнтського модуля Інтернет-радіо.

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;

КБПЗ_2023

– техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

– аналіз умов праці;

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

– програмну реалізацію системи клієнтського модуля Інтернет-радіо;

– 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.

КОПЗ – 2023

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 RAD Studio Delphi 10.

КбПЗ 2023

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.

КБПЗ – 2023

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 – 104 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 22.12.2023 р.

КОПЗ_2023

Додаток Б

(обов’язковий)

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Кислун О.А.

Дослідження та програмна реалізація системи клієнтського модуля Інтернет-радіо

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 34

Літера: РП

Кропивницький – 2023 року

КБПЗ – 2023

Основна програма – Програма організації зв`язку для інтернет радіо

```
program internet_radio;

uses
  Forms,
  Dialogs,
  Windows,
  Sysutils,
  Messages,
  usimple in 'usimple.pas' {Form1},
  Unit2 in 'Unit2.pas' {Form2},
  Unit3 in 'Unit3.pas' {Form3},
  Unit1 in 'Unit1.pas' {Form0},
  Unit4 in 'Unit4.pas' {Form4};
{$R simple.RES}
var
  i:integer;
{        function Crypt(Text,Key: String; Encode: boolean): String;
        var
          i, KeyLength: integer;
          Sign: ShortInt;
        begin
         KeyLength:=Length(Key);
         if Encode then Sign :=-1 else Sign:=1;
         for i:=1 to Length(Text) do
          Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
          Result:=Text;
        end;}
begin
Application.Initialize;
{
if FileExists('main.dat')=false then
 begin
    MessageDlg('Файл       main.dat       не       знайдено!       завершення
програми',mtInformation,[mbOK],0);
    Application.Terminate;
 end else
 begin
   AssignFile(F, 'main.dat');
   Reset(F);
   Readln(F, S);
   CloseFile(F);
      if Crypt(InputBox('Увага!','Введіть пароль',Y),KEY,false)=S then
      begin
           MessageDlg('Дякую, пароль вірний!', mtInformation,[mbOK],0);}
            Try
                Form2:=TForm2.Create(Application);
                Form2.Show;
                Form2.Update;
                for i:=1 to 10 do
                begin
                  sleep(200);
                  Form2.ProgressBar1.Position:=i*10;
                  Form2.Update;
                end;
               Application.Title := 'Internet_radio';
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm3, Form3);
  Application.CreateForm(TForm0, Form0);
  Application.CreateForm(TForm4, Form4);
  Finally
                Form2.Free;
   End;
     Application.Run;
{      end
      else
```

КОПІЯ 2023

```
      begin
      MessageDlg('Невірний пароль!',mtInformation,[mbOK],0);
      Application.Terminate;
      end;
  end;}
end.
```

КбПЗ_2023

Підпрограма завантаження загрузочного вікна

```
unit U_SPLASH;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, StdCtrls, Gauges;

type
  TForm_SPLASH = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Gauge1: TGauge;
    Timer1: TTimer;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form_SPLASH: TForm_SPLASH;

implementation

{$R *.dfm}

end.
```

КбПЗ_2023

Головне вікно програми

```
unit usimple;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  telefoon, StdCtrls, ComCtrls, ExtCtrls, jpeg, Menus, Buttons;

type
  TForm1 = class(TForm)
    Telefoon1: TTelefoon;
    StatusBar1: TStatusBar;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    Panel1: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    Label4: TLabel;
    Edit1: TEdit;
    Button2: TButton;
    Button1: TButton;
    Panel5: TPanel;
    TabSheet2: TTabSheet;
    Panel6: TPanel;
    Label2: TLabel;
    Panel7: TPanel;
    MainMenu1: TMainMenu;
    Panel8: TPanel;
    ListBox1: TListBox;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    ListBox2: TListBox;
    Panel9: TPanel;
    Label3: TLabel;
    Panel10: TPanel;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    N1: TMenuItem;
    IP1: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    BitBtn5: TBitBtn;
    Memo1: TMemo;
    TabSheet3: TTabSheet;
    Panel11: TPanel;
    Image3: TImage;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    N11: TMenuItem;
    N12: TMenuItem;
    ListBox3: TListBox;
    Label9: TLabel;
    Label10: TLabel;
    N2: TMenuItem;
    PB1: TProgressBar;
    PB2: TProgressBar;
    T1: TTimer;
    Image1: TImage;
    Timer1: TTimer;
```

КОПІЯ 2023

```
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure ListBox1Click(Sender: TObject);
    procedure ListBox2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure N10Click(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);
    procedure N12Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure N3Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure T1Timer(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

uses Unit3, Unit2, Unit4, Unit1;
var
  KEY:string='2%#$T%&(*qwrdas@@@#45';
{$R *.DFM}
procedure TForm1.Button1Click(Sender: TObject);
begin
 Telefoon1.placecall(Edit1.text);
 ListBox3.items.add('D:'+DateToStr(now)+'                T:'+TimeToStr(now)+'
IP:'+Edit1.text);
  T1.Enabled:=true;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
 Telefoon1.calling:=false;
 T1.Enabled:=false;
 PB1.Position:=0;
 PB2.Position:=0;
end;
procedure TForm1.ListBox1Click(Sender: TObject);
begin
 Edit1.text:=ListBox1.Items.Strings[ListBox1.ItemIndex];
end;
procedure TForm1.ListBox2Click(Sender: TObject);
begin
 Label4.Caption:=ListBox2.Items.Strings[ListBox2.ItemIndex];
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
var
  InputString: string;
begin
  InputString:=  InputBox('Додавання  ip  адреси',  'Прошу',  'Введення  IP  не
здійснено');
  if (InputString<>'Введення IP не здійснено') then
  begin
  Listbox1.Items.add(InputString);
  end;
end;
procedure TForm1.BitBtn3Click(Sender: TObject);
var
```

КОПІЯ 2023

```
  InputString: string;
begin
 InputString:=  InputBox('Додавання легенди ip адреси', 'Прошу', 'Введення
легенди не здійснено');
 if (InputString<>'Введення легенди не здійснено') then
 begin
 Listbox2.Items.add(InputString);
 end;
end;
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
 ListBox1.Items.Strings[ListBox1.ItemIndex]:='';
 ListBox1.Items.Delete(ListBox1.ItemIndex);
end;

procedure TForm1.BitBtn4Click(Sender: TObject);
begin
 ListBox2.Items.Strings[ListBox1.ItemIndex]:='';
 ListBox2.Items.Delete(ListBox1.ItemIndex);
end;
procedure TForm1.N10Click(Sender: TObject);
begin
 Application.Terminate;
end;
procedure TForm1.BitBtn5Click(Sender: TObject);
var
 i:integer;
 G:boolean;
begin
 Form3.ListBox1.Clear;
 g:=false;
 for i:=0 to (ListBox1.Items.Count - 1) do
 begin
  if ListBox1.Selected[i] then
    begin
        Form3.ListBox1.Items.add(ListBox1.Items.Strings[i]);
        g:=true;
    end;
    if g then Form3.show;
 end;
end;
procedure TForm1.N12Click(Sender: TObject);
var
  F: TextFile;
  H,S:string;
  OLD:string;
        function Crypt(Text,Key: String; Encode: boolean): String;
        var
         i, KeyLength: integer;
         Sign: ShortInt;
        begin
         KeyLength:=Length(Key);
         if Encode then Sign :=-1 else Sign:=1;
         for i:=1 to Length(Text) do
          Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
          Result:=Text;
        end;
begin
if FileExists('main.dat') then
begin
    AssignFile(F, 'main.dat');
    Reset(F);
    Readln(F, S);
     CloseFile(F);
  if Crypt(InputBox('Увага!','Введіть старий пароль',''),KEY,false)=S then
    begin
        H:=Crypt(InputBox('Увага!','Введіть новий пароль',''),KEY,false);
        if H<>'' then
        begin
```

КОПІЯ 2023

```
          MessageDlg('Дякую пароль змінено',mtInformation,[mbOK],0);
          AssignFile(F, 'main.dat');
          Rewrite(F);
          Writeln(F,H);
          CloseFile(F);
        end else  MessageDlg('Введіть значення!',mtInformation,[mbOK],0);
    end
    else
     begin
        MessageDlg('Файл main.dat не знайдено чи пароль невірний! завершення
програми',mtInformation,[mbOK],0);
        Application.Terminate;
      end;
end;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
randomize;
PB1.Position:=0;
PB2.Position:=0;
T1.Enabled:=false;
Timer1.Enabled:=true;
if FileExists('MainLegend.dat')=false then
 begin
    MessageDlg('Файл MainLegend.dat не знайдено!',mtInformation,[mbOK],0);
 end else
 begin
    ListBox2.Items.LoadFromFile('MainLegend.dat');
  end;
if FileExists('MainIP.dat')=false then
 begin
    MessageDlg('Файл MainIP.dat не знайдено!',mtInformation,[mbOK],0);
 end else
 begin
    ListBox1.Items.LoadFromFile('MainIP.dat');
  end;
if  FileExists('MainHISTORY.dat')=false  then  MessageDlg('Файл  MainHISTORY.dat
не знайдено!',mtInformation,[mbOK],0)
else ListBox3.Items.LoadFromFile('MainHISTORY.dat');
if  FileExists('mainHelp.dat')=false  then  MessageDlg('Файл  mainHelp.dat  не
знайдено!',mtInformation,[mbOK],0)
else Memo1.Lines.LoadFromFile('mainHelp.dat');
end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
if FileExists('MainLegend.dat')=false then MessageDlg('Файл MainLegend.dat не
знайдено!',mtInformation,[mbOK],0)
else ListBox2.Items.SaveToFile('MainLegend.dat');
if  FileExists('MainIP.dat')=false  then  MessageDlg('Файл  MainIP.dat  не
знайдено!',mtInformation,[mbOK],0)
else ListBox1.Items.SaveToFile('MainIP.dat');
if  FileExists('MainHISTORY.dat')=false  then  MessageDlg('Файл  MainHISTORY.dat
не знайдено!',mtInformation,[mbOK],0)
else ListBox3.Items.SaveToFile('MainHISTORY.dat');
end;
procedure TForm1.N3Click(Sender: TObject);
begin
TabSheet2.Visible:=true;
end;
procedure TForm1.N6Click(Sender: TObject);
begin
 Telefoon1.calling:=false
end;
procedure TForm1.N2Click(Sender: TObject);
begin
Form0.show;
end;
procedure TForm1.FormShow(Sender: TObject);
begin
```

КПІ-2023

```
PB1.Position:=0;
PB2.Position:=0;
end;
procedure TForm1.T1Timer(Sender: TObject);
var
i1,i2:integer;
begin
PB1.Position:=random(100);
PB2.Position:=PB1.Position;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
Form4.show;
Form1.hide;
Timer1.Enabled:=false;
end;
end.
```

КбПЗ_2023

Підпрограма налагодження звукової карти AudioSettings

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComCtrls, StdCtrls, AMixer, MMSystem;
type
  TForm0 = class(TForm)
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    TrackBar: TTrackBar;
    CheckBox: TCheckBox;
    Label1: TLabel;
    Label2: TLabel;
    Mixer: TAudioMixer;
    Label3: TLabel;
    Label4: TLabel;
    ComboBox3: TComboBox;
    LabelStereo: TLabel;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure ComboBox2Change(Sender: TObject);
    procedure MixerControlChange(Sender: TObject; MixerH, ID: Integer);
    procedure TrackBarChange(Sender: TObject);
    procedure CheckBoxClick(Sender: TObject);
    procedure ComboBox3Change(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
    Setting:Boolean;
  public
    { Public declarations }
  end;
var
  Form0: TForm0;
implementation
uses usimple;
{$R *.DFM}
procedure TForm0.FormCreate(Sender: TObject);
var A:Integer;
begin
  For A := 0 to Mixer.MixerCount - 1 do
    ComboBox3.Items.Add ('Mixer '+IntToStr(A));
  If (ComboBox3.Items.Count > 0) then
    ComboBox3.ItemIndex := 0;
  ComboBox3Change (Sender);
end;
procedure TForm0.ComboBox1Change(Sender: TObject);
var A:Integer;
begin
  ComboBox2.Items.Clear;                                    ComboBox2.Items.Add
(Mixer.Destinations[ComboBox1.ItemIndex].Data.szName);
  For A:=0 to Mixer.Destinations[ComboBox1.ItemIndex].Connections.Count-1 do
ComboBox2.Items.Add(Mixer.Destinations[ComboBox1.ItemIndex].Connections[A].Dat
a.szName);
  If ComboBox2.Items.Count>0 then
  begin
    ComboBox2.ItemIndex:=0;     ComboBox2Change (Self);
  end;
end;
procedure TForm0.ComboBox2Change(Sender: TObject);
var L,R,M:Integer;
    VD,MD:Boolean;
    Stereo:Boolean;
    IsSelect:Boolean;
begin
```

КСПЗ 2023

```
  Mixer.GetVolume                    (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,L,R,M,Stereo,VD,MD,IsSelect);
  Setting:=True;
  TrackBar.Visible:=not VD;
  Label1.Visible:=not VD;
  Label3.Visible:=VD;
  If TrackBar.Visible then
    TrackBar.Position:=L;
  CheckBox.Visible:=not MD;
  Label2.Visible:=not MD;
  Label4.Visible:=MD;
  If CheckBox.Visible then
    CheckBox.Checked:=M<>0;
  If (Stereo) then
    LabelStereo.Caption := '- stereo -'
  else
    LabelStereo.Caption := '- mono -';
  Setting:=False;
end;
procedure TForm0.MixerControlChange(Sender: TObject; MixerH, ID: Integer);
begin
  ComboBox2Change (Self);
end;
procedure TForm0.TrackBarChange(Sender: TObject);
begin
  If (not Setting) then
  begin
    Setting:=True;
    Mixer.SetVolume                  (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer(CheckBox.Checked));
    Setting:=False;
  end;
end;
procedure TForm0.CheckBoxClick(Sender: TObject);
begin
  If not Setting then
  begin
    Setting:=True;
    Mixer.SetVolume                  (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer(CheckBox.Checked));
    Setting:=False;
  end;
end;
procedure TForm0.ComboBox3Change(Sender: TObject);
var A:Integer;
begin
  If (ComboBox3.ItemIndex >= 0) AND (ComboBox3.ItemIndex < Mixer.MixerCount)
then
    Mixer.MixerId := ComboBox3.ItemIndex;
  ComboBox1.Items.Clear;
  If Mixer.MixerCount>0 then
  begin
    For A:=0 to Mixer.Destinations.Count-1 do
      ComboBox1.Items.Add (Mixer.Destinations[A].Data.szName);
    If ComboBox1.Items.Count>0 then
    begin
      ComboBox1.ItemIndex:=0;    ComboBox1Change (Self);
    end;
  end
  else
   begin
    ComboBox1.OnChange:=nil;     ComboBox2.OnChange:=nil;
    TrackBar.OnChange:=nil;       CheckBox.OnClick:=nil;
    MessageDlg ('No mixer present in the system !',mtError,[mbOK],0);
  end;
  Setting:=False;
end;
procedure TForm0.Button1Click(Sender: TObject);
begin
```

КОПІЯ 2023

```
Form0.hide;  Form1.show;
end;
procedure TForm0.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Form0.hide; Form1.show;
end;
end.
```

КбПЗ_2023

Підпрограма виклику вікна налагодження звукової карти AudioSettings

```
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  jpeg, ExtCtrls, StdCtrls, ComCtrls;

type
  TForm2 = class(TForm)
    Image1: TImage;
    RichEdit1: TRichEdit;
    ProgressBar1: TProgressBar;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.DFM}

end.
```

КбПЗ_2023

Підпрограма реалізації інтернет радіо

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls, jpeg, ComCtrls;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    ListBox1: TListBox;
    Animate1: TAnimate;
    Panel4: TPanel;
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    BitBtn3: TBitBtn;
    Image1: TImage;
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses usimple;

{$R *.DFM}

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
form3.hide;
end;

procedure TForm3.BitBtn1Click(Sender: TObject);
begin
Animate1.Active:=true;
form1.telefoon1.placecall(form1.edit1.text);
end;

procedure TForm3.BitBtn3Click(Sender: TObject);
begin
Form1.Telefoon1.calling:=false;
Animate1.Active:=false;
end;

end.
```

КПЗ_2023

Підпрограма парольного захисту

```
unit Unit4;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Mask, ExtCtrls;
type
  TForm4 = class(TForm)
    Panel1: TPanel;
    M: TMaskEdit;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form4: TForm4;
implementation
uses usimple;
VAR
  KEY:string='2%#$T%&(*qwrdas@@@#45';
  DAT:string='VUW';
{$R *.DFM}
        function Crypt(Text,Key: String; Encode: boolean): String;
        var
          i, KeyLength: integer;
          Sign: ShortInt;
        begin
         KeyLength:=Length(Key);
         if Encode then Sign :=-1 else Sign:=1;
         for i:=1 to Length(Text) do
          Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
          Result:=Text;
        end;
procedure TForm4.BitBtn1Click(Sender: TObject);
var
  F: TextFile;
  i:integer;
  s:string;
  Y:string;
  UUU:string;
begin
if FileExists('main.dat')=TRUE then
 begin
   AssignFile(F, 'main.dat');
   Reset(F);
   Readln(F, S);
   CloseFile(F);
UUU:=Crypt(M.text,KEY,false);
if UUU=S then
      begin
           Form4.hide;
           Form1.show;
           MessageDlg('Дякую, пароль вірний!',mtInformation,[mbOK],0);
      end
      else MessageDlg('Введіть пароль!',mtInformation,[mbOK],0);
 end
      else       begin
    MessageDlg('Файл       main.dat       не       знайдено!       завершення
програми',mtInformation,[mbOK],0);
    Application.Terminate;
      end;
end;
end.
```

КОПІЯ 2023

```
unit AMixer;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  MMSystem;

type
  TAudioMixer=class;

  TPListFreeItemNotify=procedure (Pntr:Pointer) of object;
  TMixerChange=procedure (Sender:TObject;MixerH:HMixer;ID:Integer) of object;
    {MixerH is handle of mixer, which sent this message.
     ID is ID of changed item (line or control).}

  TPointerList=class(TObject)
  private
    FOnFreeItem:TPListFreeItemNotify;
    Items:Tlist;
  protected
    function GetPointer (Ind:Integer):Pointer;
    function GetCount :integer;
  public
    constructor Create;
    destructor Destroy; override;
    procedure Clear;
    procedure Add (Pntr:Pointer);
    property Count:Integer read GetCount;
    property Pointer[Ind:Integer]:Pointer read GetPointer; default;
    property    OnFreeItem:TPListFreeItemNotify    read    FOnFreeItem    write
FOnFreeItem;
  end;

  TMixerControls=class(TObject)
  private
    heap:pointer;
    FControls:TPointerList;
  protected
    function GetControl (Ind:Integer):PMixerControl;
    function GetCount:Integer;
  public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Control[Ind:Integer]:PMixerControl read GetControl; default;
    property Count:Integer read GetCount;
  end;

  TMixerConnection=class(TObject)
  private
    XMixer:TAudioMixer;
    FData:TMixerLine;
    FControls:TMixerControls;
  public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Controls:TMixerControls read FControls;
    property Data:TMixerLine read FData;
  end;

  TMixerConnections=class(TObject)
  private
    XMixer:TAudioMixer;
    FConnections:TPointerList;
  protected
    procedure DoFreeItem (Pntr:Pointer);
    function GetConnection (Ind:Integer):TMixerConnection;
```

КБТЗ – 2023

```
    function GetCount:Integer;
  public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property  Connection[Ind:Integer]:TMixerConnection  read  GetConnection;
default;
    property Count:Integer read GetCount;
  end;

  TMixerDestination=class(TObject)
  private
    XMixer:TAudioMixer;
    FData:TMixerLine;
    FControls:TMixerControls;
    FConnections:TMixerConnections;
  public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Connections:TMixerConnections read FConnections;
    property Controls:TMixerControls read FControls;
    property Data:TMixerLine read FData;
  end;

  TMixerDestinations=class(TObject)
  private
    FDestinations:TPointerList;
  protected
    function GetDestination (Ind:Integer):TMixerDestination;
    procedure DoFreeItem (Pntr:Pointer);
    function GetCount:Integer;
  public
    constructor Create (AMixer:TAudioMixer);
    destructor Destroy; override;
    property Count:Integer read GetCount;
    property  Destination[Ind:Integer]:TMixerDestination  read  GetDestination;
default;
  end;

  TAudioMixer = class(TComponent)
  private
    XWndHandle:HWnd;

    FDestinations:TMixerDestinations;
    FMixersCount:Integer;
    FMixerHandle:HMixer;
    FMixerId:Integer;
    FMixerCaps:TMixerCaps;
    FDriverVersion: MMVERSION;
    FManufacturer: String;
    FProductId: Word;
    FNumberOfLine: Integer;
    FProductName: String;
    FOnLineChange:TMixerChange;
    FOnControlChange:TMixerChange;
  protected
    procedure SetMixerId (Value:Integer);
    procedure MixerCallBack (var Msg:TMessage);
    procedure CloseMixer;
  published
    constructor Create (AOwner:TComponent); override;
    destructor Destroy; override;
    property DriverVersion: MMVERSION read FDriverVersion;
    property ProductId: WORD read FProductId;
    property NumberOfLine: Integer read FNumberOfLine;
    property Manufacturer: string read FManufacturer;
    property ProductName: string read FProductName;
    property MixerId:Integer read FMixerId write SetMixerId;
      {Opened mixer - value must be in range 0..MixersCount-1
       If no mixer is opened this value is  -1}
```

КПЗ_2023

```
    property OnLineChange:TMixerChange read FOnLineChange write FOnLineChange;
    property OnControlChange:TMixerChange read FOnControlChange write
FOnControlChange;
  public
    function GetVolume (ADestination, AConnection:Integer; var LeftVol,
RightVol, Mute:Integer; var Stereo, VolDisabled, MuteDisabled,
MuteIsSelect:Boolean):Boolean;
    function SetVolume (ADestination, AConnection:Integer; LeftVol, RightVol,
Mute:Integer):Boolean;

    function GetPeak(ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer):Boolean;
    function GetMute(ADestination, AConnection:Integer; var
Mute:Boolean):Boolean;
    function SetMute(ADestination, AConnection:Integer; Mute:Boolean):Boolean;

    property Destinations:TMixerDestinations read FDestinations;
      {Ind must be in range 0..DestinationsCount-1}
    property MixerCaps:TMixerCaps read FMixerCaps;
    property MixerCount:Integer read FMixersCount;
      {Number of mixers present in system; mostly 1}
    property MixerHandle:HMixer read FMixerHandle;
      {Handle of opened mixer}
  end;

procedure Register;

implementation

{------------}
{TPointerList}
{------------}

constructor TPointerList.Create;
begin
  Items := TList.Create;
end;

destructor TPointerList.Destroy;
begin
  Clear;
  Items.Free;
end;

procedure TPointerList.Add (Pntr:Pointer);
begin
  Items.Add (Pntr);
end;

function TPointerList.GetPointer (Ind:Integer):Pointer;
begin
  Result := nil;
  If (Ind < Count) then
    Result := Items[Ind];
end;

procedure TPointerList.Clear;
var I:Integer;
begin
  for I := 0 to Items.Count-1 do begin
    If Assigned (FOnFreeItem) then
      FOnFreeItem (Items[I])
  end;
  Items.Clear;
end;

function TPointerList.GetCount:Integer;
begin
  Result := Items.Count;
```

КОПЗ_2023

```
end;

{--------------}
{TMixerControls}
{--------------}
constructor TMixerControls.Create (AMixer:TAudioMixer; AData:TMixerLine);
var MLC:TMixerLineControls;
    A,B:Integer;
    P:PMixerControl;
begin
  FControls := TPointerList.Create;
  GetMem (P, SizeOf(TMixerControl)*AData.cControls);
  heap := P;
  MLC.cbStruct := SizeOf(MLC);
  MLC.dwLineID := AData.dwLineID;
  MLC.cbmxctrl := SizeOf(TMixerControl);
  MLC.cControls := AData.cControls;
  MLC.pamxctrl := P;
  A          :=          MixerGetLineControls(AMixer.MixerHandle,          @MLC,
MIXER_GETLINECONTROLSF_ALL);
  If A = MMSYSERR_NOERROR then
  begin
    For B := 0 to AData.cControls-1 do
    begin
      FControls.Add (P);
      P := PMixerControl (DWORD(P) + sizeof (TMixerControl));
    end;
  end;
end;

destructor TMixerControls.Destroy;
begin
  FControls.free;
  freemem(heap);
  inherited;
end;

function TMixerControls.GetControl (Ind:Integer):PMixerControl;
begin
  Result := FControls.Pointer[Ind];
end;

function TMixerControls.GetCount:Integer;
begin
  Result := FControls.Count;
end;

{----------------}
{TMixerConnection}
{----------------}

constructor TMixerConnection.Create (AMixer:TAudioMixer; AData:TMixerLine);
begin
  FData := AData;
  XMixer := AMixer;
  FControls := TMixerControls.Create (AMixer, AData);
end;

destructor TMixerConnection.Destroy;
begin
  FControls.Free;
  inherited;
end;

{-----------------}
{TMixerConnections}
{-----------------}

constructor TMixerConnections.Create (AMixer:TAudioMixer; AData:TMixerLine);
```

КОПЗ – 2023

```
var A,B:Integer;
    ML:TMixerLine;
begin
  XMixer := AMixer;
  FConnections := TPointerList.Create;
  FConnections.OnFreeItem := Dofreeitem;
  ML.cbStruct := SizeOf(TMixerLine);
  ML.dwDestination := AData.dwDestination;
  For A := 0 to AData.cConnections-1 do
  begin
    ML.dwSource := A;
    B        :=        MixerGetLineInfo        (AMixer.MixerHandle,        @ML,
MIXER_GETLINEINFOF_SOURCE);
    If B = MMSYSERR_NOERROR then
      FConnections.Add (Pointer(TMixerConnection.Create (XMixer, ML)));
  end;
end;

destructor TMixerConnections.Destroy;
begin
  FConnections.Free;
  inherited;
end;

procedure TMixerConnections.DoFreeItem (Pntr:Pointer);
begin
  TMixerConnection(Pntr).Free;
end;

function TMixerConnections.GetConnection (Ind:Integer):TMixerConnection;
begin
  Result := FConnections.Pointer[Ind];
end;

function TMixerConnections.GetCount:Integer;
begin
  Result := FConnections.Count;
end;

{-----------------}
{TMixerDestination}
{-----------------}

constructor TMixerDestination.Create (AMixer:TAudioMixer; AData:TMixerLine);
begin
  FData := AData;
  XMixer := AMixer;
  FConnections := TMixerConnections.Create (XMixer, FData);
  FControls := TMixerControls.Create (XMixer, AData);
end;

destructor TMixerDestination.Destroy;
begin
  Fcontrols.Free;
  FConnections.Free;
  inherited;
end;

{-----------------}
{TMixerDestinations}
{-----------------}

constructor TMixerDestinations.Create (AMixer:TAudioMixer);
var A,B:Integer;
    ML:TMixerLine;
begin
  FDestinations := TPointerList.Create;
  FDestinations.OnFreeItem := DoFreeItem;
  if (AMixer = nil) then
```

RST3 - 2023

```
    Exit;
  For A := 0 to AMixer.MixerCaps.cDestinations-1 do
  begin
    ML.cbStruct := SizeOf(TMixerLine);
    ML.dwDestination := A;
    B        :=        MixerGetLineInfo        (AMixer.MixerHandle,        @ML,
MIXER_GETLINEINFOF_DESTINATION);
    If B = MMSYSERR_NOERROR then
      FDestinations.Add (Pointer(TMixerDestination.Create (AMixer, ML)));
  end;
end;

procedure TMixerDestinations.DoFreeItem (Pntr:Pointer);
begin
  TMixerDestination(Pntr).Free;
end;

destructor TMixerDestinations.Destroy;
begin
  FDestinations.Free;
  inherited;
end;

function TMixerDestinations.GetDestination (Ind:Integer):TMixerDestination;
begin
  Result := nil;
  If (Assigned (FDestinations)) then
    Result := FDestinations.Pointer[Ind];
end;

function TMixerDestinations.GetCount:Integer;
begin
  Result := FDestinations.Count;
end;

{-----------}
{TAudioMixer}
{-----------}

constructor TAudioMixer.Create (AOwner:TComponent);
begin
  inherited Create (AOwner);
  XWndHandle := AllocateHWnd (MixerCallBack);
  FMixersCount := mixerGetNumDevs;
  FMixerId := -1;
  if (FMixersCount = 0) then
    FDestinations := TMixerDestinations.Create (nil)
  else
  begin
    FDestinations := nil;
    SetMixerId (0);
  end;
end;

destructor TAudioMixer.Destroy;
begin
  CloseMixer;
  if XWndHandle <> 0 then
    DeAllocateHwnd (XWndHandle);
  inherited;
end;

procedure TAudioMixer.CloseMixer;
begin
  If FMixerId >= 0 then
  begin
    mixerClose (FMixerHandle);
    FMixerId := -1;
  end;
```

КОПІЯ 2023

```
  FDestinations.Free;
  FDestinations := nil;
end;

procedure TAudioMixer.SetMixerId (Value:Integer);
label AllOK;
begin
  If (Value < 0) OR (Value >= FMixersCount) then
    Exit;
  CloseMixer;

  If mixerOpen (@FMixerHandle, Value, XWndHandle, 0, CALLBACK_WINDOW OR
MIXER_OBJECTF_MIXER) = MMSYSERR_NOERROR then
    goto AllOK;

  // we will go here very rarely, but sometimes it could help
  If mixerOpen (@FMixerHandle, Value, XWndHandle, 0, CALLBACK_WINDOW) =
MMSYSERR_NOERROR then
    goto AllOK;
  If mixerOpen (@FMixerHandle, Value, 0, 0, 0) = MMSYSERR_NOERROR then
    goto AllOK;

  // an error has occured
  FMixerId := -1;
  FDestinations := TMixerDestinations.Create (nil);

  Exit;

AllOK:
  FMixerId := Value;
  mixerGetDevCaps (MixerId, @FMixerCaps, SizeOf (TMixerCaps));

  if FMixerCaps.wMid = MM_MICROSOFT then
    FManufacturer := 'Microsoft'
  else
    FManufacturer := IntToStr(FMixerCaps.wMid) + ' = Unknown';
  FDriverVersion := FMixerCaps.vDriverVersion;
  FProductId := FMixerCaps.wPid;
  FProductName := StrPas(FMixerCaps.szPName);
  FNumberOfLine := FMixerCaps.cDestinations;

  FDestinations := TMixerDestinations.Create (Self);
end;

procedure TAudioMixer.MixerCallBack (var Msg:TMessage);
begin
  case Msg.Msg of
    MM_MIXM_LINE_CHANGE:
        If Assigned (OnLineChange) then
          OnLineChange (Self, Msg.wParam, Msg.lParam);
    MM_MIXM_CONTROL_CHANGE:
        If Assigned (OnControlChange) then
          OnControlChange (Self, Msg.wParam, Msg.lParam);
    else
      Msg.Result   :=   DefWindowProc   (XWndHandle,   Msg.Msg,   Msg.WParam,
Msg.LParam);
  end;
end;

const MIXER_LONG_NAME_CHARS = 64;

type MIXERCONTROLDETAILS_LISTTEXT = record
       dwParam1:DWORD;
       dwParam2:DWORD;
       szName:Array [0..MIXER_LONG_NAME_CHARS-1] of Char;
     end;

type ListTextArray = array [0..1000] of MIXERCONTROLDETAILS_LISTTEXT;
```

KOTB 2023

```
function TAudioMixer.GetVolume (ADestination,AConnection:Integer;var LeftVol,
RightVol,     Mute:Integer;var     Stereo,     VolDisabled,     MuteDisabled,
MuteIsSelect:Boolean):Boolean;
var MD:TMixerDestination;
    MC:TMixerConnection;
    Cntrls:TMixerControls;
    MCD:TMixerControlDetails;
    Cntrl:PMixerControl;
    A,B:Integer;
    ML:TMixerLine;
    details:array [0..100] of Integer;
    ltext:^ListTextArray;
    MCDText:TMixerControlDetails;

begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  Stereo := False;
  MuteDisabled := True;
  VolDisabled := True;
  LeftVol := -1;
  RightVol := -1;
  Mute := -1;
  MuteIsSelect := False;
  MD := Destinations[ADestination];
  MC := nil;
  If AConnection <> -1 then
    MC := MD.Connections[AConnection];

  If MD <> nil then
  begin

    Result := True;

//      If Mute = -1 then
    begin
      If AConnection <> -1 then
      begin
        Cntrls := MD.Controls;
        ML := MD.Data;
        If (MC <> nil) AND (Cntrls <> nil) then
        begin
          A := 0;
          while (Mute = -1) AND (A < Cntrls.Count) do
          begin
            Cntrl := Cntrls[A];
//                If (Cntrl.dwControlType AND MIXERCONTROL_CONTROLTYPE_MIXER =
MIXERCONTROL_CONTROLTYPE_MIXER) OR
//                   (Cntrl.dwControlType AND MIXERCONTROL_CONTROLTYPE_MUX =
MIXERCONTROL_CONTROLTYPE_MUX) then

//                  If (Cntrl.dwControlType AND MIXERCONTROL_CT_CLASS_MASK =
MIXERCONTROL_CT_CLASS_LIST) then

            If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
               // Mux is similar to mixer, but only one line can be selected
at a time
            begin
              MCD.cbStruct := SizeOf(TMixerControlDetails);
              MCD.dwControlID := Cntrl.dwControlID;
              If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                MCD.cChannels := 1
              else
                MCD.cChannels := ML.cChannels;
              If   Cntrl.fdwControl   AND   MIXERCONTROL_CONTROLF_MULTIPLE   =
MIXERCONTROL_CONTROLF_MULTIPLE then
                MCD.cMultipleItems := Cntrl.cMultipleItems
              else
```

КОТЗ – 2023

```
              MCD.cMultipleItems := 0;
            MCD.cbDetails := 4;
            MCD.paDetails := @Details;
            B                    :=                    mixerGetControlDetails
(FMixerHandle,@MCD,MIXER_GETCONTROLDETAILSF_VALUE);
            If B <> MMSYSERR_NOERROR then
            begin
              Inc (A);
              continue;
            end;

            MCDText.cbStruct := sizeof (MCDText);
            MCDText.dwControlID := Cntrl.dwControlID;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
              MCDText.cChannels := 1
            else
              MCDText.cChannels := ML.cChannels;
            If   Cntrl.fdwControl   AND   MIXERCONTROL_CONTROLF_MULTIPLE   =
MIXERCONTROL_CONTROLF_MULTIPLE then
              MCDText.cMultipleItems := Cntrl.cMultipleItems
            else
              MCDText.cMultipleItems := 0;
            GetMem  (ltext,  MCDText.cChannels  *  MCDText.cMultipleItems  *
sizeof (MIXERCONTROLDETAILS_LISTTEXT));
            MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
            MCDText.paDetails := ltext;
            B    :=    mixerGetControlDetails    (FMixerHandle,    @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
            If B <> MMSYSERR_NOERROR then
            begin
              FreeMem (ltext);
              Inc (A);
              continue;
            end;
            B := MCD.cChannels - 1;
            while (B < integer(MCD.cMultipleItems)) do
            begin
              if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                break;
              Inc (B, MCD.cChannels);
            end;
            FreeMem (ltext);

            If (B < integer (MCD.cMultipleItems)) then
            begin
              Mute := Details[B];
              MuteDisabled          :=          Cntrl.fdwControl          AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
              MuteIsSelect := True;
              break;
            end;
          end;
          Inc (A);
        end;
      end;
    end;

    If AConnection = -1 then
    begin
      Cntrls := MD.Controls;
      ML := MD.Data;
    end
    else
    begin
      If MC <> nil then
      begin
        Cntrls := MC.Controls;
        ML := MC.Data;
```

КОПІЯ – 2023

```
      end
      else
        Cntrls := nil;
    end;
    If Cntrls <> nil then
    begin
      A := 0;
      while ((LeftVol = -1) OR (Mute = -1)) AND (A < Cntrls.Count) do
      begin
        Cntrl := Cntrls[A];
        If Cntrl <> nil then
        begin
          If ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) OR
              ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) AND (Mute
= -1))) AND
             (Cntrl.fdwControl     AND     MIXERCONTROL_CONTROLF_MULTIPLE     <>
MIXERCONTROL_CONTROLF_MULTIPLE)
             then
          begin
            if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) then
              MCD.cbStruct := SizeOf(TMixerControlDetails)
            else
              MCD.cbStruct := SizeOf(TMixerControlDetails);
            MCD.dwControlID := Cntrl.dwControlID;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
              MCD.cChannels := 1
            else
              MCD.cChannels := ML.cChannels;
            MCD.cMultipleItems := 0;
            MCD.cbDetails := SizeOf(Integer);
            MCD.paDetails := @details;
            B      :=      mixerGetControlDetails      (FMixerHandle,      @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
            If B = MMSYSERR_NOERROR then
            begin
              If  (Cntrl.dwControlType  =  MIXERCONTROL_CONTROLTYPE_VOLUME)  AND
(LeftVol = -1) then
              begin
                VolDisabled          :=          Cntrl.fdwControl          AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
                LeftVol := details[0];
                If MCD.cChannels > 1 then
                begin
                  RightVol := Details[1];
                  Stereo := True;
                end
                else
                  RightVol := LeftVol;
              end
              else  If  (Cntrl.dwControlType  =  MIXERCONTROL_CONTROLTYPE_MUTE)
AND (Mute = -1) then
              begin
                MuteDisabled          :=          Cntrl.fdwControl          AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
                If Details[0] <> 0 then
                  Mute := 1
                else
                  Mute := 0;
              end
// NEW ->
(*              else If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF)
AND (Mute = -1) then
              begin
                MuteDisabled          :=          Cntrl.fdwControl          AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
                If Details[0] <> 0 then
                  Mute := 1
                else
                  Mute := 0;
```

КОПІЯ 2023

```
                MuteIsSelect := True;
              end;*)
// <- NEW
            end;
          end;
        end;
        Inc (A);
      end;

{      If LeftVol = -1 then
        VolDisabled := True;
      If Mute = -1 then
        MuteDisabled := True;}
    end;
  end;
end;

function TAudioMixer.SetVolume (ADestination, AConnection:Integer; LeftVol,
RightVol, Mute:Integer):Boolean;
var MD:TMixerDestination;
    MC:TMixerConnection;
    Cntrls:TMixerControls;
    MCD:TMixerControlDetails;
    Cntrl:PMixerControl;
    A,B:Integer;
    ML:TMixerLine;
    details:array [0..100] of Integer;
    VolSet,MuteSet:Boolean;
    ltext:^ListTextArray;
    MCDText:TMixerControlDetails;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  MC := nil;
  MD := Destinations[ADestination];
  If MD <> nil then
  begin
    If AConnection <> -1 then
      MC := MD.Connections[AConnection];

    VolSet := LeftVol = -1;
    MuteSet := Mute = -1;
    Result := True;

    If not MuteSet then
    begin
      If AConnection <> -1 then
      begin
        Cntrls := MD.Controls;
        ML := MD.Data;
        If (MC <> nil) AND (Cntrls <> nil) then
        begin
          A := 0;
          while not MuteSet AND (A < Cntrls.Count) do
          begin
            Cntrl := Cntrls[A];
            If (*(Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR*)
               (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
            begin
              MCD.cbStruct := SizeOf(TMixerControlDetails);
              MCD.dwControlID := Cntrl.dwControlID;
              If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                MCD.cChannels := 1
              else
                MCD.cChannels := ML.cChannels;
              If   Cntrl.fdwControl   AND   MIXERCONTROL_CONTROLF_MULTIPLE   =
MIXERCONTROL_CONTROLF_MULTIPLE then
                MCD.cMultipleItems := Cntrl.cMultipleItems
```

КОПЗ_2023

```
              else
                MCD.cMultipleItems := 0;
              MCD.cbDetails := 4;
              MCD.paDetails := @Details;
              MuteSet := True;
              mixerGetControlDetails           (FMixerHandle,           @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
              if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
                For B := 0 to Cntrl.cMultipleItems-1 do
                  Details[B] := 0;

              GetMem  (ltext,  MCD.cChannels  *  MCD.cMultipleItems  *  sizeof
(MIXERCONTROLDETAILS_LISTTEXT));
              MCDText.cbStruct := sizeof (MCDText);
              MCDText.dwControlID := Cntrl.dwControlID;
              MCDText.cChannels := MCD.cChannels;
              MCDText.cMultipleItems := MCD.cMultipleItems;
              MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
              MCDText.paDetails := ltext;
              mixerGetControlDetails         (FMixerHandle,         @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
              B := MCD.cChannels - 1;
              while (B < integer (MCD.cMultipleItems)) do
              begin
                if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                  break;
                Inc (B, MCD.cChannels);
              end;
              FreeMem (ltext);

              If (B < integer (MCD.cMultipleItems)) then
              begin
                Details[B] := Mute;
                mixerSetControlDetails         (FMixerHandle,           @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
                break;
              end;
            end;
            Inc (A);
          end;
        end;
      end;
    end;



    If AConnection = -1 then
    begin
      Cntrls := MD.Controls;
      ML := MD.Data;
    end
    else
    begin
      If MC <> nil then
      begin
        Cntrls := MC.Controls;
        ML := MC.Data;
      end
      else
        Cntrls := nil;
    end;
    If Cntrls <> nil then
    begin

      A := 0;
      while (not VolSet OR not MuteSet) AND (A < Cntrls.Count) do
      begin
        Cntrl := Cntrls[A];
        If Cntrl <> nil then
```

КБП3 – 2023

```
        begin
          If (((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) AND not
VolSet) OR
              ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) AND not
MuteSet) (* NEW -> *) (*OR
              ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF) AND not
MuteSet)*) (* <- NEW *)) AND
             (Cntrl.fdwControl    AND    MIXERCONTROL_CONTROLF_MULTIPLE    <>
MIXERCONTROL_CONTROLF_MULTIPLE)
            then
          begin
            MCD.cbStruct := SizeOf(TMixerControlDetails);
            MCD.dwControlID := Cntrl.dwControlID;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
              MCD.cChannels := 1
            else
              MCD.cChannels := ML.cChannels;
            MCD.cMultipleItems := 0;
            MCD.cbDetails := SizeOf(Integer);
            MCD.paDetails := @Details;
            If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) then
            begin
              Details[0] := LeftVol;
              If RightVol = -1 then
                Details[1] := LeftVol
              else
                Details[1] := RightVol;
              VolSet := True;
            end
            else if ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) (*
NEW -> *) (* OR
                     (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF) *)
(* <- NEW *)) then
            begin
              For B := 0 to MCD.cChannels - 1 do
                Details[B] := Mute;
              MuteSet := True;
            end;
            mixerSetControlDetails            (FMixerHandle,            @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
          end;
        end;
        Inc (A);
      end;

    end;
  end;
end;

function TAudioMixer.GetMute(ADestination, AConnection: Integer; var Mute:
Boolean):Boolean;
var
  MD : TMixerDestination;
  MC : TMixerConnection;
  mlcMixerLineControlsMute : TMIXERLINECONTROLS;
  mcdMixerDataMute : TMIXERCONTROLDETAILS;
  pmcMixerControlMute : PMIXERCONTROL;
  pmcdsMixerDataUnsignedMute : PMIXERCONTROLDETAILSBOOLEAN;
  mlMixerLine : TMixerLine;
  Cntrl:PMixerControl;
  Cntrls:TMixerControls;
  ML:TMixerLine;
  A,B:Integer;
  details:array [0..100] of Integer;
  ltext:^ListTextArray;
  MCDText:TMixerControlDetails;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
```

КПЗ_2023

```
    Exit;
  MC := nil;
  Mute := False;
  MD := Destinations[ADestination];
  if MD <> nil then
  begin
    if AConnection = -1 then
      mlMixerLine := MD.Data
    else
    begin
      MC := MD.Connections[AConnection];
      if MC <> nil then
        mlMixerLine := MC.Data
      else
        Exit;
    end;

    GetMem(pmcMixerControlMute, SizeOf(TMIXERCONTROL));
    GetMem(pmcdsMixerDataUnsignedMute, SizeOf(TMIXERCONTROLDETAILSBOOLEAN));

    with mlcMixerLineControlsMute do
    begin
      cbStruct := SizeOf(TMIXERLINECONTROLS);
      dwLineID := mlMixerLine.dwLineID;
      dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
      cControls := 1;
      cbmxctrl := SizeOf(TMIXERCONTROL);
      pamxctrl := pmcMixerControlMute;
    end;

    if      (mixerGetLineControls(FMixerHandle,      @mlcMixerLineControlsMute,
MIXER_GETLINECONTROLSF_ONEBYTYPE) = MMSYSERR_NOERROR) then
    begin
      with mcdMixerDataMute do
      begin
        cbStruct := SizeOf(TMIXERCONTROLDETAILS);
        dwControlID := pmcMixerControlMute^.dwControlID;
        cChannels := 1;
        cMultipleItems := 0;
        cbDetails := SizeOf(TMIXERCONTROLDETAILSBOOLEAN);
        paDetails := pmcdsMixerDataUnsignedMute;
      end;

      if          mixerGetControlDetails(FMixerHandle,         @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) = MMSYSERR_NOERROR then
      begin
        Mute := pmcdsMixerDataUnsignedMute^.fValue = 1;
        Result := True;
      end;
    end
    else
    begin
      If (AConnection <> -1) then
      begin
        Cntrls := MD.Controls;
        ML := MD.Data;
        If (MC <> nil) AND (Cntrls <> nil) then
        begin
          A := 0;
          while (Result = False) AND (A < Cntrls.Count) do
          begin
            Cntrl := Cntrls[A];
            If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR
               (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
            begin
              mcdMixerDataMute.cbStruct := SizeOf(TMixerControlDetails);
              mcdMixerDataMute.dwControlID := Cntrl.dwControlID;
              If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                mcdMixerDataMute.cChannels := 1
```

КПЗ – 2023

```
              else
                mcdMixerDataMute.cChannels := ML.cChannels;
              If   Cntrl.fdwControl   AND   MIXERCONTROL_CONTROLF_MULTIPLE   =
MIXERCONTROL_CONTROLF_MULTIPLE then
                mcdMixerDataMute.cMultipleItems := Cntrl.cMultipleItems
              else
                mcdMixerDataMute.cMultipleItems := 0;
              mcdMixerDataMute.cbDetails := 4;
              mcdMixerDataMute.paDetails := @Details;
              mixerGetControlDetails     (FMixerHandle,     @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE);

              GetMem        (ltext,        mcdMixerDataMute.cChannels        *
mcdMixerDataMute.cMultipleItems * sizeof (MIXERCONTROLDETAILS_LISTTEXT));
              MCDText.cbStruct := sizeof (MCDText);
              MCDText.dwControlID := Cntrl.dwControlID;
              MCDText.cChannels := mcdMixerDataMute.cChannels;
              MCDText.cMultipleItems := mcdMixerDataMute.cMultipleItems;
              MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
              MCDText.paDetails := ltext;
              mixerGetControlDetails          (FMixerHandle,          @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
              B := mcdMixerDataMute.cChannels - 1;
              while (B < integer (mcdMixerDataMute.cMultipleItems)) do
              begin
                if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                  break;
                Inc (B, mcdMixerDataMute.cChannels);
              end;
              FreeMem (ltext);

              If (B < integer (mcdMixerDataMute.cMultipleItems)) then
              begin
                Result := True;
                Mute := Details[B] <> 0;
                break;
              end;
            end;
            Inc (A);
          end;
        end;
      end;
    end;

    FreeMem(pmcdsMixerDataUnsignedMute);
    FreeMem(pmcMixerControlMute);
  end;
end;

function   TAudioMixer.SetMute(ADestination,   AConnection:   Integer;   Mute:
Boolean):Boolean;
var
  MD : TMixerDestination;
  MC : TMixerConnection;
  mlcMixerLineControlsMute : TMIXERLINECONTROLS;
  mcdMixerDataMute : TMIXERCONTROLDETAILS;
  pmcMixerControlMute : PMIXERCONTROL;
  pmcdsMixerDataUnsignedMute : PMIXERCONTROLDETAILSBOOLEAN;
  mlMixerLine : TMixerLine;
  Cntrl:PMixerControl;
  Cntrls:TMixerControls;
  ML:TMixerLine;
  A,B:Integer;
  details:array [0..100] of Integer;
  ltext:^ListTextArray;
  MCDText:TMixerControlDetails;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
```

КОПИЯ 2023

```
    Exit;
  MC := nil;
  MD := Destinations[ADestination];
  if MD <> nil then
  begin
    if AConnection = -1 then
      mlMixerLine := MD.Data
    else
    begin
      MC := MD.Connections[AConnection];
      if MC <> nil then
        mlMixerLine := MC.Data
      else
        Exit;
    end;

    GetMem(pmcMixerControlMute, SizeOf(TMIXERCONTROL));
    GetMem(pmcdsMixerDataUnsignedMute, SizeOf(TMIXERCONTROLDETAILSBOOLEAN));

    with mlcMixerLineControlsMute do
    begin
      cbStruct := SizeOf(TMIXERLINECONTROLS);
      dwLineID := mlMixerLine.dwLineID;
      dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
      cControls := 0;
      cbmxctrl := SizeOf(TMIXERCONTROL);
      pamxctrl := pmcMixerControlMute;
    end;

    if      (mixerGetLineControls(FMixerHandle,    @mlcMixerLineControlsMute,
MIXER_GETLINECONTROLSF_ONEBYTYPE) = MMSYSERR_NOERROR) then
    begin
      with mcdMixerDataMute do
      begin
        cbStruct := SizeOf(TMixerControlDetails);
        dwControlID := pmcMixerControlMute^.dwControlID;
        cChannels := 1;
        cMultipleItems := 0;
        cbDetails := SizeOf(TMIXERCONTROLDETAILSBOOLEAN);
        paDetails := pmcdsMixerDataUnsignedMute;
      end;

      if Mute then
        pmcdsMixerDataUnsignedMute^.fValue := 1
      else
        pmcdsMixerDataUnsignedMute^.fValue := 0;

      if
(mixerSetControlDetails(FMixerHandle,@mcdMixerDataMute,MIXER_SETCONTROLDETAILS
F_VALUE) = MMSYSERR_NOERROR) then
        Result := True;
    end
    else
    begin
      If (AConnection <> -1) then
      begin
        Cntrls := MD.Controls;
        ML := MD.Data;
        If (MC <> nil) AND (Cntrls <> nil) then
        begin
          A := 0;
          while (Result = False) AND (A < Cntrls.Count) do
          begin
            Cntrl := Cntrls[A];
            If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR
               (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
            begin
              mcdMixerDataMute.cbStruct := SizeOf(TMixerControlDetails);
              mcdMixerDataMute.dwControlID := Cntrl.dwControlID;
```

Kotb_2023

```
              If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                mcdMixerDataMute.cChannels := 1
              else
                mcdMixerDataMute.cChannels := ML.cChannels;
              If   Cntrl.fdwControl   AND   MIXERCONTROL_CONTROLF_MULTIPLE   =
MIXERCONTROL_CONTROLF_MULTIPLE then
                mcdMixerDataMute.cMultipleItems := Cntrl.cMultipleItems
              else
                mcdMixerDataMute.cMultipleItems := 0;
              mcdMixerDataMute.cbDetails := 4;
              mcdMixerDataMute.paDetails := @Details;
              if   (mixerGetControlDetails   (FMixerHandle,   @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) <> MMSYSERR_NOERROR) then
              begin
                Inc (A);
                continue;
              end;
              if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
                For B := 0 to Cntrl.cMultipleItems-1 do
                  Details[B] := 0;
              If Mute then
              begin
                GetMem       (ltext,       mcdMixerDataMute.cChannels       *
mcdMixerDataMute.cMultipleItems * sizeof (MIXERCONTROLDETAILS_LISTTEXT));
                MCDText.cbStruct := sizeof (MCDText);
                MCDText.dwControlID := Cntrl.dwControlID;
                MCDText.cChannels := mcdMixerDataMute.cChannels;
                MCDText.cMultipleItems := mcdMixerDataMute.cMultipleItems;
                MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
                MCDText.paDetails := ltext;
                mixerGetControlDetails       (FMixerHandle,       @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
                B := mcdMixerDataMute.cChannels - 1;
                while (B < integer (mcdMixerDataMute.cMultipleItems)) do
                begin
                  if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                    break;
                  Inc (B, mcdMixerDataMute.cChannels);
                end;
                FreeMem (ltext);

                If (B < integer (mcdMixerDataMute.cMultipleItems)) then
                  Details[B] := 1;
              end;
              if   (mixerSetControlDetails   (FMixerHandle,   @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) = MMSYSERR_NOERROR) then
              begin
                Result := True;
                break;
              end;
            end;
            Inc (A);
          end;
        end;
      end;
    end;

    FreeMem(pmcdsMixerDataUnsignedMute);
    FreeMem(pmcMixerControlMute);
  end;
end;

function TAudioMixer.GetPeak (ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer): Boolean;
var
  MD : TMixerDestination;
  MC : TMixerConnection;
  mcdMixerDataPeak : TMIXERCONTROLDETAILS;
  pmcMixerControlPeak : PMIXERCONTROL;
```

КОПИЯ 2023

```
{  pmcdsMixerDataSignedPeak : PMIXERCONTROLDETAILSSIGNED;}
  mlMixerLine : TMixerLine;
  A:Integer;
  Cntrls:TMixerControls;
  Details:Array [1..100] of Integer;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  LeftPeak := 0;
  RightPeak := 0;
  MD := Destinations[ADestination];
  if MD <> nil then
  begin
    if AConnection = -1 then
    begin
      mlMixerLine := MD.Data;
      Cntrls := MD.Controls;
    end
    else
    begin
      MC := MD.Connections[AConnection];
      if MC <> nil then
      begin
        mlMixerLine := MC.Data;
        Cntrls := MC.Controls;
      end
      else
        Exit;
    end;
    GetMem(pmcMixerControlPeak, SizeOf(TMIXERCONTROL));

    A := 0;
    while (A < Cntrls.Count) do
    begin
      If    (Cntrls[A].dwControlType    AND    MIXERCONTROL_CT_CLASS_MASK)    =
MIXERCONTROL_CT_CLASS_METER then
        break;
      Inc (A);
    end;
    If A = Cntrls.Count then
    begin
      FreeMem(pmcMixerControlPeak);
      Exit;
    end;

    with mcdMixerDataPeak do
    begin
      cbStruct := SizeOf(TMIXERCONTROLDETAILS);
      dwControlID := Cntrls[A].dwControlID;
      cChannels := mlMixerLine.cChannels;
      If    (Cntrls[A].fdwControl    AND    MIXERCONTROL_CONTROLF_MULTIPLE)    =
MIXERCONTROL_CONTROLF_MULTIPLE then
        cMultipleItems:=Cntrls[A].cMultipleItems
      else
        cMultipleItems:=0;
      cbDetails := SizeOf(TMIXERCONTROLDETAILSSIGNED);
      paDetails := @Details;
    end;
    if
(mixerGetControlDetails(FMixerHandle,@mcdMixerDataPeak,MIXER_GETCONTROLDETAILS
F_VALUE) = MMSYSERR_NOERROR) then
    begin
      LeftPeak := Details[1];
      if mlMixerLine.cChannels = 2 then
        RightPeak := Details[2]
      else
        RightPeak := LeftPeak;
      Result := True;
```

КПЗ – 2023

```
    end;
    FreeMem(pmcMixerControlPeak);
   end;
end;
procedure Register;
begin
  RegisterComponents('Samples', [TAudioMixer]);
end;
end.
```

КбПЗ_2023