

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
« ____ » _____ 2022 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи перегляду
хмарних сервісів”

Виконав здобувач вищої освіти
II курсу, групи КН-21М-1,4
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Кузьменко Є.О.
« ____ » _____ 2022 р.

Керівник проекту
кандидат технічних наук
_____ Буравченко К.О.
« ____ » _____ 2022 р.
Рецензент _____

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2022 року

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

Кузьменку Єгору Олексійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи перегляду хмарних сервісів*

2. Керівник роботи *Буравченко Костянтин Олегович, канд. техн. наук*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 18-13 від 17.08.2022 року

3. Строк подання студентом роботи до захисту *10.12.2022 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи перегляду хмарних сервісів*

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2022	14.11.2022
Охорона праці	Оришака О.В.	06.10.2022	16.11.2022

7. Дата видачі завдання « 6 » вересня 2022 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	10.10.2022 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2022 р.	
3.	Розробка моделі компонента	20.10.2022 р.	
4.	Розробка структур даних	25.10.2022 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2022 р.	
6.	Програмування алгоритмів	10.11.2022 р.	
7.	Розрахунок економічної ефективності	13.11.2022 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2022 р.	
9.	Оформлення ПЗ	17.11.2022 р.	
10.	Попередній захист роботи	10.12.2022 р.	

Дата видачі завдання
« 6 » вересня 2022 р.

Підпис керівника

Буравченко К.О.
(прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2022 р.

Підпис здобувача

Кузьменко Є.О.
(прізвище та ініціали)

АНОТАЦІЯ

Кузьменко Є.О. Дослідження та програмна реалізація системи перегляду хмарних сервісів. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2022.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи перегляду хмарних сервісів.

Метою розробки є дослідження та програмна реалізація системи перегляду хмарних сервісів.

Об'єктом дослідження є процес перегляду хмарних сервісів.

Предметом дослідження є методи перегляду хмарних сервісів.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи перегляду хмарних сервісів.

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 10.4.

Ключові слова: комп'ютерні науки, хмарні сервіси

ABSTRACT

Kuzmenko E.O. Research and software implementation of cloud services browsing system. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the cloud services viewing system.

The goal of the development is the research and software implementation of the cloud services viewing system.

The object of the study is the process of viewing cloud services.

The subject of the study is methods of viewing cloud services.

Research methods are based on methods of cloud technologies, methods of mathematical statistics, methods of software development.

The result of the work is the software implementation of the system for viewing cloud services.

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 10.4 environment.

Keywords: computer science, cloud services

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	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.....	42
3.3 Розробка функціональної схеми	44
3.4 Розробка діаграми процесів.....	45
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	47
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	47
4.2 Захист розробленого програмного забезпечення.....	58
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	61
6 НАУКОВА НОВИЗНА	66

БКРМ-122.22.0004.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Кузьменко Є.О.			Дослідження та програмна реалізація системи перегляду хмарних сервісів	Літ.	Аркуш	Аркушів
Перев.		Буравченко К.О.				М	1	107
Н.контр.		Гермак В.С.			ЦНТУ КН-21М-1,4			
Затв.		Смірнов О.А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	67
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	67
7.2 Розрахунок трудомісткості розробки програмної продукції.....	69
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	71
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	76
7.5 Визначення собівартості розробки та ціни програмної продукції.....	80
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	83
7.7 Визначення експлуатаційних витрат.....	83
7.8 Визначення економічної ефективності програмної продукції.....	85
7.9 Висновок.....	87
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	89
8.1 Вступ.....	89
8.2 Характеристика умов праці програміста	91
8.3 Розробка заходів з умов поліпшення охорони праці.....	93
8.4 Розрахункова частина	94
8.5 Висновки до розділу.....	96
9 ОСНОВНІ ВИСНОВКИ.....	97
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	99

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- БД – база даних
- ЕОМ – електронно-обчислювальна машина
- ОЗП – оперативний запам'ятовувачий пристрій
- ОС – операційна система
- ПЗ – програмне забезпечення
- СУБД – система управління базою даних
- Microsoft Edge – Microsoft Edge

Кафедра КБПЗ – 2022 рік

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Сучасний світ неможливо представити без глобальної мережі Інтернет. Через Інтернет зараз відбувається доступ до різного виду інформації (починаючи від фільмів та музики й закінчуючи новинами та науковими статтями), спілкування між людьми, які розташовані на величезних відстанях один від іншого, покупки в Інтернет-магазинах та ще багато-багато інших функцій. Інтернет надає доступ до різного роду хмарних сервісів.

Але для того, щоб працювати з інтернетом необхідна програма доступу до веб-сторінок, з яких складається Інтернет. Такою програмою є веб-браузер, або, якщо казати по іншому – Інтернет-браузер, як складова системи перегляду хмарних сервісів. Веб-оглядач, або браузер, як складова системи перегляду хмарних сервісів – це програмне забезпечення для пошуку, перегляду веб-сайтів, тобто для запиту веб-сторінок (переважно з Інтернету), для їхньої обробки, висновку й переходу від однієї сторінки до іншої. Більшість браузерів також наділені здатностями до перегляду змісту FTP-серверів. Браузери постійно розвивалися із часів зародження Інтернету, і з його ростом ставали усе більше важливою програмою типового персонального комп'ютера. Нині браузер, як складова системи перегляду хмарних сервісів – комплексний додаток для обробки й виведення різних складових веб-сторінки, і для надання інтерфейсу між веб-сайтом і його відвідувачем. Практично всі популярні браузери поширюються безкоштовно або «у комплекті» з іншим додатком: Microsoft Edge (як невід'ємна частина Microsoft Windows), Mozilla Firefox, Opera, Safari (разом з Mac OS або безкоштовно для Windows). Крім цих браузерів існує ще велика кількість інших браузерів, які мають ті або інші функції, крім основної, перегляду веб-сторінок. Але не дивлячись на таке розмаїття браузерів, немає вітчизняного продукту, який міг би створити конкуренцію цим браузерам. У цьому магістерському проекті зроблена спроба створити такий браузер, як складова системи перегляду хмарних сервісів.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для роботи в Internet. Internet – це глобальна комп'ютерна мережа, що охоплює всю територію земної кулі й працююча по протоколу TCP/IP. Internet сьогодні-це не тільки величезна кількість комп'ютерів, але ще й неймовірна кількість людей, для яких мережа є принципово новим способом спілкування, що майже не має аналогів у світі. Людина – істота соціальна, і спілкування із собі подібними – одна з найперших його потреб. Звичайно, що саме зацікавить вас в Інтернет у першу чергу-люди або комп'ютери, залежить тільки від вас, але не буде перебільшенням сказати, що, виходячи в Internet, ви робите для себе доступним цілий мир.

Всі комп'ютери мережі Internet можна розділити на два типи: сервери й клієнти. Ваш комп'ютер є комп'ютером-клієнтом мережі Internet, тому що ви використовуєте ресурси Internet. Комп'ютери-сервери утворюють основу мережі й надають свої ресурси у використання іншим комп'ютерам.

Коли говорять, що комп'ютер підключений до мережі Internet, це означає, що цей комп'ютер за допомогою одного з основних засобів зв'язку – модему (Dial-Up підключення) або мережної карти з'єднаний із провайдером (службою доступу в Internet) і може звернутися до будь-якого комп'ютера мережі Internet. А під терміном Internet у цьому випадку розуміють множина серверів, до яких ваш комп'ютер має доступ і ресурсами яких може користуватися.

Одержуючи доступ у мережу Internet, ви підключаєтеся до різних серверів і одержуєте необхідну вам інформацію. «Усередині» Internet розташовується складна структура зв'язаних між собою комп'ютерних мереж, що дозволяє їм мати доступ до всіх комп'ютерів мережі.

Internet являє собою всесвітню мережу взаємозалежних комп'ютерних систем і ряду різних інформаційних служб, хоча з іншого боку – це звичайна система передачі даних з одного комп'ютера на іншій. Яку саме інформацію і як

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

ви будите передавати по мережі Internet, залежить цілком від вас,— аби тільки другий комп'ютер, з яким ви спілкуєтеся, вас зрозумів. Тому з моменту зародження Internet і до наших днів людьми періодично винаходяться всі нові й нові форми подання ресурсів і засобу доступу до них через Internet. В основі кожного такого нового засобу лежить свій мережний протокол.

Створивши такий новий протокол, його розроблювачі повинні написати програми двох видів – програми-сервери й програми – клієнти, які могли б працювати на різних комп'ютерах і обмінюватися даними по цьому протоколі через канали Internet. У функції програми-сервера входить зберігання й обслуговування даних, а також видача їх по запитах програм-клієнтів, що звертаються до неї. Програма-клієнт, у свою чергу, спілкується з «своїми» серверами на різних вузлах і передає отриману від них інформацію користувачеві.

Якщо новий протокол має ті або інші достоїнства, він починає поширюється: усе більше користувачів Internet обзаводяться програмами-клієнтами й усе більше вузлів мережі обладнають у себе сервери, що підтримують цей протокол. Роста й кількість інформації, доступної в такий спосіб. Іноді інформаційні ресурси, які раніше поширювалися з допомогою інших засобів, передаються у ведення нових серверів, але частіше створення нового сервера протоколу веде за собою появу принципово нових видів ресурсів – ресурсів, які раніше ніяк не були представлені в Internet, а можливо, і взагалі не існували.

Internet влаштований так, що той самий вузол мережі може одночасно працювати по декількох протоколах, для яких на цьому вузлі встановлені програми-сервери. Тому великі вузли Internet зараз мають повний набір серверів, і до них можна звертатися майже по кожному з існуючих протоколів: від самих древніх і примітивних до сучасних. Більше того, багато старих протоколів продовжують активно використовуватися й зараз – адже ті ресурси, для доступу до яких вони призначені, як і раніше потрібні людям.

Ситуація, щоправда, трохи полегшується тим, що найсучасніший із цих мережних інструментів – World Wide Web (WWW) – поєднує під собою майже

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

всі інші. Це значить, що користувач програми-клієнта (**браузера**) WWW може працювати з будь-якими серверами, тому що його браузер, як складова системи перегляду хмарних сервісів уміє спілкуватися з кожним типом сервера на їхній власній мові, тобто з використанням відповідного протоколу. Інформацію, отриману від будь-якого сервера, браузер, як складова системи перегляду хмарних сервісів WWW виводить на екран у стандартній і зручній для сприйняття формі, так що перемикання з одного протоколу на інший для користувача часто залишаються просто непоміченими.

І, проте, навіть якщо ви працюєте із браузером WWW (а зараз важко знайти користувача Internet, що волів би щось інше), корисно вміти розбиратися в принципах роботи й можливостях існуючих серверів і протоколів.

1.2 Область застосування

Браузер, як складова системи перегляду хмарних сервісів – це програма для перегляду ресурсів мережі Internet. Браузер, по своїй природі, є програмою – клієнтом. Строго говорячи, браузер – це WWW-клієнт, але WWW-клієнт не обов'язково повинен бути браузером – це може бути наприклад, програма – робот, що подорожує по павутині Web. Якщо бути більше точним, браузер – це клієнт WWW, призначений для інтерактивної роботи з користувачем (а іноді виконуюча й деяка стороння функції, наприклад, обробку електронної пошти). Він дозволяє одержати доступ до всіх інформаційних ресурсів мережі. За допомогою браузера можна переглядати й завантажувати картинки, звукові й відео файли, різну текстову інформацію (електронні книги, новини, журнали) і т.і.

Існує кілька видів браузерів:

– Браузер режиму командного рядка. До цього типу відносяться самі ранні браузери. Вони не дають можливості переглядати текст і графіку. Такі браузери підтримують переміщення тільки з використанням цифрових адрес (IP). У цей час практично не використовуються, тому докладно розглядати їх я не буду.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

– Повноекраний браузер. Текстовий браузер без підтримки мультимедійних (картинки, анімація й т.п.) ресурсів мережі Інтернет. За допомогою його можна переглядати тільки текст і посилання. Так як більшість користувачів звиклася подорожувати по Інтернету за допомогою браузерів підтримуючих мультимедіа, не варто остаточно забувати й ті, які відображають тільки текст. Звичайно, браузери такого виду використовуються досить рідко, але швидкість завантаження сторінок у них вражає. Без графічних і оформлювальних елементів, а також без таблиць, багато сторінок завантажуються практично миттєво. Один із самих популярних повноекраних браузерів є Lynx, що входить до складу операційної системи Lynix.

– Браузер з підтримкою мультимедіа. Найпоширеніші й популярні браузери сьогодні. Дозволяють працювати практично з усіма видами інформації, представленої в Інтернеті. 99% користувачів глобальної мережі Інтернет інтенсивно й щоденно використовують можливості цих браузерів. Найбільше часто використовувані: Microsoft Edge, Opera, Mozilla, Netscape Navigator.

– Існують ще так звані браузери-доповнення. Вони є надбудовами над повнофункціональними браузерами. Найчастіше розроблювачами доповнень використовується Microsoft Edge. Надбудови використовують для відображення сайтів “движок” цього браузера. Тому їхньої можливості в цій області повністю ідентичні з Microsoft Edge. Доповнення всього лише змінюють інтерфейс і додають деякі функції, які розроблювачі з Microsoft обійшли своєю увагою.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи перегляду хмарних сервісів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

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

Історія розвитку

Першим розповсюдженим браузером із графічним інтерфейсом був NCSA Mosaic. Вихідний код цього одного з перших браузерів був відкритий і деякі інші браузери (Netscape Navigator і Microsoft Edge) взяли його за основу. Цей браузер, як складова системи перегляду хмарних сервісів мав свої недоліки, але майже всі вони були усунуті в браузері Netscape Navigator (деякі співробітники компанії Netscape були з NCSA і брали участь у розробці Mosaic). Netscape випустила Netscape Navigator під різні операційні системи (UNIX, Windows, Mac OS) і домоглася помітного успіху, у тому числі й комерційного. Це спонукало компанію Microsoft випустити свій браузер, як складова системи перегляду хмарних сервісів Microsoft Edge.

На відміну від Netscape, Microsoft відразу випускала локалізовані версії Microsoft Edge. В 1995 році Microsoft випустила операційну систему Windows 95, у ній не було вбудованого браузера, але через якийсь час у відновлення системи (Windows 95 OSR2) браузер, як складова системи перегляду хмарних сервісів (Microsoft Edge 3.0) був убудований. За «агресивну позицію в завоюванні нового ринку й нечесні методи боротьби» Microsoft була оштрафована; за Windows 95 Microsoft брала із продавців комп'ютерів 100 доларів, а за версію Windows 95 з Microsoft Edge – 97 доларів, у такий спосіб просуваючи Microsoft Edge на ринок браузерів. Компанія також відмовлялася поставляти браузер, як складова системи перегляду хмарних сервісів окремо від ОС, за що теж була оштрафована; крім того, назва браузера й позначення ярликів дає привід думати, що мережа Інтернет

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

і браузер, як складова системи перегляду хмарних сервісів Microsoft Microsoft Edge – це те саме. До того ж, Microsoft додавала у свій браузер, як складова системи перегляду хмарних сервісів несумісні зі стандартами розширення мови HTML, і це можна вважати початком війни браузерів, що закінчилася монополізацією (більше 95 %) ринку браузером від Microsoft. Версії Microsoft Edge 1.0-3.0 у порівнянні з Netscape Navigator були набагато гірше і тільки до версії 4.0 по зручності використання вони зрівнялися.

Через втрату ринку доходи компанії Netscape упали й її придбала AOL, а вихідний код браузера Netscape був випущений під вільною ліцензією MPL (Mozilla Public License). Назва «Mozilla» було присутнє в браузері від Netscape і означало скорочення слів Mosaic+killer. Однак цей код було вирішено не використовувати й замість нього для Netscape 6 з нуля був написаний новий движок (Gecko), що орієнтувався на повну підтримку стандартів, на основі якого пізніше були створені вхідні в комплект Mozilla Suite браузер, як складова системи перегляду хмарних сервісів, поштовий і ірс-клієнт і редактор веб-сторінок.

У наслідку в Mozilla Foundation було ухвалене рішення поставляти й розвивати браузер, як складова системи перегляду хмарних сервісів окремо від загального пакета й народився проект Mozilla Firefox. Firefox містить багато можливостей, відсутніх в Microsoft Edge або інших браузерах, і поступово набирає популярність (на 2008 рік – близько 20, а в деяких країнах – і до 40 % ринку браузерів).

Монополізація ринку браузером Microsoft Edge мала й інші наслідки – Microsoft майже перестала розвивати браузер, як складова системи перегляду хмарних сервісів і він з 4-ї до 6-ї версії майже не мінявся: гірше інших браузерів підтримував стандарти, відставав по зручності використання й по швидкості роботи й відображення сторінок. Таке положення сподвигло Microsoft знову зайнятися браузером і сьома версія вийшла з деякими змінами (були додані закладки-таби, поліпшена підтримка стандартів, піднята швидкість роботи).

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Однак, на сучасний момент швидкість проходження тестів javascript (основа web 2.0) однаково майже на порядок гірше, ніж в інших браузерів. Головна відмінність Microsoft Edge від своїх попередників, у тому, що за заявою розроблювачів він буде за замовчуванням підтримувати всі сучасні веб-стандарти. Нова версія Microsoft Edge буде успішно проходити Acid2 Browser Test, розроблений Web Standards Project. Цей тест дозволяє довідатися, чи правильно програма обробляє останні специфікації HTML і CSS і чи здатна коректно відображати веб-сторінки.

Нові можливості Microsoft Edge:

– Прискорювачі. Прискорювачі дозволяють швидше виконувати повсякденні дії в Інтернеті й навіть знаходити нові служби. Зіставлення, переклад, відправлення електронної пошти й багато чого іншого тепер доступно за допомогою декількох клацань мишею.

– Приватний режим. За допомогою приватного режиму перегляду в Microsoft Edge можна переглядати веб-сторінки без збереження в журналі. Тепер при здійсненні покупок в Інтернеті можна бути впевненим, що члени родини випадково не довідаються про спеціально приготовлений подарунок, а також використовувати комп'ютер із загальним доступом без збереження адрес веб-сторінок, що зберігаються у журналі перегляду.

– Веб-фрагменти. Будьте в курсі змін найбільш важливих для вас веб-вузлів. Після додавання веб-фрагмента більше не прийде постійно вертатися на той самий веб-вузол, щоб довідатися останні новини, біржові котирування, відновлення інтернет-аукціонів, прогноз погоди й навіть результати спортивних змагань.

– Варіанти пошуку. Більш ефективний пошук з докладними рекомендаціями із кращих служб пошуку й журналу. Переглянете мініатюри попереднього перегляду й теми вмісту, що рекомендується, які відображаються в міру уведення в поле миттєвого пошуку.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– Фільтр SmartScreen. Нові функції безпеки призначені для захисту від помилкових і шкідливих веб-вузлів, які можуть поставити під погрозу інформацію, конфіденційність і особисті дані.

В 1996 році з'явився браузер, як складова системи перегляду хмарних сервісів Opera. Довгий час цей браузер, як складова системи перегляду хмарних сервісів позиціонувався як найшвидший браузер, як складова системи перегляду хмарних сервісів. Він дійсно був самим легким й у старті, і в завантаженні з Інтернету, і у відображення сторінок, і залежно від ведучого в той момент на ринку браузера була сполучим з ним (або намагався бути таким). Спочатку Opera поширювалася як умовно-безкоштовне ПЗ (shareware), але в 2005 році Opera також стала поширюватися безкоштовно.

Війна браузерів була б винятково комерційною справою корпорацій, якби основним прийомом у боротьбі не стало додавання специфічних, нестандартних можливостей до браузерів. Найбільші розходження виникали в підтримці JavaScript – мови сценаріїв, що надає інтерактивність документам. У результаті багато сайтів «оптимізувалися» під одні браузери і погано працювали в інших браузерах.

В Microsoft Edge Microsoft спочатку пропонувала ввести HTML-коментар, який би вказував браузерам, які версії движків використовувати (заголовок DOCTYPE при цьому повинен був бути відмінений, як маловикористовуємий), що обурило прихильників стандартів, тому що вказівки на використання стандартів не передбачалося. Пізніше було оголошено, що IE8 буде за замовчуванням використовувати «режим стандартів», а не «режим сумісності», а при виявленні відповідного коментарю використовувати режим імітації попередньої версії.

У локалізованих версіях операційних систем сімейства Microsoft Windows до 2021 року браузери йменувалися *оглядачами, оглядачами мережі або веб-оглядачами*.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

З огляду на зменшення частки Microsoft Edge на ринку браузерів (уперше за кілька останніх років частка Майкрософт упала нижче 90 відсотків), компанія Google вирішила «допомогти» конкуренції на ринку браузерів і випустила свій браузер, як складова системи перегляду хмарних сервісів з відкритим вихідним кодом – Chrome. Браузер, як складова системи перегляду хмарних сервісів Chrome 1.0 випереджає по швидкості роботи javascript всі відомі браузери, містить деякі нововведення (ізоляція збоїв, режим «інкогніто» й т.і.). Поки випущена версія тільки для Windows 10/11, для інших ОС можлива тільки складання з вихідних текстів, однак готові варіанти браузера для інших ОС компанія обіцяє випустити пізніше.

Розповсюджені браузери

У таблиці 2.1 приведемо дані по поширеності того або іншого браузера.

За даними компанії Net Applications у травні 2021 року ринкова частка браузерів складала:

- Google Chrome – 71,27 %;
- Firefox —,97 %;
- Safari – 6,57 %;
- Opera – 0,75 %;
- Netscape – 0,45 %;
- Mozilla Suite – 0,09 %;
- Opera Mini – 0,06 %;
- PlayStation – 0,03 %;
- Microsoft Pocket Microsoft Edge – 0,02 %;
- Blazer – 0,02 %;
- Konqueror – 0,01 %;
- ACCESS NetFront – 0,01 %;
- Danger Web Browser, BlackBerry, ANT Galio, Lotus Notes, iCab, WebTV – близько 0 %.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Таблиця 2.1 – Поширеність браузерів

Популярні	Microsoft Edge, Firefox, Safari, Opera, Flock, Maxthon
Менш розповсюджені	SeaMonkey, Avant Browser, Netscape Navigator, Google Chrome, Galeon, Epiphany, Kazehakase, Charon, Arachne, Konqueror, K-Meleon, slimbrowser, FastIE, MyBrowser, Dillo
Текстові	Alynx, ELinks, Links, Lynx, Netrik, w3m, WebbIE, DosLynx
Для портативних пристроїв	Microsoft Edge Mobile, Opera Mini, Safari для Apple iPhone

Нижче ми приведемо в табличному виді основні порівняльні дані й технічні характеристики веб-браузерів, при цьому ці дані класифікуємо по наступних ознаках:

- основна інформація;
- підтримка операційних систем;
- функціональність;
- підтримка веб-технологій і протоколів;
- підтримка форматів зображень;
- частка браузерів на ринку.

Основна інформація

Основна інформація про браузери: виробник/компанія, ліцензія/вартість і т.п. представлена в таблиці 2.2.

«Б» – Безкоштовно.

«В» – Бласницька.

Підтримка операційних систем

У таблицях наведених нижче використовуються наступні скорочення:

«+» – так, підтримує; «-» – ні, не підтримує; «Ч» – частково підтримує; «П» – плагин; «З» – припинена підтримка; «?» – невідомо; «Н/Д» – інформація не надана; «Р» – розширення; «Н/З» – Безглуздо; «І» – Інтегруємо

Таблиця 2.2– Основна інформація про браузері

Назва	Виробник	Дата першої публічної версії	Останній реліз	Вартість (дол.)	Ліцензія	Поточне ядро
Амауа	W3C, INRIA	11/96	10.0	Б	W3C	Thot
Camino	Mozilla Foundation	02/02	1.5.4	Б	MPL, потрійна ліцензія MPL/GPL/ LGPL	Gecko
Dillo	Arellano Cid, Geerken, Rota і ін.	12/99	0.8.6	Б	GPL	gzilla
Elinks (текстовий)	Baudis, Fonseca і ін.	12/01	0.11.3	Б	GPL	?
Eriphany	GNOME	12/02	2.20.1	Б	GPL	Gecko
Galeon	GNOME	06/00	2.0.3	Б	GPL	Gecko
ICab	iCab_Company	1998	3.0.3	30	В	ICab
Microsoft Edge	Microsoft Spyglass, Inc.	08/95	7.0 5.2.3 (Mac)	Частина Windows Б (Mac)	В	Trident (Win) Tasman (Mac)
K-Meleon	Doozan, Erikson, Vallet і ін.	11/00	1.1.2	Б	GPL	Gecko

Продовження таблиці 2.2

Назва	Виробник	Дата першої публічної версії	Останній реліз	Вартість (дол.)	Ліцензія	Поточне ядро
Konqueror	KDE	10/00	3.5.8	Б	GPL	KHTML
Links (текстовий)	Ратоска і ін.	11/99	0.99	Б	GPL	?
Lynx (текстовий)	Montulli, Grobe, Rezac і ін.	07/93	2.8.6	Б	GPL	?
Mosaic	Marc Andreessen і Eric Bina, NCSA	04/93	2.6	Б для некомерційного використання	В	?
Mozilla Suite	Mozilla Foundation	12/98	1.7.13	Б	MPL	Gecko
Mozilla Firefox	Mozilla Foundation	09/02	3.0.5	Б	MPL/GNU LGPL/GNU GPL	Gecko
e-Capsule Private Browser	Eisst	2005	2.1.0.6 11	19,95	В	Gecko
Netscape Navigator	Netscape Communications, Mozilla Foundation (з 2000), Mercurial Communications (з 2004)	10/94	9.0.0.6	Б	NPL	Gecko

Таблиця 2.3 – Операційні системи під якими браузери здатні працювати без емуляції

	Windows	Mac OS X	GNU/Linux	BSD	Unix	Інші
Amaya	+	+	+	+	+	?
Camino	-	+	-	-	-	?
Dillo	+	+	+	+	+	?
ELinks	+	+	+	+	+	?
Epiphany	-	+	+	+	+	?
Galeon	-	+	+	+	+	?
iCab	-	+	-	-	-	?
Microsoft Edge	+	3	-	-	3	-
Links (текстовий)	+	+	+	+	+	?
Lynx (текстовий)	+	+	+	+	+	?
K-Meleon	+	-	-	-	-	-
Konqueror	+	+	+	+	+	?
Mosaic	+	+	+	+	+	AmigaOS, AROS, ін.
Maxthon	+	-	-	-	-	-
Mozilla Firefox	+	+	+	+	+	Solaris, OS/2, BeOS, ?
Netscape Navigator	+	+	+	+	+	?
Mozilla	+	+	+	+	+	?
OmniWeb	-	+	-	-	-	?

Продовження таблиці 2.3

	Windows	Mac OS X	GNU/Linux	BSD	Unix	Інші
Opera	+	+	+	+	+	BeOS, Solaris, QNX, wii, OS/2, а також ОС мобільних пристроїв: Windows Mobile, S60, UIQ
e-Capsule Private Browser	+	-	-	-	-	-
Safari	+	+	-	-	-	?
SeaMonkey	+	+	+	+	+	OS/2
WorldWideWeb	-	-	-	-	-	тільки NeXTSTEP

Функціональність

Нижче представлена інформація про розповсюджені функціональні можливості браузерів. Зовнішні посилання ведуть на розширення, які додають зазначену функціональність браузеру.

Данні по функціональності наведені у таблиці 2.4.

Підтримка веб-технологій і протоколів

Інформація про веб-стандарти, технології й протоколи, які підтримують браузери. Зовнішні посилання ведуть до інформації про можливість підтримки їх у майбутній версії браузера.

Дані по підтримці веб-технологій наведені у таблиці 2.5.

Підтримка форматів зображень

Інформація про всі формати зображень які підтримує той або інший браузер, як складова системи перегляду хмарних сервісів. Зовнішні посилання ведуть на інформацію про їхню підтримку в майбутніх версіях браузера.

Дані по підтримці форматів зображень наведені у таблиці 2.6.

Таблиця 2.4 – Розповсюджені функціональні можливості браузерів

	Вкладки	Менеджер завантажень	Панель пошуку	Перевірка орфографії	Блокування баннерів	Блокування поп-апів	Фільтр фішіngu	Пропорційне збільшення
Amaya	+	?	?	+	?	Н/Д	-	+
Camino	+	+	+	-	-	+	-	-
Dillo	-	-	Ч	-	-	-	-	-
Epiphany	+	+	+	-	Р	+	-	+
Galeon	+	+	+	?	?	+	-	?
Microsoft Edge (з 7.0)	+	+	+	Ч	+	+	+	+
K-Meleon	+	?	?	?	?	+	-	?
Konqueror	+	+	+	+	+	+	-	+
Links	-	-	-	-	-	Н/З	-	Н/З
Lynx	-	-	-	-	-	Н/З	-	Н/З
Maxthon (колишній MyIE2)	+	+	+	-	+	+	-	+
Mosaic	-	?	-	?	-	Н/Д	-	?
Mozilla	+	+	+	-	-	+	-	?
Mozilla Firefox	+	+	+	+	+	+	+	+
Netscape Navigator	+	+	+	-	-	+	-	?
OmniWeb	+	+	+	+	+	+	-	-
Opera	+	+	+	Ч	+	+	+	+
e-Capsule Private Browser	+	+	+	-	+	+	+	+
Safari	+	+	+	+	+	+	+	+
SeaMonkey	+	+	+	+	-	+	-	У версії 2.0
WorldWide Web	-	?	-	?	-	Н/Д	-	?

Таблиця 2.5 – Підтримка веб-технологій і протоколів

	Веб-стандарти й технології								Інтернет протоколи			
	CSS2	Фрейми	Java	JavaScript	XHTML	RSS	Atom	Email	FTP	NNTP (Usenet)	SSL	IRC
Amaya	Ч	-	-	-	+	-	-	-	-	-	-	-
Camino	Ч	+	+	+	+	-	-	-	+	-	+	-
Dillo	Ч	-	-	+	-	-	-	-	+	-	Ч	-
Epiphany	Ч	+	+	+	+	Ч	Ч	-	+	-	+	-
Galeon	Ч	+	+	+	+	-	-	-	+	-	+	-
Microsoft Edge	Ч	+	+	+	-	+	+	-	+	-	+	-
K-Meleon	Ч	+	+	+	+	+	+	-	+	-	+	-
Konqueror	Ч	+	+	+	+	+	+	-	+	-	+	-
Links	-	+	-	Ч	-	-	-	-	+	-	+	-
Lynx	-	-	-	-	-	-	-	-	+	+	+	-
Maxthon	Ч	+	+	+	-	+	+	-	+	-	+	-
Mosaic	-	-	-	-	-	-	-	-	+	-	-	-
Mozilla	Ч	+	+	+	+	+	+	+	+	+	+	+
Mozilla Firefox	Ч	+	+	+	+	+	+	I	Ч + (II)	II	+	II
Netscape Navigator	Ч	+	+	+	+	-	-	+	+	+	+	-
OmniWeb	Ч	+	+	+	-	+	+	-	+	-	+	-
Opera	Ч	+	+	+	+	+	+	+	+	+	+	+
e-Capsule Private Browser	Ч	+	+	+	+	-	+	-	+	-	+	-
Safari	Ч	+	+	+	+	+	+	-	Ч	-	+	-
SeaMonkey	Ч	+	+	+	+	+	+	+	+	+	+	+
WorldWideWeb	-	-	-	-	-	-	-	-	+	+	-	-

Таблиця 2.6 – Підтримка форматів зображень

	JPEG	GIF	PNG	MNG	SVG
Amaya	+	+	+	–	+
Camino	+	+	+	–	–
Dillo	+	+	+	–	–
Epiphanу	+	+	+	–	–
Galeon	+	+	+	–	–
Microsoft Edge	+	+	+	–	–
K-Meleon	+	+	+	–	–
Konqueror	+	+	+	+	П
Links (текстовий)	–	–	–	–	–
Lynx (текстовий)	–	–	–	–	–
Mosaic	+	+	?	–	–
Mozilla	+	+	+	–	+
Mozilla Firefox	+	+	+	–	+
Netscape Navigator	+	+	+	+	–
OmniWeb	+	+	+	–	–
Opera	+	+	+	+	+
e-Capsule Private Browser	+	+	+	–	–
Safari	+	+	+	–	Неповністю
SeaMonkey	+	+	+	–	+ (з версії 1.1+)
WorldWideWeb	+	+	?	–	–

Частка браузерів на ринку

Таблиця 2.7 – Частка браузерів на ринку

Браузер, як складова системи перегляду хмарних сервісів	Частка на ринку
Microsoft Microsoft Edge	73.75%
Firefox	18.41%
Safari	6.25%
Opera	0.71%
Netscape	0.62%
Mozilla	0.08%
Opera Mini	0.05%
Playstation	0.03%
Microsoft Pocket Microsoft Edge	0.02%
Konqueror	0.02%
Blazer	0.02%
WebTV	0.01%
ACCESS NetFront	0.01%
Danger Web Browser	0.00%
ANT Galio	0.00%
Lotus Notes	0.00%
iCab	0.00%
BlackBerry	0.00%

- 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 стосуються, застосунків на VCL і FireMonkey
- RAD Studio 10.4 Короткий огляд:
- Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

– 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.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

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

Єдине керування пам'яттю

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

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

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

Розширена підтримка бібліотек C++

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

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

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими 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.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

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 масштабуються 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 компонентах 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 використовують стиль.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
- 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.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

2.3 Розгорнута постановка завдання

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

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Вимоги висунуті до браузера

Базова ідея Web-технологій полягає в тім, що будь-який користувач будь-якого браузера може підключитися до будь-якому Web-додатку. Однак на перевірку ця потенційно приваблива ідея є фікцією. Універсальний доступ до інформації став можливим лише завдяки титанічним зусиллям розроблювачів додатків для Web, яким довелося пройти дуже складний шлях, перш ніж їхні програми «навчилися» розпізнавати різні версії браузерів і динамічно адаптувати інтерфейс до особливостей цих браузерів. Це кропітка й стомлююча робота, яку доводиться проробляти щораз після появи нового браузера або чергової його версії. Але ж всі ці зусилля можуть виявитися даремними, коли повсюдне поширення загальноприйнятих стандартів зведе їх нанівець.

Відповідальність за твердження стандартів, що стосуються браузерів, покладена на Консорціум World Wide Web і Європейську асоціацію виробників комп'ютерної техніки (European Computer Manufacturer's Association – ECMA). Незважаючи на те що представники компаній, що займаються розробкою браузерів, беруть участь у діяльності комісій, вони не можуть скільки-небудь істотно вплинути на прийняття остаточних рішень. Адже тут перетинаються інтереси самих різних груп. Розроблювачам браузерів хотілося б робити враження затятих апологетів стандартів – і в той же час мати цілком лояльних клієнтів. Microsoft і Netscape протягом тривалого часу вбудовували у свої продукти всі нові й нові специфічні можливості, намагаючись у такий спосіб залучити розроблювачів. Користувачі ж віддають перевагу найбільш зручному для себе браузеру й вимагають, щоб додатка були сумісні з обраним ними варіантом. Природно, все це жадає від розроблювачів додаткових зусиль, а код

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Web-додатків непомірно роздувається, оскільки в ньому повинні враховуватися всі розходження між представленими на ринку браузерами.

З випуском кожної нової версії виробники пропонують своє оригінальне трактування стандартів. Якби всі браузери дійсно відповідали стандартам, розроблювачеві досить було б написати всього один інтерфейс свого Web-додатка, і цей інтерфейс виглядав би абсолютно однаково в середовищі будь-якого браузера. Захвати, що супроводжували появу кожної нової версії, через деякий час перемінялися розчаруванням, оскільки користувачі починали зауважувати, що прірва, яка розділяє продукти різних виробників, стає усе ширше.

На щастя, часи міняються. Зокрема, останні версії браузерів компаній Microsoft, Mozilla і Opera Software повністю відповідають вимогам існуючих стандартів. Те ж саме можна сказати й про Konqueror – складову частину відомого середовища KDE (K Desktop Environment). Звичайно, повністю гарантувати те, що всі браузери будуть точно відповідати специфікаціям стандартів, ще не можна, але розроблювачі вже переходять до створення Web-додатків, орієнтованих на стандарти, а не на специфічні характеристики конкретних браузерів.

Призначення стандартів

Стандарти браузерів охоплюють два основних аспекти Web-технологій у частині інтерфейсу: його зовнішній вигляд і функціональна реалізація тих або інших можливостей. До стандартів зовнішнього подання екранного інтерфейсу відносяться затверджені W3C специфікації HTML 5 і CSS (Cascading Style Sheets) рівня 3. Мова HTML визначає базову структуру й форматування Web-документа, а каскадні таблиці стилів дозволяють додати до неї точний контроль за атрибутами екранного інтерфейсу: шрифтами, квітами й схемами розташування. Стандарти функціональної реалізації ECMA-262 (або ECMAScript) і запропонована W3C об'єктна модель документа CSS DOM (Document Object

					ВКРМ-122.22.0004.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		33

Model) описують клієнтські сценарії і їхню можливість динамічно змінювати інформаційне наповнення Web-сторінки.

Специфікації HTML 5, затверджені W3C в 1999 році, описують лише децицію базових конструкцій, що лежать в основі Web-додатків. Однак незважаючи на розходження, що існують між браузерами різних виробників, всі вони підтримують HTML – якщо й не версії 4.01, то 4.0. У користувача, що має у своєму розпорядженні останню редакцію браузера Opera, Firefox, Microsoft Edge або Konqueror, не повинне виникати ніяких труднощів з переглядом HTML-сторінок. Їхній код не містить специфічних розширень типу керуючих елементів Active або модулів Netscape. Будь-який розроблювач, що бажає встановити, чи відповідає його Web-сторінка вимогам стандартів, має можливість перевірити це за допомогою служби HTML Validator на сайті www.w3c.org. Цей програмний інструментарій ідентифікує всі нестандартні конструкції HTML, які можуть бути оброблені браузерами неправильно.

Формування стилів

Стандарт W3C CSS1 (CSS Level 1) з'явився ще в 1996 році, але виробники браузерів не поспішали підтримувати його у своїх продуктах. У браузерах Netscape до появи шостої версії було дуже багато помилок у підтримці цього стандарту, розроблювачі не забезпечили повної сумісності з CSS1. Корпорація Microsoft реалізувала функції CSS1 тільки у версії Microsoft Edge 6.0. Сьогодні обоє розроблювача повністю підтримують стандарт CSS1. Недавно до браузерів шостого покоління Microsoft і Netscape приєдналися продукти Opera 5 і Konqueror 2.2, розроблювачі яких також заявили про повну сумісність їхнього програмного забезпечення з CSS1. Через п'ять років після твердження специфікацій стандарту нарешті з'явилася повсюдна підтримка CSS1.

Але що це дає розроблювачам Web-додатків? Кожному елементу HTML-коду, у тому числі й відображуваним на екрані (наприклад, заголовкам і гіперпосиланням), ставиться у відповідність певний стиль CSS1. Стандарт CSS1 і модель DOM дозволяють розроблювачам динамічно змінювати властивості

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

поточної сторінки. Підтримка стандарту CSS1 всіма виробниками браузерів стала великою перемогою розроблювачів додатків для Web.

В 1998 році W3C затвердив специфікації стандарту CSS2 (CSS Level 2), у якому підтримка таблиць стилів була поліпшена за рахунок функцій розміщення текстових і графічних елементів у довільному місці сторінки. На щастя, всі виробники браузерів уже оголосили про сумісність майбутніх версій своїх продуктів з CSS2.

Підтримка сценаріїв

Мова Netscape JavaScript, доповнена технологією Microsoft JScript, перетворилася згодом у стандарт ECMA-262. У грудні 1999 року ECMA затвердила третю редакцію стандарту ECMA-262, що розроблялася під найменуванням ECMAScript. Сьогодні всі популярні браузери підтримують ECMAScript, а Microsoft і Netscape оголосили про повну сумісність своїх продуктів із третьою редакцією ECMAScript. Проте розроблювачі як і раніше продовжують використовувати клієнтський код, у якому присутні нестандартні функції JavaScript і JScript. Виробникам варто ретельно вивчити свої клієнтські сценарії й внести зміни в ті з них, де не виправдано використовуються специфічні можливості браузера. Конструкції JavaScript і JScript можна трансформувати в код ECMAScript при мінімальних зусиллях.

Додати до браузера базовий рівень підтримки сценаріїв досить просто. Організувати керування зв'язками між кодом сценаріїв і документом Web складніше, тут браузери найбільшою мірою відрізняються один від іншого. Всі популярні браузери підтримують сьогодні сукупність стандартів DOM1 (DOM Level 1) і DOM2 (DOM Level 2), надаючи за допомогою сценаріїв простий доступ до елементів HTML і властивостям таблиць CSS. В останніх версіях Microsoft Edge, Navigator і Konqueror забезпечена повна підтримка DOM1. А от Opera має тут дуже серйозні обмеження, особливо це стосується можливості додавання й видалення елементів HTML на відображуваній сторінці.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Гарантії безпеки

Відповідність Web-браузерів вимогам стандартів залишає поки бажати кращого, однак уже сьогодні розроблювачі додатків для Web можуть використовувати набір базових функцій, що буде підтримуватися всіма популярними браузерами. Здається, що першим кроком у будь-якому новому проекті створення додатків для Web (а також у будь-якому проекті, ціль якого полягає в адаптації вже існуючого коду до особливостей стандартів) повинне стати надання клієнтам гарантій технічної підтримки й відновлення ПЗ в майбутньому. Практика показує, що не можна змусити користувачів Navigator перейти на Microsoft Edge, але при цьому більшість охоче встановлюють у себе останню версію улюбленого браузера.

Попередньо необхідно все вивчити й протестувати, щоб виявити області, у яких функції часткової підтримки браузерами специфікацій DOM2 і CSS2 перетинаються.

В ідеалі всі ці стандарти повинні позбавити нас від додаткових зусиль по розпізнаванню браузерів і адаптації до їхніх особливостей. На практиці ж завжди залишаються користувачі, у яких будуть установлені застарілі версії або не настільки широко розповсюджені програми браузерів. Якщо виявиться, що клієнтський браузер, як складова системи перегляду хмарних сервісів не підтримує якісь функції вашого додатка, переадресуйте його на більше простий сайт і повідомите про переваги, які принесе установка оновленого варіанта браузера.

GUI інтерфейс браузера

У сучасному браузері повинні бути реалізовані наступні елементи інтерфейсу GUI:

- Адресний рядок.
- Автопошук.
- Вибір протоколу за замовчуванням в адресному рядку.
- Механізм автопідстановки адреси сайту.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

– Префікс і суфікс за замовчуванням. За замовчуванням стандартним префіксом є префікс www. а стандартним суфіксом – .com.

– Механізм автозаповнення – браузер, як складова системи перегляду хмарних сервісів перевіряє рядок, що ви вводите, з рядками з галузі реєстру, і якщо знаходить збіги, тоді пропонує вам вибрати їх у списку, що розкривається.

– **Панель пошуку.**

– Зміна пошукової машини за замовчуванням.

– Меню виправлення.

– Меню Вид.

– Меню Вибране.

– Меню Сервіс.

– Меню Довідка.

– Спливаючі панелі.

– Вкладка Додатково.

– **Кнопка режиму передперегляду.**

– Панель команд.

– **Контекстне меню браузера.**

– **Зміна масштабу.**

Стандартні функції браузера

До стандартних функцій сучасного браузера відносяться наступні:

– перегляд із вкладками;

– здатність зберігати множина вкладок під однією закладкою;

– download менеджер;

– блокувальник спливаючих вікон;

– фішинг-фільтр;

– вбудований RSS-агрегатор;

– підтримку інтернаціональних доменних імен;

– підтримку засобів групової політики;

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

- підтримку HTML 5;
- підтримку CSS Level 1;
- підтримку XML 1.0;
- підтримку DOM Level 1 і частково CSS Level 2 і DOM Level 2;
- підтримку підключення розширень, що реалізується через об'єктну модель компонентів (COM);
- hotclick;
- переклад тексту на іншу мову;
- перевірка орфографії;
- розпізнавання мишачих жестів;
- функцію блокування рекламних баннерів і флеш-роликів;
- меню швидкого перемикачання прокси-серверів зі списку;
- опцію коректування змінної User Agent String, відповідальної за ідентифікацію браузера веб-серверами;
- конвертор сторінок у графічні зображення форматів JPG, BMP, GIF, PNG або TIFF;
- механізм підключення користувальницьких сценаріїв для маніпулювання HTML-контентом на стороні браузера;
- функцію відновлення всіх відкритих вкладок, що збереглися з попередньої робочої сесії браузера;
- опцію Super Drag&Drop, що дозволяє банальним перетаскуванням об'єктів на веб-сторінці виконувати звичні речі: робити швидкий пошук виділеного тексту в заданому за замовчуванням пошуковому сервісі, ефектно зберігати зображення й документи, а також проробляти інші речі спритним рухом мишки;
- кілька готових надбудов і скриптів для скачивання відеороликів з Google Video, Youtube і інших онлайн-сервісів.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Панелі

Функція, пов'язана з переглядом електронної пошти й іншого. Крім того, користувач може завантажити додаткові панелі або створити свою власну.

- Контакти – служить як адресна книга;
- Історія – надає журнал з усіма сторінками, які були відвідані, в хронологічному порядку, починаючи із самої останньої;
- Посилання – перераховані всі посилання, які користувач використовував у період перегляду сайтів;
- Примітки – дозволяє користувачеві копіювати й вставляти вміст зі сторінок в убудований текстовий редактор для наступного перегляду й редагування;
- Інформація – показує інформацію про сторінку, у тому числі її тип MIME, розмір і кодування;
- Вікна – приводиться коротка інформація із всіх вкладок і вікон, відкритим у цей момент.

Перспективи розвитку функцій браузера

До перспективних функцій браузера за результатами дослідження OpenAjax Alliance відносяться наступні:

- Більш повна й послідовна підтримка 2D і векторної графіки, спрямована на забезпечення платформної й вендору-незалежності.
- Скасування обмеження у два одночасних підключення до сервера. Воно проявляється, якщо браузер, як складова системи перегляду хмарних сервісів запитує більше 2 сторінок одночасно й серйозно ускладнює роботу Ajax web 2.0 додатків. Зараз браузери, що встановили більше 2 сесій спеціально блокують наступні з'єднання до завершення попередніх.
- Швидкодія HTML DOM операцій. У частині споживання ресурсів, це самі не економічні із всіх JavaScript операцій, що вириваються «уперед» на тисячі відсотків. І тому що вони є одними з найважливіших, необхідно починати роботу з їхньої оптимізації.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Опис розроблюємого Інтернет–браузера

При написанні різних браузерів спостерігається таке явище: основна кількість браузерів роблять на якихось певних движках.

Движок – виділена частина програмного коду для реалізації конкретної прикладної задачі – програма, частина програми, комплекс програм або бібліотека, залежно від задачі й реалізації. Як правило, прикладна частина виділяється із програми для використання в декількох проектах і/або роздільної розробки/тестування.

Використання готового движка при розробці програми, сайту або іншого продукту скорочує час розробки, дозволяє приділити більше часу розробці інших підсистем, наприклад користувальницькому інтерфейсу (або інформаційної наповненості сайту).

Разом з тим продукти, зроблені з використанням движків, успадковують їхні помилки й проблеми безпеки.

Найбільше браузерів зроблено на движку Microsoft Edge, це й не дивно, тому що браузер, як складова системи перегляду хмарних сервісів древній як динозавр, і ще не швидко він вимре, тому що його розвитком і підтримкою займається Майкрософт. На розробку браузерного движку потрібні чималі ресурси, тому програмістам легше робити різні оригінальні примочки й доповнення до движка, іменованому Microsoft Edge, і називати цей браузер, як складова системи перегляду хмарних сервісів уже своїм, оригінальним ім'ям, хоча це все той же Microsoft Edge тільки модифікований як у кращу, так і в гіршу (що буває досить часто) сторону.

У магістерському проєкті за основу браузера взято движок Trident. Trident (також відомий як MSHTML) – назва браузерного движка для Windows–версії Microsoft Microsoft Edge. Уперше Trident був реалізований у четвертій версії Microsoft Edge, і з тих пор піддавався постійним поліпшенням і переробкам.

Trident був розроблений як програмний компонент, що дозволяв розроблювачам програмного забезпечення легко додавати можливість перегляду

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

веб-сторінок у їхні власні додатки. Він використовує технологію COM (компонентну модель об'єктів) для перегляду й виправлення веб-сторінок у будь-якому оточенні, що підтримує інтерфейс цієї моделі – наприклад, в Delphi, C++ або .NET. Відповідний елемент керування може бути доданий, скажемо, у програму, написану на Delphi – і Trident буде використовуватися для доступу до задалегідь заданого веб-сторінці, для читання або зміни інформації, що перебуває на ній. Події елемента керування будуть перехоплюватися й передаватися в основну програму. Функції ядра Trident стануть доступні при додаванні бібліотеки mshtml.dll до програмного проекту.

3.2 Розробка структурної схеми

Структурна схема розробленої системи зображена на рисунку 3.1.

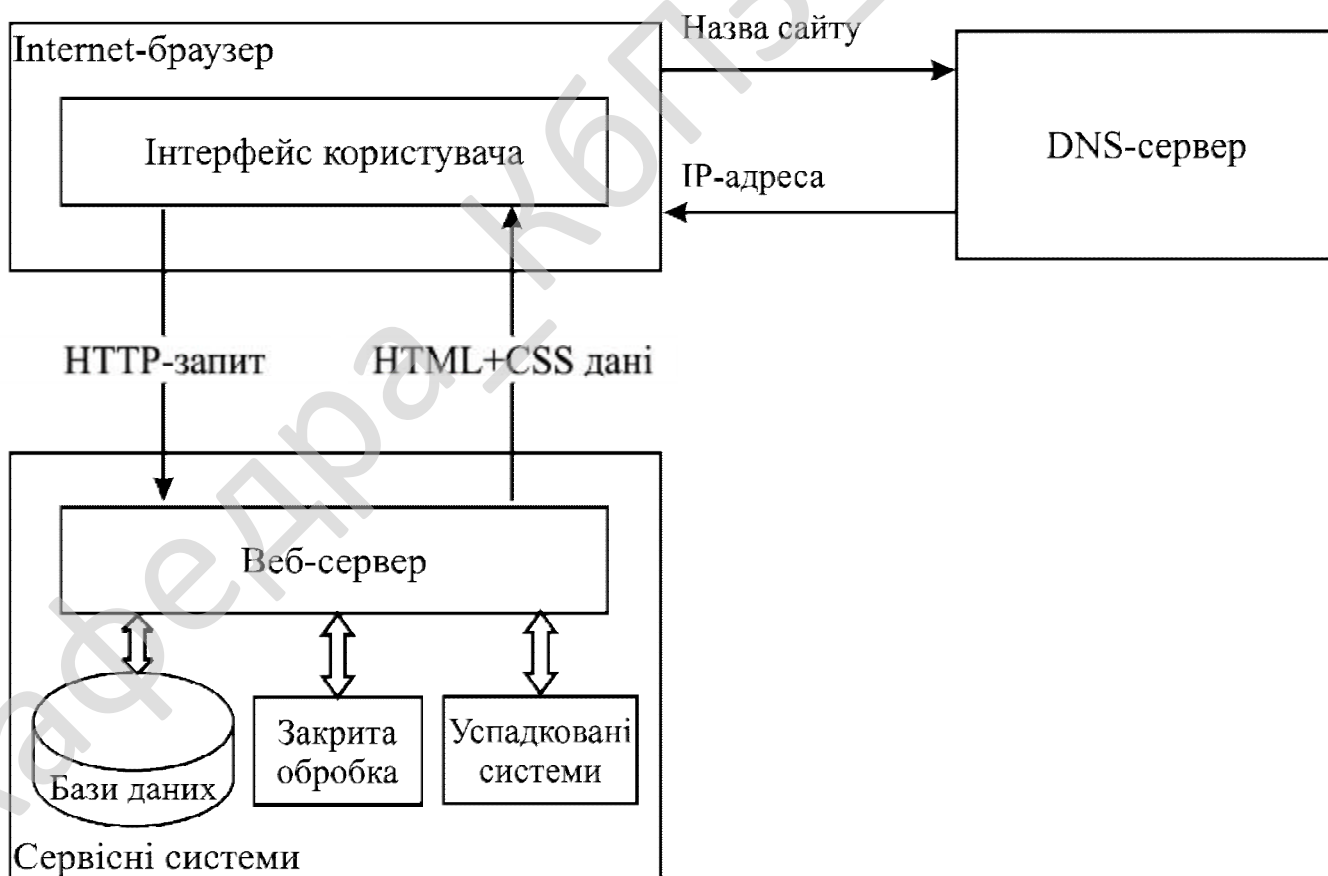


Рисунок 3.1 – Структурна схема системи

Веб-сервер

Веб-сервер – це програмне забезпечення, встановлене на комп'ютері, за допомогою якого по протоколу HTTP надається доступ до веб-сторінок. Найпоширенішими веб-серверами є Apache, що працює на платформі Unix/Linux і IIS (Internet Information Service), що працює під керуванням Windows. Також веб-сервером називається й сам комп'ютер, на якому встановлене це програмне забезпечення й зберігаються файли веб-сайтів. Для захисту сервера від атак і взлому звичайно встановлюється мережевий екран і відповідне програмне забезпечення.

Функції сервера:

- на сервері зберігаються різні ресурси (наприклад веб-сторінки);
- для кожного ресурсу встановлюються права доступу;
- сервер обробляє запити клієнтів, яким необхідний той або інший ресурс;
- сервер здійснює обмін інформацією із клієнтами й іншими серверами.

Internet-браузер, як складова системи перегляду хмарних сервісів (клієнт)

Розроблений Internet-браузер, як складова системи перегляду хмарних сервісів являється програмою клієнтом.

Функції клієнта:

- браузер, як складова системи перегляду хмарних сервісів здійснює запит необхідного ресурсу;
- браузер, як складова системи перегляду хмарних сервісів обробляє отриманий ресурс;
- у рядку адреси браузера набирається адреса сайту, на який хоче потрапити користувач (наприклад <http://www.eucp.com>);
- браузер, як складова системи перегляду хмарних сервісів відправляє запит спеціальному комп'ютеру, який зветься DNS-сервер (Domain Name System);
- DNS-сервер перетворить набрану адресу в числову (IP-адреса) адресу

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

сервера, на якому розташований сайт (наприклад 212.147.139.162) і поверне його браузеру;

– браузер, як складова системи перегляду хмарних сервісів відправляє запит на отриману адресу й у відповідь одержує запитуваний ресурс.

Після того, як ресурс переданий, з'єднання між клієнтом і сервером розривається.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема розробленої системи. З неї ми бачимо, що у системі існують наступні основні функціональні блоки:

- блок відображення веб-сторінки;
- навігація;
- інтерпретатор HTML коду;
- параметри відображення веб-сторінок
- меню користувача.

Функціональне призначення блоків очевидне з їх назв.

Більш детально розглянемо меню користувача. Воно складається з основних функціональних блоків, таких як:

- файл;
- редагування;
- вид;
- журнал;
- обране;
- сервіс;
- довідка.

Меню «Файл» має наступні вкладки:

- нова вкладка;
- відкрити;

Меню «Вид» дозволяє виконати наступні операції:

- переглянути HTML-код;
- вибрати кодування шрифту;
- вибрати розмір тексту.

Меню «Журнал» дозволяє передивитися список усіх відвіданих веб-сторінок, які зберігаються у кукісах.

Меню «Обране» дозволяє додати до списку збережених веб-посилань нові зсилки. Або швидко перейти на раніше збережені веб-посилання.

Меню «Сервіс» дозволяє виконати наступні операції:

- додати HTML-код;
- переглянути кукіси;
- визначити параметри доступу до Інтернету.

Останнім меню є меню «Довідка» у якому наводяться дані про автора програми та місце її написання.

3.4 Розробка діаграми процесів

На рисунку 3.3 зображена діаграма процесів розробленої системи.

Перший процес після запуску системи – це відкриття стартової сторінки.

Потім стають доступними для запуску наступні процеси:

- Вказання адреси сайту.
- Пошук сайту.
- Створення html-коду.
- Перегляд сторінок доданих в “Обране”.
- Перегляд журналу історії.

Процес створення html-коду запускає після себе процес перегляду створеного html-коду у браузері.



Рисунок 3.3 – Діаграма процесів системи

Після того як, здійснено процес завантаження веб-сторінки, що може відбутися після прямого вказання адреси сайту, або здійснення пошуку по вказаній користувачем фразі, стають доступними для запуску процеси:

- Запис веб-сторінки в обране.
- Поновлення веб-сторінки.
- Збереження веб-сторінки на диску.
- Друк веб-сторінки.
- Перегляд html-коду відкритої сторінки.

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.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 зображена блок-схема основної програми.

Виконання програми починається з виведення основного вікна програми.

За цим слідує виведення стартової веб-сторінки. Відповідно назва веб-сторінки відображається у полі URL.

Після виведення основної сторінки з відповідним URL можлива робота з наступними кнопками:

- перейти;
- вкладка «HTML-код»;
- вибір параметрів відображення веб-сторінки;
- збереження відкритої веб-сторінки;
- друкування веб-сторінки;
- здійснення пошуку;
- перегляд журналу;
- додати в обране;
- переглянути обране;
- додати HTML-код;
- переглянути Cookies;
- вихід.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

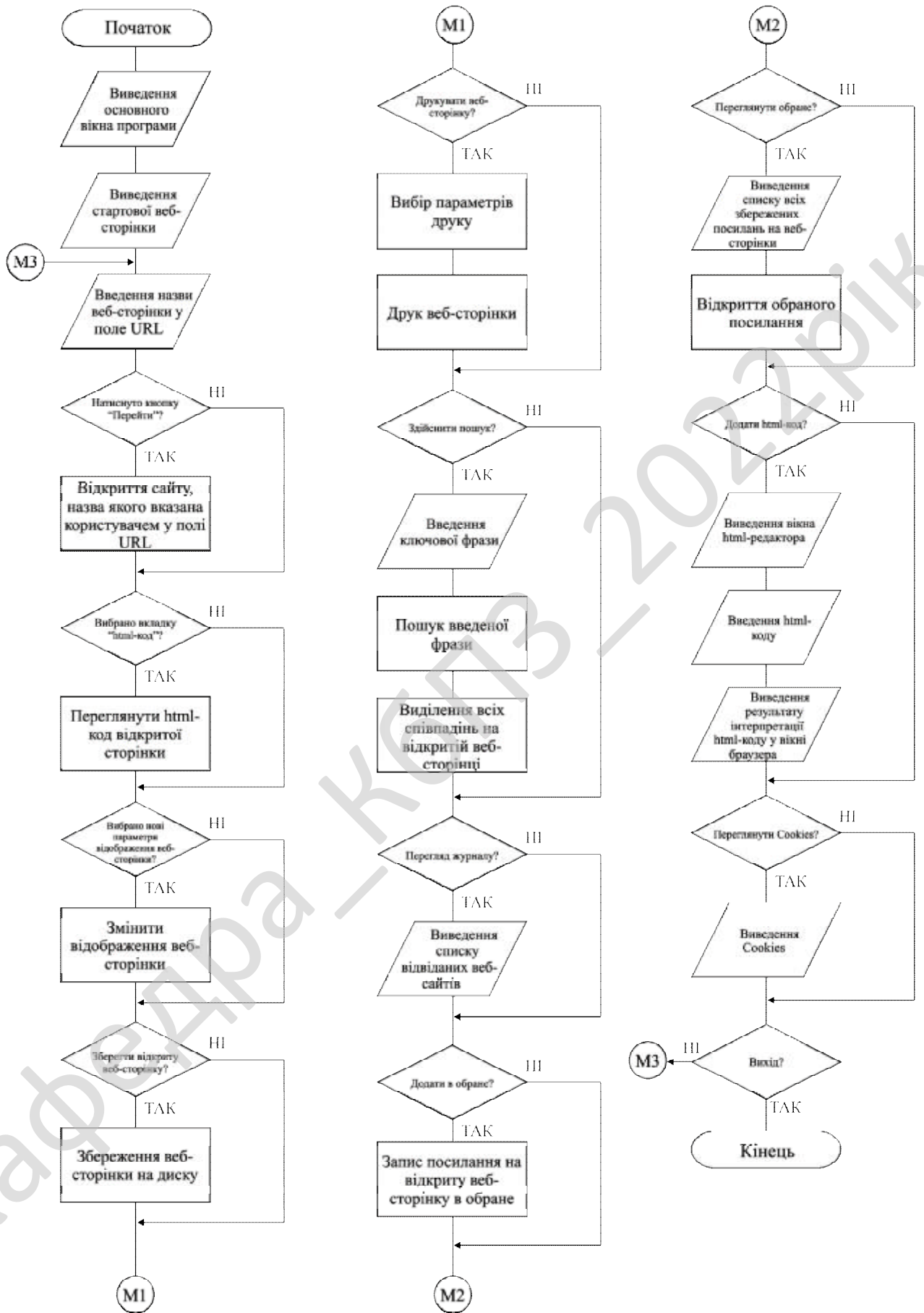


Рисунок 4.1 – Блок-схема роботи основної програми

Розглянемо дії, до яких приводить натискання цих кнопок у псевдокодї.

Вхід «Перейти».

Вихід «ТАК». У цьому випадку відбувається відкриття сайту, назва якого вказана на URL адресі.

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки. **Вхід «Вкладка «HTML-код»».**

Вихід «ТАК». У цьому випадку відбувається перегляд HTML-коду відкритої сторінки.

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки.

Вхід «Вибір параметрів відображення веб-сторінки».

Вихід «ТАК». У цьому випадку відбувається зміна відображення веб-сторінки.

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки.

Вхід «Збереження відкритої веб-сторінки».

Вихід «ТАК». У цьому випадку відбувається збереження відкритої веб-сторінки у файл на диску.

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки.

Вхід «Друкування веб-сторінки».

Вихід «ТАК». У цьому випадку відбувається вибір параметрів друку, та друк веб-сторінки.

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки.

Вхід «Здійснення пошуку».

Вихід «ТАК». У цьому випадку відбувається введення ключової фрази для пошуку. Після цього відбувається пошук введеної фрази та виділення всіх співпадінь на відкритій веб-сторінці.

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки.

Вхід «Перегляд журналу».

Вихід «ТАК». У цьому випадку відбувається виведення списку відвіданих веб-сайтів.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки.

Вхід «Додати в обране».

Вихід «ТАК». У цьому випадку відбувається запис посилання на відкриту веб-сторінку в обране.

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки.

Вхід «Переглянути обране».

Вихід «ТАК». У цьому випадку відбувається виведення списку всіх збережених посилань на веб-сторінки, та відкриття за потребою користувача обраного посилання.

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки.

Вхід «Додати HTML-код».

Вихід «ТАК». У цьому випадку відбувається виведення вікна HTML-редактору. У ньому вводиться HTML-код, який обробляється інтерпретатором. Відповідно після обробки інтерпретатором відбувається виведення результату у вікно браузера.

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки.

Вхід «Переглянути Cookies».

Вихід «ТАК». У цьому випадку відбувається виведення змісту папки Cookies.

Вихід «НІ». У цьому випадку відбувається перехід до наступної кнопки.

Вхід «Вихід».

Вихід «ТАК». У цьому випадку відбувається вихід з програми.

Вихід «НІ». У цьому випадку відбувається перехід до початку програми.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

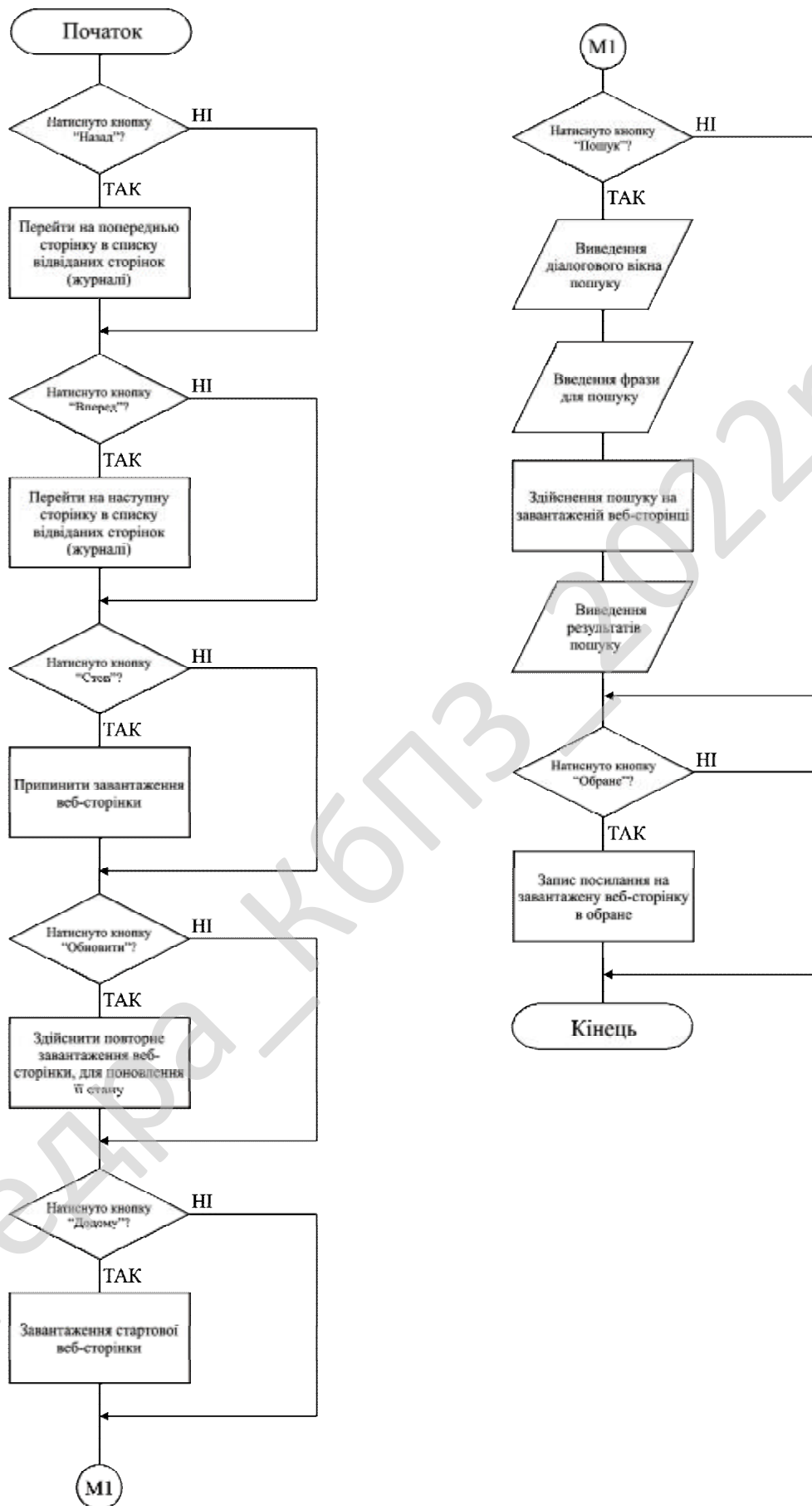


Рисунок 4.2 – Блок-схема роботи підпрограми навігації

На рисунку 4.2 зображено блок-схему роботи підпрограми навігації. Опишемо її у псевдокодї, як це зроблено вище.

Спершу відбувається перевірка натискання на кнопку «Назад»

Вхід «натискання на кнопку «Назад»».

Вихід «ТАК». У цьому випадку відбувається перехід на попередню сторінку в списку відвіданих сторінок (журналі).

Вихід «НІ». У цьому випадку відбувається перехід до наступної дії.

Вхід «натискання на кнопку «Вперед»».

Вихід «ТАК». У цьому випадку відбувається перехід на наступну сторінку в списку відвіданих сторінок (журналі)..

Вихід «НІ». У цьому випадку відбувається перехід до наступної дії.**Вхід** «натискання на кнопку «Стоп»».

Вихід «ТАК». У цьому випадку відбувається припинення завантаження веб-сторінок.

Вихід «НІ». У цьому випадку відбувається перехід до наступної дії.

Вхід «натискання на кнопку «Обновити»».

Вихід «ТАК». У цьому випадку відбувається повторне завантаження веб-сторінки, для поновлення її стану.

Вихід «НІ». У цьому випадку відбувається перехід до наступної дії.

Вхід «натискання на кнопку «Додому»».

Вихід «ТАК». У цьому випадку відбувається завантаження стартової веб-сторінки.

Вихід «НІ». У цьому випадку відбувається перехід до наступної дії.

Вхід «натискання на кнопку «Пошук»».

Вихід «ТАК». У цьому випадку відбувається виведення діалогового вікна пошуку, після чого вводиться фраза для пошуку, здійснюється пошук на завантаженій веб-сторінці та виводяться результати пошуку.

Вихід «НІ». У цьому випадку відбувається перехід до наступної дії.

Вхід «натискання на кнопку «Обране»».

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

Вихід «ТАК». У цьому випадку відбувається запис посилання на завантажену веб-сторінку в обране.

Вихід «НІ». У цьому випадку відбувається перехід до наступної дії.

Попередні установки

Спершу необхідно запустити Delphi. Саме він дозволяє дуже швидко й без особливих проблем написати свій власний браузер, як складова системи перегляду хмарних сервісів. Після запуску, необхідно перейти на закладку "Internet". Тут повинен бути компонент "WebBrowser" (він повинен знаходитися останнім). Якщо версія Delphi менше, ніж п'ята, то цього компонента може й не бути. Він може бути відсутнім, якщо не встановлені інтернет-компоненти (за замовчуванням вони відносяться).

Необхідно вибрати "Import Active Control" з меню "Component". У списку вибору цього вікна (зверху) необхідно знайти рядок "Microsoft Internet Controls (Version 1.1)" і натиснути кнопку "Install". Після цього відкривається вікно й запит. На запит необхідно відповісти "Yes".

Після того, як Delphi виконає заданий запит, з'явиться вікно, що повідомить про успішну установку нового компонента. Необхідно натиснути "OK" і закрити все, що відкрив Delphi. Для цього необхідно вибрати "Close All" з меню "File". Тепер встановлений компонент "WebBrowser", тільки він розташований на сторінці "Active" палітри компонентів.

Використовується "Microsoft Internet Controls", тобто движок встановленого на комп'ютері Microsoft Edge.

Необхідно створити новий проект ("File" -> "New Application") і відразу змінити заголовок і іконку.

Необхідно встановити на форму наш компонент "WebBrowser" (він перебуває на закладці "Internet" або "Active") – після цього з'явиться білий квадрат з ім'ям WebBrowser1. Після цього необхідно перемістити на форму "CoolBar", що перебуває на закладці "Win32" палітри компонентів. Це панелька, що повинна вирівнятися по верхньому краю на заданій формі. Тепер необхідно


```
StatusBar1.SimpleText := Text;  
end;
```

Тут привласнюється змінна "Text" (у ній зберігається текст підказки) в StatusBar1. Тепер можна бачити підказки в рядку стану.

Додамо індикатор завантаження. Для цього необхідно перемістити на форму ProgressBar із закладки "Win32". Змінити в нього властивість "Align" на "alBottom", щоб він перебував уздовж нижньої границі форми. Знову необхідно виділити WebBrowser1 і клацни по закладці "Events" в ObjectInspector. Двічі клацнути по рядку "OnProgressChange" і написати в створеній процедурі:

```
procedure TForm1.WebBrowser1ProgressChange(Sender: TObject; Progress,  
    ProgressMax: Integer);  
begin  
    ProgressBar1.Max := ProgressMax;  
    ProgressBar1.Position := Progress;  
end;
```

Тут створеному ProgressBar1 (індикатор завантаження) привласнюється максимальне значення (ProgressMax) і поточне значення (Progress).

Для зміни кнопок браузеру необхідно зробити наступні дії.. Необхідно переміститися на форму ImageList і зробити по ньому DoubleClick. Після цього відкриється вікно. Сюди потрібно додати картинки розміром 16x16. Для цього необхідно натиснути кнопку "Add", і відкрити стандартне вікно відкриття файлу. Потрібно знайти картинку й натиснути "Відкрити". Необхідно повторити цю процедуру 6 разів (6 картинок для 6-и кнопок). Після всього цього необхідно натиснути "ОК".

Тепер необхідно виділити ToolBar1 і в ObjectInspector-Е змінити властивість Images на "ImageList1". На кнопках повинні з'явитися картинки. Якщо картинка додавалася не в тій порядку, як вони стоять на формі, то можемо пересортувати їх за допомогою властивості ImageIndex у кнопки. Наприклад клацнути по кнопці "Стояти" і змінити ImageIndex на 0. На кнопці повинна з'явитися картинка, зазначена першою в ImageList1.

виконання, що стало причиною для критики і однією з перепон для обрання як нового стандарту.

Деталі RC6

Так само, як і RC5, RC6 – повністю параметризована сім'я алгоритмів шифрування. Для специфікації алгоритму з конкретними параметрами, прийнято позначення RC6-w/r/b, де

- W – довжина машинного слова в бітах.
- R – число раундів.
- B – довжина ключа в байтах. Можливі значення 0 .. 255 байт.

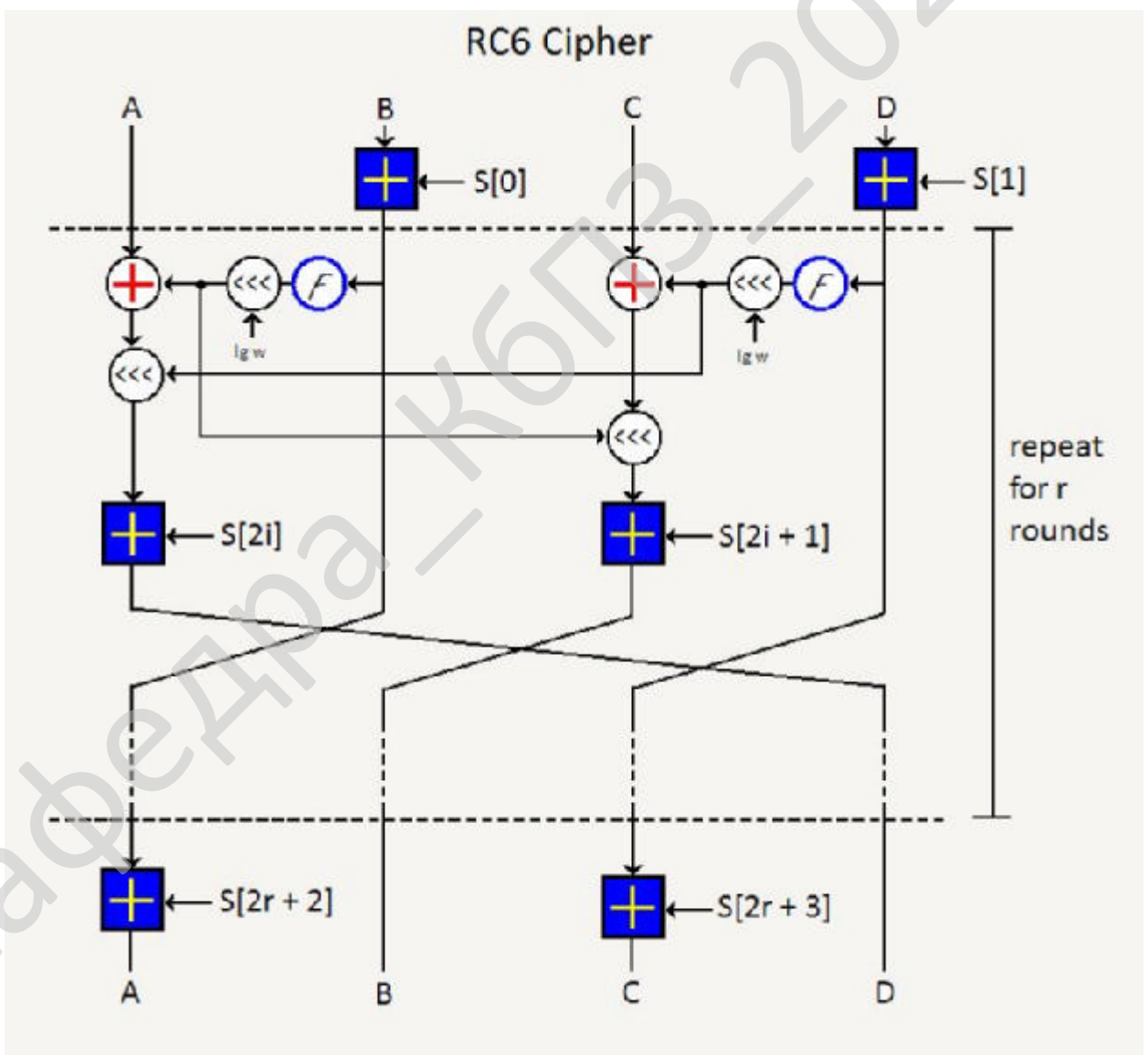


Рисунок 4.3 – Структура RC6

Для того щоб відповідати вимогам AES, блочний шифр повинен працювати з 128-бітовими блоками. Так як RC5 – виключно швидкий блочний шифр, розширення його, щоб працювати з 128-бітовими блоками, привело б до використання двох 64-бітових робочих регістрів. Але архітектура і мови програмування ще не підтримують 64-бітні операції, тому довелося змінити проект так, щоб використовувати чотири 32-бітних регістри замість двох 64-бітних.

Кафедра _ КБПЗ _ 2022 рік

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60


5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблений Internet-браузер, як складова системи перегляду хмарних сервісів дозволяє шукати та переглядати веб-сайти, зберігати веб-сторінки на диску та відправляти їх на друк, переглядати html-код відкритої сторінки тощо.

Програмно-апаратні вимоги:

- Загальний обсяг ОЗП: 512 Мбайт.
- Вільний простір на жорсткому диску: 3,5 Мбайт.
- Операційна система Microsoft Windows 10/11.
- Підключення до мережі Інтернет.

Головне вікно програми зображене на рисунку 5.1.

Для пошуку сайту треба ввести ключову фразу для пошуку у поле «Шукати» та натиснути кнопку з зображенням лупи . Після цього на вкладці «Веб-браузер, як складова системи перегляду хмарних сервісів» з'явиться список сайтів з вказаною фразою.

Для завантаження сайту слід набрати його назву у полі «URL» та натиснути кнопку «Перейти». Після цього він відкриється на вкладці «Веб-браузер, як складова системи перегляду хмарних сервісів». На рисунку 5.2 зображений приклад відкриття сайту <http://www.google.com.ua/>.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

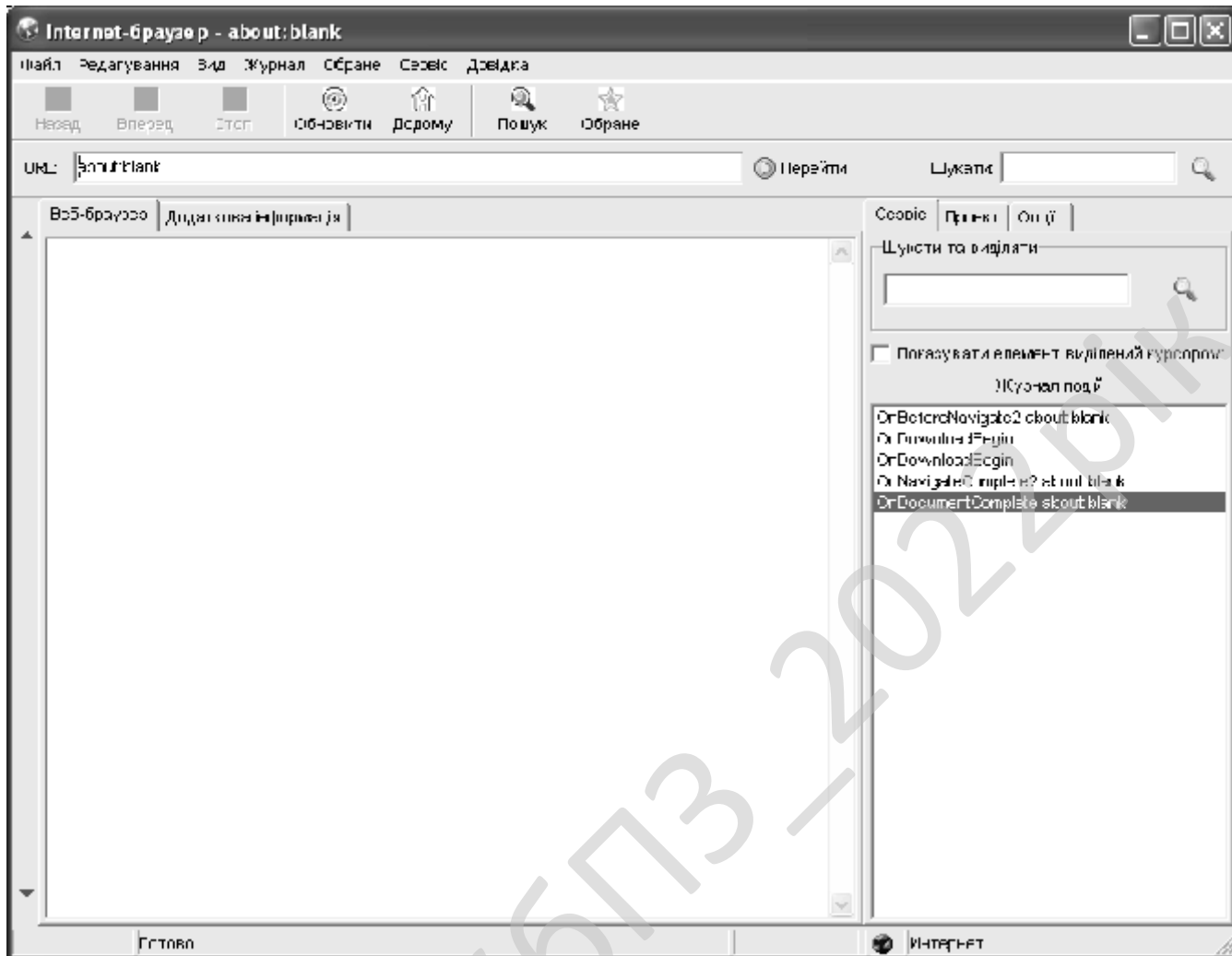


Рисунок 5.1 – Головне вікно програми

Для перегляду html-коду відкритої веб-сторінки слід перейти на вкладку «Додаткова інформація» і вибрати із вкладок розташованих на ній внизу «html-код» (рисунок 5.3). Або вибрати пункт меню користувача **Вид->Переглянути html-код**, у такому разі він відкриється у стандартному блокноті.

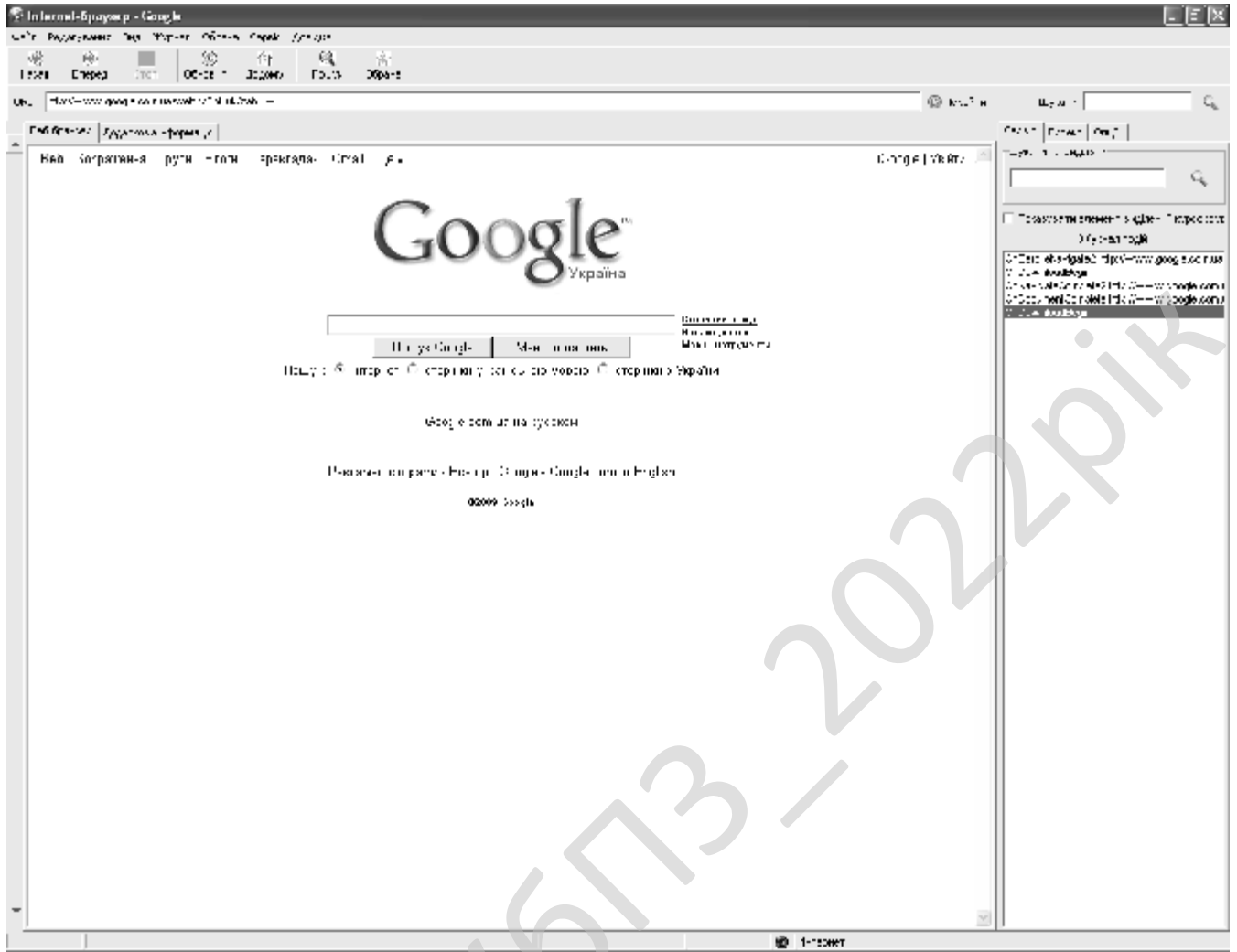


Рисунок 5.2 – Приклад відкриття сайту

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63



Рисунок 5.4 – Створення html-коду

Для перегляду довідки про програму та її автора, слід натиснути пункт меню **Довідка->Про програму...**, після цього на екрані з'явиться вікно, зображене на рисунку 5.5.

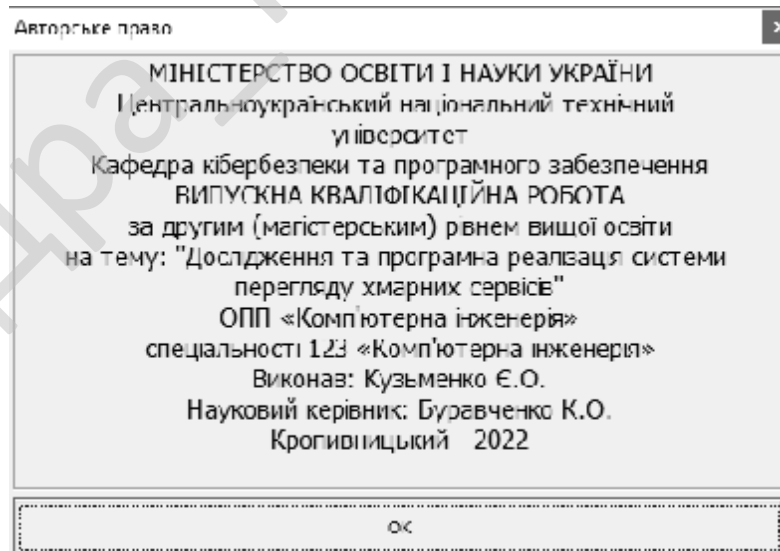


Рисунок 5.5 – Довідка про програму

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи перегляду хмарних сервісів.

Метою розробки є дослідження та програмна реалізація системи перегляду хмарних сервісів.

Об'єктом дослідження є процес перегляду хмарних сервісів.

Предметом дослідження є методи перегляду хмарних сервісів.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод перегляду хмарних сервісів.
- Розроблено вітчизняний продукт перегляду хмарних сервісів, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Продовження табл. 7.1

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

Продовження табл. 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	40000
33. Норматив додаткової зарплати, % :	Нд	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	55
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.:

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

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

де A – коефіцієнт Боєма, $A=2,45$; $Size$ – загальний об'єм відлагодженого програмного коду, тис. рядків; B – показник ступеня, що визначається співвідношенням

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

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

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

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

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

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

де $\prod V_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

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

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

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

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

де C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПО згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 100 = 168 \text{ люд/день}$$

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

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	168	Ф 7.1-7.4
Впровадження	13	Д13
Всього	209	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

де F_{pq} – плановий фонд робочого часу одного спеціаліста, днів,
 T_{nz} – трудомісткість розробки програмного забезпечення люд-дні,

$$Ч = \frac{209 \cdot 1}{60 - 5} = 3,8 \text{ ставки}$$

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

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

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

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	10	900	15
Монітор	60	10	600	10
Клавіатура	30	10	300	5
Маніпулятор «мишка»	30	10	300	5
Принтер матричний	60	1	60	1
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	2	40	0,67
Концентратор-маршрутизатор	30	3	90	1,5
Кабельні господарства ЛОМ на 1 м. п.	2,5	250	625	10,42
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	55,92

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

$$\Phi_{\text{др}}^c = \frac{56 \cdot 3}{1,2} = 140 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість 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 сайт	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців:

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	7890	23670
Продакт-менеджер	0,25	7000	5250
Інженер-програміст	3,8	8700	99180
Інженер-електронщик	0,3	7000	6300
Інженер-системотехнік	0,25	7000	5250
Адміністратор мережі	0,5	7000	10500
Системний програміст	0,25	7000	5250
Дизайнер WEB	0,25	7000	5250
Інженер-верстальник	0,25	7000	5250
Бухгалтер-економіст	0,5	7000	10500
Всього за період розробки	$R_{cn}=7,35$	-	$\Phi_{роб}=176400$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{co} = \frac{176400}{7,25 \cdot 60} = 400 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

$$B_{y\partial} = R_{cn}^1 S_y \Pi_{пл}, \quad (7.9)$$

де R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць. S_y – питома площа на одне робоче місце, m^2 ; $\Pi_{пл}$ – вартість одного квадратного метра площі, грн.

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

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

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

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

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

де $\Pi_{м}$ – ціна меблів для одного робочого місця, грн.

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7. Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу Інтернет магазину Компбест за 01.11.22 – джерело <https://compbest.com.ua>.

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	LG W2363V-WF Wide LCD 2ms, 70 000:1, 300кд/м2, 170/160, D-Sub / Glossy White	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струменевий	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	10947	8757,6	96333,6
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	114885,1

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	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57442,5
Група 5			
4. Вимірювальні пристрої	5190	-	-
5. Господарський інвентар	28000	-	-
Всього по групі	33190	25	8297,5
Нематеріальні активи			
6. Нематеріальні активи	40000	10	4000
Разом	$K_p = 1596075$		$A_p = 140140$

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців

$$z_o = \frac{z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де N_e – Кількість екземплярів програм, шт.

$$Z_o = 400 \cdot 209 / 40 = 2090 \text{ грн}$$

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

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де H_q – норматив додаткової зарплати, %

$$Z_d = 2090 \cdot 10 \cdot 0,01 = 209 \text{ грн}$$

Відрахування на соціальні потреби за нормативом $H_c=37\%$ від суми основної та додаткової зарплати

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де H_c – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 37 (2090 + 209) = 506 \text{ грн}$$

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

$$G_{ocn} = Z_o \cdot H_g \cdot 0,01, \quad (7.14)$$

де H_g – загальногосподарські витрати, %

$$G_{ocn} = 2090 \cdot 15 \cdot 0,01 = 314 \text{ грн}$$

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

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де Z_{M1} – вартість паперу, грн., Z_{M2} – вартість запам'ятовуючих пристроїв, грн., Z_{M3} – вартість фарби, картриджей, тонеру, грн., N_e – кількість екземплярів програм, шт.

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

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

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

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

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

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

де C_{∂} – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box, DVD-R LG 4,7Gb, 16x speed Cake box – 49,2 грн/шт.

$$Z_{M2} = 49,2 \cdot 10 = 492 \text{ грн.}$$

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

$$Z_{M3} = \sum C_{з}, \quad (7.18)$$

де: $C_{з}$ – 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.

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

$$Z_M = (105 + 492 + 1702) / 40 = 57 \text{ грн.}$$

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

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

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

$$O_n = 2090 \cdot 15 \cdot 0,01 = 314 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 40$ прим.)

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

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 140140 \cdot 3 / (40 \cdot 12) = 876 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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

$$C_n = 2090 + 209 + 506 + 314 + 57 + 314 + 876 = 4366 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 55%

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

де P_n – рівень рентабельності, %

$$P_p = 0,01 \cdot 55 \cdot 4366 = 2401 \text{ грн.}$$

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. Основна зарплата виконавців	Z_o	2090
2. Додаткова зарплата виконавців	Z_d	209
3. Відрахування на соціальні потреби	C_{oc}	506
4. Загальногосподарські витрати	Γ_{ocn}	314
5. Витрати на матеріали	Z_m	57
6. Освоєння нових операційних систем, мов програмування	O_n	314
7. Амортизація основних фондів	A_m	876
8. Повна собівартість програмного забезпечення	C_n	4366
9. Плановий прибуток	P_p	2401
10. Ціна підприємства $C_n + P_p$	C_n	6767
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{об} \cdot C_n$	$ПДВ$	1353,4
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	9168

Витрати на оплату праці:

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

де T_p – кількість годин обслуговування системи за рік, год.; Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 600 годин на рік до 150 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 600 \cdot 100 \cdot 1,1 \cdot 1,22 = 80520 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 150 \cdot 100 \cdot 1,1 \cdot 1,22 = 20130 \text{ грн.}$$

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

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

$$Z_{ел \text{ баз}} = 0,475 \cdot 2455 \cdot 3,2 = 3732 \text{ грн}$$

$$Z_{ел \text{ нов}} = 0,475 \cdot 1227 \cdot 3,2 = 1865 \text{ грн}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	9168	–	2292
Всього відрахувань	-	–	9168	–	2292

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

$$T_{cn} = \frac{9168}{84252 - 24287} = 0,2 \text{ року}$$

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.	40
2. Повна собівартість розробленої програми	Грн	4366
3. Ціна розробленої програми	Грн.	6767
4. Плановий прибуток від реалізації розробленої програми	Грн.	2401
5. Рентабельність програмної продукції	%	55
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1596075
7. Загальний прибуток від реалізації програмної продукції	Грн.	96040
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	61005
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років.	0,5
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	9168
11. Величина економічного ефекту у користувача програмної продукції	Грн.	57673
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,2

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.

Кафедра _ КБПЗ _ 2022 рік

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Сучасний розвиток технічного та технологічного стану виробництва передбачає постійну автоматизацію та оптимізацію виробничих процесів. Комп'ютер – невід'ємна складова сучасного життя. За допомогою обчислювальної техніки вирішують складні робочі задачі, ведуться наукові дослідження, створюються архітектурні креслення і твори мистецтва. Сьогодні, напевно, важко уявити компанію, господарська діяльність в якій здійснювалась би без використання комп'ютерної техніки. Незважаючи на видиму безпеку та розвитку сучасних технологій, при роботі за комп'ютером є ряд чинників, які можуть вплинути на здоров'я людини. Через масовий характер робіт, що виконуються працівниками за допомогою комп'ютера, законодавством України чітко врегульовано норми та вимоги до використання комп'ютерної техніки на підприємстві, безпосередньо й охорона праці на підприємстві при роботі за комп'ютером.

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями, затверджені наказом Мінсоцполітики від 14.02.2018р. № 207, зареєстровані в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 [5].

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою ЕОМ.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

У розділі даної магістерської роботи висвітлюються основні питання охорони праці працівників, робота яких пов'язана з роботою за комп'ютером, планування робочого приміщення, де працюють користувачі ПК; параметри мікроклімату, освітленість робочих місць та виробничих приміщень; шумові завади.

Правильна організація і раціональне устаткування робочого місця можливість ефективно і з якнайменшими витратами праці виконувати свої функції, плідно спілкуватися співробітниками і підлеглими, підтримувати високу працездатність і робочий настрій.

Велике значення має раціональна конструкція і розташовує елементів робочого місця, що важливе для підтримки оптимальної робочої пози людини-оператора, а також необхідно дотримувати правильний режим праці і відпочинку.

Що стосується питання охорони праці людини необхідно вирішувати на всіх стадіях трудового процесу незалежно від виду професійної діяльності.

Забезпечення безпечних і здорових умов праці в значній мірі залежить від правильної оцінки небезпечних, шкідливих виробничих факторів. Однакові по складності зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища, надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин.

Робота працівників пов'язана з роботою за комп'ютером, тому актуальною є розгляд саме умов праці та стану охорони праці працівників які постійно працюють з комп'ютерною технікою.

Завдання даного розділу полягає у тому, щоб розробити якісний програмний продукт необхідно організувати безпеку на робочому місці програміста. Під час проектування безпеки робочому місці з ПК необхідно домагатися високої якості та надійності технічного забезпечення, але й створювати комфортні параметри довкілля для розробників.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

8.2 Характеристика умов праці програміста

В приміщенні, в якому проводиться розробка і дослідження програмного продукту, відсутні умови, які можуть створювати підвищену або особливо підвищену небезпеку, тому воно відноситься до класу звичайних приміщень згідно ПУЕ. Джерелом живлення є трифазна мережа напруги 380/220 В з глухо заземленою нейтралі, з частотою 50 Гц згідно За пожежо-вибухонебезпекою приміщення відноситься до класу В. В таблиці 8.1 наведена загальна характеристика приміщення щодо вибухопожеженобезпеки та важкості робіт.

Таблиця 8.1 – Загальна характеристика приміщення щодо вибухопожеженобезпеки та важкості робіт

Характеристика приміщень за вибухопожежною категорією та класом зони	Загальна характеристика приміщення	Категорія за важкістю робіт згідно ГН 3.3.5-8.6.6.1 -2002
В – пожежонебезпечне клас П – П	Звичайне без ознак хімічного забруднення та нормальної вологості і за санітарними нормами	1а.....до 139 Вт/м ² 1б.....до 140-174 Вт/м ² Клас умов праці – оптимальний

Температура повітря в приміщенні визначається температурою зовнішнього повітря і тепловою енергією, що виділяється всередині приміщення. Джерелами теплоти в даному приміщенні є люди, електроустаткування, а також освітлювальні прилади в темний час доби. Зовнішнім джерелом надлишкового тепла є сонячна радіація у світлий час доби. Робота, виконувана в даному приміщенні, відноситься до категорії І-а. Людиною в цьому випадку виділяється

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

до 120 ккал теплової енергії в годину. Вологість повітря в приміщенні визначається вологістю атмосферного і видихуваного людьми повітря, а також випарами з поверхні шкіри.

У таблиці 8.2 приведені оптимальні значення параметрів мікроклімату для категорії ваги робіт I-а, а також фактичні значення цих параметрів у розглянутому приміщенні. У приміщеннях з використанням обчислювальної техніки рекомендується застосування тільки оптимальних значень показників мікроклімату, тобто таких, при яких людина відчуває себе комфортно.

Таблиця 8.2 – Значення параметрів небезпечних факторів праці

Найменування параметра	Значення параметра		Нормативний документ	
	Фактичне	Нормоване		
Освітленість штучна, лк	300	300	ДБН.В 2.5-28:2018 [2]	
Значення КПО,%	1,0	1,1	ДБН.В 2.5-28:2018 [2]	
Повітрообмін,м /год	взимку	76	80	ДСН 3.3.6.042-99[3]
	влітку	36	80	ДСН 3.3.6.042-99 [3]
Температура повітря. °С	взимку	22	21-25	ДСанПіН 3.3.2-007-98
	влітку	24	27-28	ДСанПіН 3.3.2-007-98
Відносна вологість,%	взимку	60	<75	ДСанПіН 3.3.2-007-98
	влітку	55	<60	ДСанПіН 3.3.2-007-98
Швидкість переміщення повітря, м/с	взимку	0,16	<0,2	ДСанПіН 3.3.2-007-98
	влітку	0,10	<0,2	ДСанПіН 3.3.2-007-98

У приміщенні немає виділення шкідливих газів. Тому що в ньому не проводиться монтажних робіт, пайки чи інших робіт, при яких виділяються шкідливі гази.

Для нормалізації параметрів повітряного середовища також періодично здійснюється провітрювання приміщення і вологе прибирання. У всьому будинку діє встановлена загально обмінна витяжна вентиляція.

Раціональне освітлення приміщення сприяє кращому виконанню виробничого завдання і забезпеченню комфорту при роботі. Для забезпечення нормального освітлення застосовуються природне, однобічне, бічне і штучне освітлення, а також сполучене, нормуються згідно ДБН В.2.5-28-2006 Природне і штучне освітлення [1]. Дані по нормах освітлення наведені в таблиці 8.3

Таблиця 8.3 – Норми освітлення

Мінімальний розмір об'єкта розрізювання, мм	Фон	Контрас	Розряд, під розряд зорової роботи	Нормоване значення		
				Природне освітлення КПО,%	Штучне освітлення	
					Е _{мін.} лк	Тип ламп
Від 0,3 до 0,5	Світлий	Середній	Шг	1,5	300	Газоро зрядні

За результатами виміру освітленості величина освітленості від системи загального штучного висвітлення лежить у межах – 320 лк, що відповідає вимогам, які пред'являються до даного приміщення.

Основними джерелами шуму на робочих місцях, обладнаних відео дисплейними терміналами, є принтер, сканер факс і обладнання для кондиціонування повітря, в самих відео дисплейних терміналів – вентилятори систем охолодження і трансформатори.

Згідно ДСанПіН 3.3.2.007-98 [2] допустимий еквівалентний рівень шуму для робочого місця програміста складає 50 дБА (акустичних децибела).

8.3 Розробка заходів з умов поліпшення охорони праці

У комплексі заходів, що сприяють удосконаленню організації праці програміста, збереженню здоров'я і підвищенню працездатності, його велике значення має організація робочих місць.

Проводячи аналіз умов праці в розглянутому приміщенні, ми одержали наступні результати:

- обсяг приміщення, що приходить на одному працюючого, відповідає нормативному значенню;
- показники мікроклімату відповідають нормативному значенню;
- акустичні умови роботи в нормі.

Важлива роль в ефективному забезпеченні праці належить моральному мікроклімату. Відносини працівників повинні ґрунтуватися на об'єктивності, доброзичливості, взаємодопомозі, глибокій повазі до кожного члена колективу, турботі про молодих співробітників.

Для забезпечення найбільш ефективного виконання обов'язків, плануючи розпорядок, слід дотримуватися таких принципів:

- для зняття втоми через кожні 1,5-2 год. роботи перерви для відпочинку на 5-10 хв.;
- для усунення монотонності виконуваних робіт чергувати характер праці.

За умови неправильної організації праці та відпочинку, втома може нагромаджуватися щоденно й переходити в перевтому або захворювання. У зв'язку з цим режим праці та відпочинку користувачів ПК необхідно будувати з урахуванням працездатності, яка змінюється протягом доби.

Виходячи із наступного можна зробити висновок, що основною причиною втомлюваності та зниження працездатності працівника, який постійно працює за комп'ютером є психофізіологічний фактор, тому основною пропозицією правильна організація робочого місця з урахуванням ергономічних вимог, а також дотримання регламентованого режиму праці та відпочинку.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

8.4 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: прокат сталевий круглого перерізу діаметром 35 мм., довжиною $L=2$ м., та горизонтальний електрод – металева полоса з перетином $40 \cdot 4$ мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – у ряд.

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунта: $H=0,5$ м. Відстань між вертикальними заземлювачами (електродами) $A=2$ м. Глибина закладення горизонтального контура заземлення $t=0,7$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,7+2/2=1,7 \text{ м.}$$

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

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

де

$\psi = 1,36$ – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багат шаровому ґрунті [10];

$\rho = 40$ Ом·м. – табличне значення питомого опору нижнього шару ґрунта (глина) [10].

Діаметр вертикального електрода (задан):

$$D_{\text{в}} = 35 \text{ мм.} = 0,035 \text{ м.}$$

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи перегляду хмарних сервісів.

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 перегляду хмарних сервісів.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем перегляду хмарних сервісів.
- Досліджена система перегляду хмарних сервісів.
- На основі отриманих результатів досліджень створена програмна реалізація системи перегляду хмарних сервісів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання перегляду хмарних сервісів.

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

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

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

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

Програма реалізована на мові високого рівня RAD Studio Delphi 10.4. 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 RC6.

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 57673 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,2 роки.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кузьменко Є.О. Дослідження та програмна реалізація системи перегляду хмарних сервісів // Збірник праць молодих науковців ЦНТУ. – Вип. 13. – Кропивницький: ЦНТУ, 2022.
2. Королев А.В. Адаптивная маршрутизация в корпоративных сетях / А.В. Королев, Г.А. Кучук, А.А. Пашнев. – Х.: ХВУ, 2003. – 224 с.
3. Кучерявый Е.А. Управление трафиком и качество обслуживания в сети Интернет / Евгений Андреевич Кучерявый. – СПб.: Наука и техника, 2004. – 336 с.
4. Кучук Г.А. Управление ресурсами инфотелекоммуникаций / Г.А. Кучук, Р.П. Гахов, А.А. Пашнев. – М.: Физматлит, 2006. – 220 с.
5. Лагутин В.С., Степанов С.Н. Телетрафик мультисервисных сетей связи / В.С. Лагутин, С.Н. Степанов. – М.: Радио и связь, 2000. – 320 с.
6. Майника Э. Алгоритмы оптимизации на сетях и графах: пер. с англ. / Э. Майника; под ред. Е.К. Масловского. – М.: Мир, 1981. – 321 с.
7. Смирнов А.А. Разработка математической GERT-модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / А.А.Смирнов, Мохамад Гани Абу Таам // Информационные системы в управлении, образовании, промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2014. – 498 с.
8. Смирнов А.А. Метод управления доступом в интеллектуальных узлах коммутации / Мохамад Гани Абу Таам, А.А.Смирнов // Информационные технологии и защита информации в информационно-коммуникационных системах: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2015. – 486 с.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

9. Смирнов А.А. Математическая GERT-модель технологии передачи метаданных в облачные антивирусные системы / В.В.Босько, А.А.Смирнов, И.А.Березюк, Мохамад Гани Абу Таам // Збірник наукових праць "Системи обробки інформації". – Випуск 1(117). – Х.: ХУПС – 2014. – С. 137-141.

10. Смирнов А.А. Структурно-логическая GERT-модель технологии распространения компьютерных вирусов / А.А.Смирнов, И.А.Березюк, Мохамад Гани Абу Таам // Системи управління, навігації та зв'язку. – Випуск 1(29). – П.: ПНТУ. – 2014. – С. 120-125.

11. Смирнов А.А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Гани Абу Таам, А.А. Смирнов, А.В. Коваленко, С.А. Смирнов // Збірник наукових праць "Системи обробки інформації". – Випуск 9(125). – Х.: ХУПС – 2014. – С. 105-110.

12. Смирнов А.А. Математическая модель интеллектуального узла коммутации с обслуживанием информационных пакетов различного приоритета / Мохамад Гани Абу Таам, А.А. Смирнов, Н.С. Якименко, С.А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. Випуск 4 (41). – Харків: ХУПС. – 2014. – С. 48-52.

13. Смирнов А.А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в телекоммуникационных системах и сетях / Мохамад Гани Абу Таам, А.А. Смирнов, Н.С. Якименко, С.А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 4(17). – Харків: ХУПС. – 2014. – С.90-95.

14. Смирнов А.А. Усовершенствованный алгоритм управления доступом к «облачным» телекоммуникационным ресурсам / Мохамад Гани Абу Таам, А.А. Смирнов, Н.С. Якименко, С.А. Смирнов // Збірник наукових праць "Системи обробки інформації". – Випуск 1(126). – Х.: ХУПС – 2015. – С. 150-153.

15. Смирнов А.А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты данных / Мохамад Гани Абу

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

Таам, А.А. Смирнов, С.А. Смирнов // Системи озброєння і військова техніка. – Випуск 3(43) – Х.: ХУПС – 2015. – С. 100-107.

16. Смирнов А.А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 3(19). – Х.: ХУПС. – 2015. – С. 134-141.

17. Mohamad Abou Taam Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.

18. Смирнов А.А. GERT-модель технологии передачи данных в облачные антивирусные системы / А.А. Смирнов, В.В. Босько, Мохамад Гани Абу Таам // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». м. Харків. 12-13 березня 2014 р. – Харків. АВВ МВС. – 2014. – С. 18-19.

19. Смирнов А.А. Математическое моделирование технологии передачи сигнатур в облачные антивирусные системы / Мохамад Гани Абу Таам, А.А. Смирнов // Збірник тез VI міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 17-18 квітня 2014 р. – Харків: ХНЕУ. – 2014. – С. 260.

20. Смирнов А.А. Анализ требований к качеству обслуживания в информационно-телекоммуникационных системах / А.А. Смирнов, Мохамад Гани Абу Таам // Збірник тез XVI міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 11-12 квітня 2014 р. – Кіровоград: КНТУ. – 2014. – С. 124-126.

21. Смирнов А.А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / Мохамад Гани Абу Таам, С.А. Смирнов // Збірник тез науково-

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

практичної конференції «Інформаційні технології та комп'ютерна інженерія». м. Кіровоград. 4 грудня 2014 р. – Кіровоград: КНТУ. – 2014. – С. 168.

22. Смирнов А.А. Исследование математических моделей технологии распространения компьютерных вирусов / А.А. Смирнов, Мохамад Гани Абу Таам, С.А. Смирнов // Збірник наукових праць міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 25-28 лютого 2015 р. – Київ: Європейський університет. – 2015. – С. 90-91.

23. Смирнов А.А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Збірник тез всеукраїнської науково-практичної конференції «Інформаційна безпека держави, суспільства та особистості». м. Кіровоград. 16 квітня 2015. – Кіровоград: КНТУ. – 2015. – С. 50-52.

24. Смирнов А.А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Гани Абу Таам, С.А. Смирнов // Збірник тез VII міжнародної науково-практичної конференції «Проблеми і перспективи розвитку ІТ-індустрії». м. Харків. 17-18 квітня 2015 р. – Харків: ХНЕУ. – 2015. – С. 14.

25. Смирнов А.А. Реализация метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Гани Абу Таам // Збірник тез XVII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 17-18 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 91-92.

26. Смирнов А.А. Реализация математической модели интеллектуального узла коммутации для обеспечения защищенности телекоммуникационной сети / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Збірник тез II Міжнародної науково-практичної Інтернет-конференції «Інформаційна та економічна безпека» (INFECO-2015)». м. Харків. 21-22 травня 2015 р. – Харків: ХІБС УБС НБУ. – 2015. – С. 20-24.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

27. Смирнов А.А. Разработка математической модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Сборник тезисов XI международной конференции "Стратегия качества в промышленности и образовании". г. Варна. Болгария. 01 – 06 июня 2015 г – Варна. ТУВ. – 2015. – С. 488-491

28. Смирнов А.А. Метод управления доступом к облачным телекоммуникационным ресурсам для обеспечения защиты данных / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Збірник тез Міжнародної науково-практичної конференції «Комп'ютерні технології та інформаційна безпека». м. Кіровоград. 2-3 липня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 4-5.

29. Смирнов А.А. Имитационная модель системы управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Збірник тез першої всеукраїнської науково-практичної конференції «Перспективні напрями захисту інформації». м. Затока. 7-9 вересня 2015 р. – Одеса: ОНАЗ. – 2015. – С. 90-94.

30. МСЭ-Т Рекомендация G.101. Международные телефонные соединения и цепи – Общие определения //11/2003. [Электронный ресурс]. – Режим доступа до ресурсу: [http://www. telecom61.ru/SharedFiles/Download.aspx? ...pageid=106](http://www.telecom61.ru/SharedFiles/Download.aspx?...pageid=106)

31. Одом Ш. Коммутаторы CISCO / Ш. Одом, Х. Ноттингем – М.: "Кудиц-Образ", 2003. – 528 с.

32. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов / В.Г. Олифер, Н.А. Олифер. – 2-е изд. – СПб.: Питер, 2007. – 958 с.

33. Руководство по технологиям объединенных сетей. 4-е изд. / пер.с англ. и ред. А.Н. Крикуна – М.: Изд. дом «Вильямс», 2005. – 1040 с.

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

34. Свами М.Н., Тхуласираман К. Графы, сети и алгоритмы: пер. с англ. / М.Н. Свами, К. Тхуласираман; под ред. В.А. Горбатова. – М.: Мир, 1984. – 454 с.

35. Семенов С.Г. Анализ методов прогнозирования в телекоммуникационных сетях автоматизированных систем управления / С.Г.Семенов // Збірник наукових праць «Системи управління, навігації та зв'язку», – К.:ЦНДІ навігації і управління, – 2008.-Вип. 2(6) .- С.134-137

36. Семенов С.Г. Математическая модель процесса доставки информационных пакетов в компьютерной сети системы критического применения / С.Г.Семенов, И.В.Ильина // Науково-технічний журнал «Радіоелектронні і комп'ютерні системи» Х.:ХАІ, – 2008.-Вип. 1(28) – С.162-165

37. Семенов С.Г. Оптимизация трафика на основе сбалансированной загрузки информационно-телекоммуникационной сети // Системи обробки інформації. – Х.: ХВУ, 2004. – № 8(36). – С.206-210

38. Семенов С.Г. Математическая модель мультисервисного канала связи на основе экспоненциальной GERT-сети / С.Г. Семенов, Є.В. Мелешко, Я.В. Ілюшко // Системи озброєння і військова техніка. – Х.:ХУ ПС. – 2011. –Вип. 3(27). – С. 64-67.

39. Семенов С.Г. Математична модель системи криптографічного захисту електронних повідомлень на основі GERT-мережі / С.Г. Семенов, О.О. Сур // Системи управління, навігації та зв'язку. – К.:ЦНДІ навігації і управління. – 2012. – Том 1. Вип. 1(21). – С. 131-137

40. Семенов С.Г. Исследования вероятностно-временных характеристик мультисервисного канала связи с использованием математического аппарата GERT-сети / С.Г. Семенов, В.В. Босько, І.А. Березюк // Системи обробки інформації. – Х.: ХУ ПС. – 2012. – Том 1. Вип. 3(101). – С. 139-142.

41. Семенов С.Г. Моделирование защищенного канала связи с использованием экспоненциальной GERT-сети / С.Г. Семенов,

50. Уолрэннд Дж. Телекоммуникационные и компьютерные сети / Дж. Уолрэннд. – М.: Постмаркет, 2001. – 480 с.
51. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1103 с.
52. Шелухин О.И. Фрактальные процессы в телекоммуникациях: моногр. / О.И. Шелухин, А.М. Тенякшев, А.В. Осин – М.: Радиотехника, 2003. – 480 с.
53. A. Elwalid Routing and Protection in GMPLS Networks: From Shortest Paths to Optimized Designs / A. Elwalid, D. Mitra, I. Saniee, and I. Widjaja. // Journal of lightwave technology. – 2003. – №21(11), P. 2828-28-38.
54. A.B. Bagula Online Traffic Engineering: The Least Interference Optimization Algorithm / A.B. Bagula, M. Botha, and A.E Krzesinski. // IEEE Communications Society – 2004, P. 1232-1236.
55. Anees. Shaikh Evaluating the Impact of Stale Link State on Quality-of-Service Routing / Anees Shaikh, Jennifer Rexford, and Kang G. Shin. // IEEE/ACM Transactions on Networking. – 2001. – №9(2), P. 162-176.
56. Chakraborty Basabi Simultaneous Search for Multiple Routes using Genetic Algorithm / Basabi Chakraborty // IEEE International Conference on Computational Intelligence for Measurement System and Applications Boston, MA, USA, 14-16, July 2004, P. 77-80/
57. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ>
58. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>
59. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

60. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

61. 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>

62. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ IBM 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 до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/4358>

63. 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>

64. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : “Колос”, 1973. – 238 с.

65. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: <https://cpo.stu.cn.ua>

66. 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 19.09.22).

					ВКРМ-122.22.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.22.0004.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Кузьменко Є.О.				<i>Дослідження та програмна реалізація системи перегляду хмарних сервісів</i>	Літ.	Аркуш	Аркушів
Перевірів	Буравченко К.О.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КН-21М-1,4			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи перегляду хмарних сервісів.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 18-13 від 17.08.2022 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи перегляду хмарних сервісів.

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;

					ВКРМ-122.22.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- 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 перегляду хмарних сервісів;
- 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.

					ВКРМ-122.22.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище RAD Studio Delphi 10.4.

					ВКРМ-122.22.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2022 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута характеристика умов праці програміста.

					ВКРМ-122.22.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

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

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2022 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 22.12.2022 р.

					ВКРМ-122.22.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Буравченко К.О.

*Дослідження та програмна реалізація
системи перегляду хмарних сервісів*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 27

Літера: РП

Кропивницький – 2022 року

Основна програма**Файл Webbrowser.dpr основної програми**

```
program Webbrowser;

uses
  Forms,
  Unit1 in 'Unit1.pas' {frmMyBrowser},
  uAddHTML in 'uAddHTML.pas' {frmAddHTML},
  about in 'about.pas' {TForm_about};

{$R *.res}

begin
  Application.Initialize;
  Application.Title := 'Internet-браузер';
  Application.CreateForm(TfrmMyBrowser, frmMyBrowser);
  Application.CreateForm(TForm_about, Form_about);
  Application.Run;
end.
```

Кафедра _ КБПЗ _ 2022 рік

Файл Unit1.pas основної програми

```

unit Unit1;

{$I versions.inc}

interface

uses
  Windows, Messages, SysUtils, {$IFDEF DELPHI6UP}Variants, {$ENDIF}AppEvnts,
  Dialogs, ExtDlgs, Graphics, Menus, ExtCtrls, ImgList, Controls, StdCtrls,
  ComCtrls,
  OleCtrls, SHDocVw, ToolWin, Buttons, WBFuncs, Classes, Forms,
  HistoryMenu, FavMenu, ActiveX, about;

{ ***** }
// Клас TWebbrowser
// Заголовок WM_CLOSE
type
  TWebbrowser = class(TMyBrowser)
  private
    // Збереження CSC_NAVIGATEFORWARD та CSC_NAVIGATEBACK станів
    FNavForward: Boolean;
    FNavBack: Boolean;
    FTitle: string;
    // люк події WM_CLOSE
    procedure WMClose(var Msg: TWMClose); message WM_CLOSE;
  public
  end;

{ ***** }
// Розширення TStatusBar
// Прийнятні компоненти
type
  TStatusBar = class(ComCtrls.TStatusBar)
  public
    constructor Create(AOwner: TComponent); override;
  end;
{ ***** }

type
  TfrmMyBrowser = class(TForm)
    Panell: TPanel;
    edURL: TEdit;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Print1: TMenuItem;
    PrintPreview1: TMenuItem;
    PageSetup1: TMenuItem;
    Properties1: TMenuItem;
    N1: TMenuItem;
    Edit2: TMenuItem;
    Copy1: TMenuItem;
    Pastel: TMenuItem;
    Cut1: TMenuItem;
    N2: TMenuItem;
    Search1: TMenuItem;
    ImageListToolBar: TImageList;
    Panell2: TPanel;
    CoolBarMainIcons: TCoolBar;
    ToolBar1: TToolBar;
    ToolbtnBack: TToolButton;
    ToolBtnForward: TToolButton;
    ToolBtnStop: TToolButton;
    ToolButton10: TToolButton;
    ToolBtnRefresh: TToolButton;
  end;

```

```
ToolButton11: TToolButton;
ToolBtnSearch: TToolButton;
ToolBtnFavorites: TToolButton;
ToolBtnHome: TToolButton;
edSearch: TEdit;
btnSearch: TSpeedButton;
btnGo: TSpeedButton;
StatusBar1: TStatusBar;
N3: TMenuItem;
Exit1: TMenuItem;
ProgressBar1: TProgressBar;
SelectAll1: TMenuItem;
N4: TMenuItem;
Label1: TLabel;
Extras1: TMenuItem;
AddHTML1: TMenuItem;
DesignModel: TMenuItem;
SaveAs1: TMenuItem;
N5: TMenuItem;
NewTabBlank: TMenuItem;
OfflineModel: TMenuItem;
OpenDialog1: TOpenDialog;
Open1: TMenuItem;
InternetOptions1: TMenuItem;
ImageListObjectView: TImageList;
Timer1: TTimer;
Label8: TLabel;
N8: TMenuItem;
CharacterSet1: TMenuItem;
N9: TMenuItem;
PCPopup: TPopupMenu;
CloseTab1: TMenuItem;
DuplicateTab1: TMenuItem;
Panel8: TPanel;
PageControl: TPageControl;
TabSheet1: TTabSheet;
WebBrowser1: TWebBrowser;
TabSheet7: TTabSheet;
PageControlInfos: TPageControl;
TabSheet2: TTabSheet;
TabSheet3: TTabSheet;
LbLinks: TListBox;
TabSheet4: TTabSheet;
LbFrames: TListBox;
TabSheet5: TTabSheet;
lbImages: TListBox;
TabSheet6: TTabSheet;
LbFields: TListBox;
TabObjectView: TTabSheet;
TreeView1: TTreeView;
Panel6: TPanel;
SpeedButtonScrollBottom: TSpeedButton;
SpeedButtonScrollTop: TSpeedButton;
Splitter1: TSplitter;
Cookie1: TMenuItem;
View1: TMenuItem;
ViewSource1: TMenuItem;
Label9: TLabel;
SourceEdit: TMemo;
PageControl1: TPageControl;
TabSheet8: TTabSheet;
TabSheet9: TTabSheet;
Panel2: TPanel;
GroupBox1: TGroupBox;
btnSearchAndHighlight: TSpeedButton;
edSearchAndHighlight: TEdit;
OpenPictureDialog1: TOpenPictureDialog;
N10: TMenuItem;
GroupBox3: TGroupBox;
```

```

chkLiveResize: TCheckBox;
chkMultipleSelection: TCheckBox;
SaveDialog1: TSaveDialog;
TabSheet10: TTabSheet;
GroupBox2: TGroupBox;
chk3DBorderStyle: TCheckBox;
PnlElementInf1: TPanel;
lvEventLog: TListBox;
chkNoScrollBars: TCheckBox;
Panel9: TPanel;
Zoom1: TMenuItem;
Largest1: TMenuItem;
Largel: TMenuItem;
Medium1: TMenuItem;
Small1: TMenuItem;
Smallest1: TMenuItem;
N7: TMenuItem;
ApplicationEvents1: TApplicationEvents;
chkDisablePopupMenu: TCheckBox;
PnlElementInfo: TPanel;
PnlHandle: TPanel;
Label3: TLabel;
edElementTag: TEdit;
Panel11: TPanel;
Label4: TLabel;
edElementId: TEdit;
Panel5: TPanel;
Label5: TLabel;
edElementInnerHTML: TEdit;
Panel7: TPanel;
Label7: TLabel;
edElementClassName: TEdit;
Panel10: TPanel;
Label6: TLabel;
edElementInnerText: TEdit;
Panel4: TPanel;
chkShowElementInfo: TCheckBox;
TabSheet11: TTabSheet;
PlainTextEdit: TMemo;
Panel3: TPanel;
Label2: TLabel;
edElementHREF: TEdit;
Panel13: TPanel;
Label10: TLabel;
edElementSrc: TEdit;
N11: TMenuItem;
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure btnGoClick(Sender: TObject);
procedure WebBrowser1ProgressChange(Sender: TObject; Progress,
    ProgressMax: Integer);
procedure WebBrowser1TitleChange(Sender: TObject;
    const Text: WideString);
procedure Print1Click(Sender: TObject);
procedure PrintPreview1Click(Sender: TObject);
procedure PageSetup1Click(Sender: TObject);
procedure Properties1Click(Sender: TObject);
procedure Copy1Click(Sender: TObject);
procedure Paste1Click(Sender: TObject);
procedure Cut1Click(Sender: TObject);
procedure Edit2Click(Sender: TObject);
procedure Search1Click(Sender: TObject);
procedure ToolBtnBackClick(Sender: TObject);
procedure ToolBtnForwardClick(Sender: TObject);
procedure ToolBtnStopClick(Sender: TObject);
procedure ToolBtnSearchClick(Sender: TObject);
procedure ToolBtnFavoritesClick(Sender: TObject);
procedure ToolBtnHomeClick(Sender: TObject);
procedure btnSearchClick(Sender: TObject);

```

```

procedure File1Click(Sender: TObject);
procedure WebBrowser1StatusTextChange(Sender: TObject;
  const Text: WideString);
procedure Exit1Click(Sender: TObject);
procedure WebBrowser1CommandStateChange(Sender: TObject;
  Command: Integer; Enable: WordBool);
procedure FormResize(Sender: TObject);
procedure WebBrowser1DownloadComplete(Sender: TObject);
procedure SelectAll1Click(Sender: TObject);
procedure StatusBar1DrawPanel(StatusBar: TStatusBar;
  Panel: TStatusPanel; const Rect: TRect);
procedure FormShow(Sender: TObject);
procedure edURLKeyPress(Sender: TObject; var Key: Char);
procedure AddHTML1Click(Sender: TObject);
procedure Extras1Click(Sender: TObject);
procedure SpeedButtonScrollBottomClick(Sender: TObject);
procedure SpeedButtonScrollTopClick(Sender: TObject);
procedure WebBrowser1NewWindow2(Sender: TObject; var ppDisp: IDispatch;
  var Cancel: WordBool);
procedure DesignModelClick(Sender: TObject);
procedure NewTabBlankClick(Sender: TObject);
procedure OfflineModelClick(Sender: TObject);
procedure Open1Click(Sender: TObject);
procedure InternetOptions1Click(Sender: TObject);
procedure edSearchKeyPress(Sender: TObject; var Key: Char);
procedure Timer1Timer(Sender: TObject);
procedure btnSearchAndHighlightClick(Sender: TObject);
procedure edSearchAndHighlightKeyPress(Sender: TObject; var Key: Char);
procedure PageControlChanging(Sender: TObject;
  var AllowChange: Boolean);
procedure PageControlMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure CloseTab1Click(Sender: TObject);
procedure DuplicateTab1Click(Sender: TObject);
procedure PageControlInfosChange(Sender: TObject);
procedure PageControlChange(Sender: TObject);
procedure WebBrowser1DocumentComplete(Sender: TObject;
  const pDisp: IDispatch; var URL: OleVariant);
procedure WebBrowser1DownloadBegin(Sender: TObject);
procedure WebBrowser1BeforeNavigate2(Sender: TObject;
  const pDisp: IDispatch; var URL, Flags, TargetFrameName, postData,
  Headers: OleVariant; var Cancel: WordBool);
procedure Cookie1Click(Sender: TObject);
procedure ViewSource1Click(Sender: TObject);
procedure View1Click(Sender: TObject);
procedure N9Click(Sender: TObject);
procedure WebBrowser1NavigateComplete2(Sender: TObject;
  const pDisp: IDispatch; var URL: OleVariant);
procedure Zoom1Click(Sender: TObject);
procedure Smallest1Click(Sender: TObject);
procedure ToolBtnRefreshClick(Sender: TObject);
procedure chk3DBorderStyleClick(Sender: TObject);
procedure urkishWindows1Click(Sender: TObject);
procedure SaveAs1Click(Sender: TObject);
procedure ApplicationEvents1Message(var Msg: tagMSG;
  var Handled: Boolean);
procedure chkNoScrollBarsClick(Sender: TObject);
procedure chkShowElementInfoClick(Sender: TObject);
procedure edURLDb1Click(Sender: TObject);
procedure N11Click(Sender: TObject);
private
  { *****
}

  { Private declarations }
  FPrevBrowser: TWebbrowser;
  HistoryMenu: THistoryMenu;
  FavoritesMenu: TFavoritesMenu;
  function DrawZoneIcon(WB: TWebbrowser): TIcon;
  function CreateTabBrowser(sURL, sCaption: string): TTabSheet;

```

```

    procedure SetWBMenuEnabled(WB: TWebbrowser; Sender: TMenuItem);
    procedure AddEventLog(s: string);
    procedure HistoryMenuURLSelected(Sender: TObject; Url: string);
    procedure FavoritesMenuURLSelected(Sender: TObject; Url: string);
public
    { ***** }
}
    { Public declarations }
    function GetCurrentWB: TWebbrowser;
end;

const
    APP_CAPTION = 'Internet-браузер - ';

var
    frmMyBrowser: TfrmMyBrowser;

implementation

uses
    ShellAPI, MSHTML, UrlMon, CommCtrl, WinInet,
    uAddHTML, ComObj;

{$R *.dfm}

{ ***** }
// Показати діалогове вікно друку сторінки
procedure TfrmMyBrowser.Print1Click(Sender: TObject);
begin
    WB_ShowPrintDialog(GetCurrentWB);
end;

{ ***** }
// Показати діалогове вікно перегляду сторінки перед друком
procedure TfrmMyBrowser.PrintPreview1Click(Sender: TObject);
begin
    WB_ShowPrintPreview(GetCurrentWB);
end;

{ ***** }
// Показати діалогове вікно вибору параметрів
procedure TfrmMyBrowser.PageSetup1Click(Sender: TObject);
begin
    WB_ShowPageSetup(GetCurrentWB);
end;

{ ***** }
// Показати діалогове вікно властивостей оглядача
procedure TfrmMyBrowser.Properties1Click(Sender: TObject);
begin
    WB_ShowPropertiesDialog(GetCurrentWB);
end;

{ ***** }
// Копіювати
procedure TfrmMyBrowser.Copy1Click(Sender: TObject);
begin
    WB_Copy(GetCurrentWB);
end;

{ ***** }
// Вставити

```

```

procedure TfrmMyBrowser.Paste1Click(Sender: TObject);
begin
  WB_Paste(GetCurrentWB);
end;

{ ***** }
// Вирізати

procedure TfrmMyBrowser.Cut1Click(Sender: TObject);
begin
  WB_Cut(GetCurrentWB);
end;

{ ***** }
// Виділити все

procedure TfrmMyBrowser.SelectAll1Click(Sender: TObject);
begin
  WB_SelectAll(GetCurrentWB);
end;

{ ***** }
// Поновити сторінку

procedure TfrmMyBrowser.ToolBtnRefreshClick(Sender: TObject);
begin
  WB_Refresh(GetCurrentWB);
  ToolBtnStop.Enabled := True;
end;

{ ***** }
// Шукати слово

procedure TfrmMyBrowser.ToolBtnSearchClick(Sender: TObject);
begin
  WB_ShowFindDialog(GetCurrentWB);
end;

{ ***** }
// Шукати фразу

procedure TfrmMyBrowser.Search1Click(Sender: TObject);
begin
  WB_ShowFindDialog(GetCurrentWB);
end;

{ ***** }
// Переглянути html-код

procedure TfrmMyBrowser.ViewSource1Click(Sender: TObject);
begin
  WB_ShowSourceCode(GetCurrentWB);
end;

{ ***** }
// Показати опції програми

procedure TfrmMyBrowser.InternetOptions1Click(Sender: TObject);
begin
  InvokeCMD(GetCurrentWB, HTMLID_OPTIONS);
end;

{ ***** }
// Повернутися назад
// Перехід на один пункт назад у списку історії

procedure TfrmMyBrowser.ToolbtnBackClick(Sender: TObject);
begin
  WB_GoBack(GetCurrentWB);

```

```

end;

{ ***** }
// Перейти вперед
// Перехід на один пункт вперед у списку історії

procedure TfrmMyBrowser.ToolBtnForwardClick(Sender: TObject);
begin
  WB_GoForward(GetCurrentWB);
end;

{ ***** }
// Прокручувати відкриту сторінку за допомогою повзунка

procedure TfrmMyBrowser.SpeedButtonScrollBottomClick(Sender: TObject);
begin
  WB_ScrollToBottom(GetCurrentWB);
end;

{ ***** }
// Прокручувати у початок сторінки

procedure TfrmMyBrowser.SpeedButtonScrollTopClick(Sender: TObject);
begin
  WB_ScrollToTop(GetCurrentWB);
end;

{ ***** }
// Перемикач відключення

procedure TfrmMyBrowser.OfflineModelClick(Sender: TObject);
begin
  OfflineModel.Checked := not OfflineModel.Checked;
  SetGlobalOffline(OfflineModel.Checked);
end;

{ ***** }
// заголовок закриття події TWebbrowser

procedure TWebbrowser.WMClose(var Msg: TWMClose);
begin
  Msg.Result := 0;
  // Редагування стартової сторінки
  Navigate('about:blank');
end;

{ ***** }
// створення TStatusBar для батькалюбих елементів управління

constructor TStatusBar.Create(AOwner: TComponent);
begin
  inherited Create(AOwner);
  ControlStyle := ControlStyle + [csAcceptsControls];
end;

{ ***** }

procedure TfrmMyBrowser.btnGoClick(Sender: TObject);
begin
  if edURL.Text <> '' then
  begin
    lvEventLog.Clear;
    WB_Navigate(GetCurrentWB, edURL.Text);
    // встановлення документа
    WB_SetFocus(GetCurrentWB);
  end;
end;

{ ***** }

```

```

// Заголовок edURL dblClick Подія: Вибрати все

procedure TfrmMyBrowser.edURLDblClick(Sender: TObject);
begin
  edURL.SelectAll;
end;

{ ***** }
// Заголовок введення ключа при редагуванні

procedure TfrmMyBrowser.edURLKeyPress(Sender: TObject; var Key: Char);
begin
  if Key = #13 then
  begin
    Key := #0;
    btnGoClick(self);
  end;
end;

procedure TfrmMyBrowser.edSearchKeyPress(Sender: TObject; var Key: Char);
begin
  if Key = #13 then
  begin
    Key := #0;
    btnSearchClick(self);
  end;
end;

procedure TfrmMyBrowser.edSearchAndHighlightKeyPress(Sender: TObject;
var Key: Char);
begin
  if Key = #13 then
  begin
    Key := #0;
    btnSearchAndHighlightClick(self);
  end;
end;

{ ***** }
// Появлення/зникнення 3D стилю ліній

procedure TfrmMyBrowser.chk3DBorderStyleClick(Sender: TObject);
begin
  WB_Set3DBorderStyle(GetCurrentWB, not chk3DBorderStyle.Checked);
end;

{ ***** }
// // Появлення/зникнення скролбара

procedure TfrmMyBrowser.chkNoScrollBarsClick(Sender: TObject);
begin
  WB_ShowScrollBar(GetCurrentWB, chkNoScrollBars.Checked);
end;

{ ***** }
// встановлення типу шрифту

procedure TfrmMyBrowser.urkishWindows1Click(Sender: TObject);
begin
  if WB_SetCharSet(GetCurrentWB, TMenuItem(Sender).Hint) then
    TMenuItem(Sender).Checked := True;
end;

{ ***** }
// Пошук тексту в TWebBrowser

procedure TfrmMyBrowser.btnSearchAndHighlightClick(Sender: TObject);
begin

```

```

WB_SearchAndHighlightText(GetCurrentWB, edSearchAndHighlight.Text);
end;

{ ***** }
// Отримання поточного TWebbrowser

function TfrmMyBrowser.GetCurrentWB: TWebbrowser;
begin
  Result := nil;
  with PageControl do
    if ActivePage.ControlCount > 0 then
      begin
        if ActivePage.Controls[0] is TWebbrowser then
          begin
            Result := (TWebbrowser(ActivePage.Controls[0]));
          end else
            Result := FPrevBrowser;
        end;
      end;
end;

{ ***** }
// Завантаження стартової сторінки

procedure TfrmMyBrowser.NewTabBlankClick(Sender: TObject);
begin
  CreateTabBrowser('about:blank', 'Blank Page');
end;

{ ***** }
// Відкриття сторінки HTML

procedure TfrmMyBrowser.Open1Click(Sender: TObject);
begin
  if OpenDialog1.Execute then
    CreateTabBrowser(OpenDialog1.FileName,
    ExtractFileName(OpenDialog1.FileName));
end;

{ ***** }
// Зберегти відкриту сторінку на диску

procedure TfrmMyBrowser.SaveAs1Click(Sender: TObject);
begin
  WB_Save(GetCurrentWB);
end;

{ ***** }
// повернення сторінки з усіма куками

procedure TfrmMyBrowser.Cookie1Click(Sender: TObject);
var
  sCookie: string;
begin
  sCookie := WB_GetCookie(GetCurrentWB);
  if Length(sCookie) = 0 then
    ShowMessage('There are no cookies in this webpage.')
  else
    ShowMessage(sCookie);
end;

{ ***** }
// Отримання елемента під курсором мишки

function GetElementAtPos(Doc: IHTMLDocument2; x, y: integer): IHTMLDocument2;
begin
  Result := nil;
  Result := Doc.elementFromPoint(x, y);
end;

```

```

{ ***** }
// Меню історії: OnURLSelectedHistory

procedure TfrmMyBrowser.HistoryMenuURLSelected(Sender: TObject; Url: string);
var
  WB: TWebbrowser;
begin
  WB := GetCurrentWB;
  if Assigned(WB) then
    WB.Navigate(Url);
end;

// Меню фаворитів: OnUrlSelected

procedure TfrmMyBrowser.FavoritesMenuURLSelected(Sender: TObject; Url: string);
var
  WB: TWebbrowser;
begin
  WB := GetCurrentWB;
  if Assigned(WB) then
    WB.Navigate(Url);
end;

{ ***** }

procedure TfrmMyBrowser.FormCreate(Sender: TObject);
begin
  // створити пункт меню «Журнал», що містить список історії
  HistoryMenu := THistoryMenu.Create(self);
  HistoryMenu.MainMenu := MainMenu1;
  HistoryMenu.MenuPos := 4;
  HistoryMenu.CreateMenu;
  HistoryMenu.OnURLSelected := HistoryMenuURLSelected;
  MainMenu1.Items[3].Caption := 'Журнал';
  // створити пункт меню «Обране»
  FavoritesMenu := TFavoritesMenu.Create(self);
  FavoritesMenu.OnUrlSelected := FavoritesMenuURLSelected;
  FavoritesMenu.MainMenu := MainMenu1;
  FavoritesMenu.MenuPos := 5;
  FavoritesMenu.CreateMenu;
  MainMenu1.Items[4].Caption := 'Обране';

  FPrevBrowser := Webbrowser1;
  // Ініціалізувати Webbrowser1 властивості
  Webbrowser1.FNavForward := False;
  Webbrowser1.FNavBack := False;
  Webbrowser1.FTitle := '';
end;

{ ***** }

// Завантажити стартову сторінку

procedure TfrmMyBrowser.FormShow(Sender: TObject);
begin
  Webbrowser1.GoHome;
  FormResize(nil);
end;

{ ***** }

procedure TfrmMyBrowser.FormDestroy(Sender: TObject);
begin
  // очищення меню історії
  HistoryMenu.Free;
end;

// кінець *****

```

```

{ ***** }
// Додавання подій браузера до lvEventLog ListBox

procedure TfrmMyBrowser.AddEventLog(s: string);
begin
  lvEventLog.ItemIndex := lvEventLog.Items.Add(s);
end;

{ ***** }
// Появлення прогресу завантаження сторінки

procedure TfrmMyBrowser.WebBrowser1ProgressChange(Sender: TObject; Progress,
ProgressMax: Integer);
var
  Position: Integer;
begin
  try
    // Position := (Progress * 100) div ProgressMax;
    // ProgressBar1.Position := Position;
    // StatusBar1.Panels[0].Text := Format('%d "% loaded..."', [Position]);
  except

  end;
end;

{ ***** }
// Появлення заголовку веб-браузера

procedure TfrmMyBrowser.WebBrowser1TitleChange(Sender: TObject;
const Text: WideString);
begin
  Caption := APP_CAPTION + Text;
  TWebbrowser(Sender).FTitle := Text;
end;

{ ***** }
Меню файлів: дозволити, заборонити

procedure TfrmMyBrowser.SetWBMenuEnabled(WB: TWebbrowser; Sender: TMenuItem);
begin
  if Assigned(WB) then
    Sender.Enabled := Assigned(WB.Document)
  else
    Sender.Enabled := False;
end;

procedure TfrmMyBrowser.File1Click(Sender: TObject);
var
  CurrentWB: TWebbrowser;
begin
  CurrentWB := GetCurrentWB;
  SetWBMenuEnabled(CurrentWB, Print1);
  SetWBMenuEnabled(CurrentWB, PrintPreview1);
  SetWBMenuEnabled(CurrentWB, PageSetup1);
  SetWBMenuEnabled(CurrentWB, Properties1);
  SetWBMenuEnabled(CurrentWB, SaveAs1);
  OfflineModel.Checked := IsGlobalOffline;
end;

{ ***** }
// Меню «Редагування»: Вибір меню, дозвіл, заборона

procedure TfrmMyBrowser.Edit2Click(Sender: TObject);
var
  CurrentWB: TWebbrowser;
begin
  CurrentWB := GetCurrentWB;

```

```

Copy1.Enabled := WB_QueryCommandEnabled(CurrentWB, 'Copy');
Paste1.Enabled := WB_QueryCommandEnabled(CurrentWB, 'Paste');
Cut1.Enabled := WB_QueryCommandEnabled(CurrentWB, 'Cut');
SetWBMenuEnabled(CurrentWB, SelectAll1);
SetWBMenuEnabled(CurrentWB, Search1);
end;

{ ***** }
// Меню «Сервіс»: Вибір меню, дозвіл, заборона

procedure TfrmMyBrowser.Extras1Click(Sender: TObject);
var
  CurrentWB: TWebbrowser;
begin
  CurrentWB := GetCurrentWB;
  SetWBMenuEnabled(CurrentWB, AddHTML1);
  SetWBMenuEnabled(CurrentWB, InternetOptions1);
  SetWBMenuEnabled(CurrentWB, Cookie1);
end;

{ ***** }
// Меню «Вид»: Вибір меню, дозвіл, заборона

procedure TfrmMyBrowser.View1Click(Sender: TObject);
var
  CurrentWB: TWebbrowser;
begin
  CurrentWB := GetCurrentWB;
  SetWBMenuEnabled(CurrentWB, DesignModel);
  SetWBMenuEnabled(CurrentWB, CharacterSet1);
  SetWBMenuEnabled(CurrentWB, ViewSource1);
end;

{ ***** }
// Зупинити завантаження
// Відмінняє будь-яке незавершене відкриття сторінки чи
// завантаження і зупиняє будь-які динамічні елементи сторінки,
// такі як фонові звуки й анімація.

procedure TfrmMyBrowser.ToolBtnStopClick(Sender: TObject);
begin
  WB_Stop(GetCurrentWB);
end;

{ ***** }
// оновлення TWebbrowser

procedure TfrmMyBrowser.ToolBtnFavoritesClick(Sender: TObject);
begin
  ShowMessage('Ще не відбулося');
end;

{ ***** }
// Завантажити сторінку «Додому»

procedure TfrmMyBrowser.ToolBtnHomeClick(Sender: TObject);
begin
  GetCurrentWB.GoHome;
end;

{ ***** }
// Пошук сторінки по заданій фразі за допомогою Гуглу.

procedure TfrmMyBrowser.btnSearchClick(Sender: TObject);
const
  GOOGLE_QUERY = 'http://www.google.com/search?ie=ISO-8859-1&hl=de&q=';
var

```

```

    sQuery: string;
begin
    sQuery := GOOGLE_QUERY + edSearch.Text;
    WB_Navigate(GetCurrentWB, sQuery);
end;

{ ***** }
// Додаток завершення

procedure TfrmMyBrowser.Exit1Click(Sender: TObject);
begin
    Close;
end;

{ ***** }
// дозвіл, заборона кнопок «вперед», «назад»

procedure TfrmMyBrowser.WebBrowser1CommandStateChange(Sender: TObject;
    Command: Integer; Enable: WordBool);
begin
    case Command of
        CSC_NAVIGATEBACK: begin
            // Зберігає CSC_NAVIGATEBACK положення для Webbrowser
            TWebbrowser(Sender).FNavBack := Enable;
            ToolbtnBack.Enabled := Enable;
        end;
        CSC_NAVIGATEFORWARD: begin
            // Зберігає CSC_NAVIGATEFORWARD положення для Webbrowser
            TWebbrowser(Sender).FNavForward := Enable;
            ToolBtnForward.Enabled := Enable;
        end;
    end;
end;

{ ***** }
// Заголовок форми події OnResize.

procedure TfrmMyBrowser.FormResize(Sender: TObject);
var
    r: TRect;
const
    SB_GETRECT = WM_USER + 10;
begin
    // Встановлення позиції зміни
    StatusBar1.Perform(SB_GETRECT, 2, Integer(@R));
    ProgressBar1.Parent := StatusBar1;
    ProgressBar1.SetBounds(r.Left, r.Top, r.Right - r.Left - 5, r.Bottom - r.Top);
    // Зміна розміру панелі
    with StatusBar1 do
    begin
        Panels[1].Width := Width div 2 - 10;
        Panels[0].Width := 80;
        Panels[2].Width := 90;
        Panels[3].Width := 25;
        Panels[4].Width := 150;
        Refresh;
    end;
end;

function TfrmMyBrowser.DrawZoneIcon(WB: TWebbrowser): TIcon;
var
    ZoneAttr: TZoneAttributes;
    ZoneIcon: TIcon;
begin
    ZoneAttr := GetZoneAttributes(WB.LocationURL);
    ZoneIcon := TIcon.Create;
    try
        if GetZoneIcon(ZoneAttr.szIconPath, ZoneIcon) then
            StatusBar1.Panels[4].Text := ZoneAttr.szDisplayName;
    end;
end;

```

```

    Result := ZoneIcon;
  finally
    //
  end;
end;

{ ***** }
// Виведення іконок

procedure TfrmMyBrowser.StatusBar1DrawPanel(StatusBar: TStatusBar;
  Panel: TStatusPanel; const Rect: TRect);
var
  ZoneIcon: TIcon;
  CurrentWB: TWebbrowser;
begin
  if Panel = StatusBar1.Panels[3] then
  begin
    CurrentWB := GetCurrentWB;
    if Assigned(CurrentWB) then
    begin
      ZoneIcon := DrawZoneIcon(CurrentWB);
      if ZoneIcon.Handle <> 0 then
      begin
        StatusBar1.Canvas.Font.Color := clRed;
        StatusBar1.Canvas.FillRect(Rect);
        StatusBar1.Canvas.Draw(Rect.Left, Rect.Top, ZoneIcon);
      end;
      ZoneIcon.Free;
    end else
      StatusBar1.Panels[3].Text := '';
  end;
end;

{ ***** }
// показує кнопку статусу

procedure TfrmMyBrowser.WebBrowser1StatusTextChange(Sender: TObject;
  const Text: WideString);
var
  CurrentWB: TWebbrowser;
begin
  CurrentWB := GetCurrentWB;
  if Assigned(CurrentWB) then
  begin
    if TControl(CurrentWB).Hint = TControl(Sender).Hint then
      StatusBar1.Panels[1].Text := Text;
  end;
end;

{ ***** }
// OnDownloadComplete відбувається коли дія навігації завершена, зупинена або
отримало невдачу.

procedure TfrmMyBrowser.WebBrowser1DownloadComplete(Sender: TObject);
begin
  WB_Set3DBorderStyle(GetCurrentWB, not chk3DBorderStyle.Checked);
  WB_ShowScrollBar(GetCurrentWB, chkNoScrollBars.Checked);
  StatusBar1.Repaint;
  ToolBtnStop.Enabled := False;
end;

{ ***** }
// Встановлює наближення

procedure TfrmMyBrowser.Zoom1Click(Sender: TObject);
var
  CurrentWB: TWebbrowser;
  ZoomIndex: TWBFontSize;
begin

```

```

CurrentWB := GetCurrentWB;
if WB_DocumentLoaded(CurrentWB) then
begin
  ZoomIndex := WB_GetZoom(CurrentWB);
  case ZoomIndex of
    4: Largest1.Checked := True;
    3: Large1.Checked := True;
    2: Medium1.Checked := True;
    1: Small1.Checked := True;
    0: Smallest1.Checked := True;
  end;
end;
end;

procedure TfrmMyBrowser.Smallest1Click(Sender: TObject);
var
  CurrentWB: TWebbrowser;
  k: Integer;
begin
  CurrentWB := GetCurrentWB;
  if WB_DocumentLoaded(CurrentWB) then
  begin
    // встановлює наближення
    WB_SetZoom(CurrentWB, TMenuItem(Sender).Tag);
    // знімає значення наближення
    for k := 0 to Zoom1.Count - 1 do
      Zoom1.Items[k].Checked := False;
    // встановлює поточний розмір наближення
    TMenuItem(Sender).Checked := True;
  end;
end;

{ ***** }

procedure TfrmMyBrowser.PageControlChanging(Sender: TObject;
  var AllowChange: Boolean);
begin
  // Get the Previous (Current) Webbrowser
  with PageControl do
    if ActivePage.ControlCount > 0 then
      if ActivePage.Controls[0] is TWebbrowser then
        FPrevBrowser := GetCurrentWB;
end;

{ ***** }

procedure TfrmMyBrowser.PageControlChange(Sender: TObject);
var
  CurrentWB: TWebbrowser;
begin
  // встановлює Toolbutton стан повернення назад
  ToolbtnBack.Enabled := False;
  ToolBtnForward.Enabled := False;

  with PageControl do
    if ActivePage.ControlCount > 0 then
      if ActivePage.Controls[0] is TWebbrowser then
        begin
          CurrentWB := TWebbrowser(ActivePage.Controls[0]);
          WB_SetFocus(CurrentWB);
          ToolbtnBack.Enabled := CurrentWB.FNavBack;
          ToolBtnForward.Enabled := CurrentWB.FNavForward;
          edURL.Text := CurrentWB.LocationURL;
          Self.Caption := TWebbrowser(CurrentWB).FTitle;
        end;

    if PageControl.ActivePage.Caption = 'Document Info' then
      PageControlInfosChange(nil);
end;

```

```

{ ***** }

// Отримує ім'я веб-браузеру

procedure WB_GetFrames(WB: TWebbrowser; sl: TStrings);

function EnumProc(AHtmlDocument: IHtmlDocument2; Data: Integer): Boolean;
begin
    frmMyBrowser.lbFrames.Items.Add(AHtmlDocument.url);
    Result := True;
end;

begin
    frmMyBrowser.lbFrames.Clear;
    EnumFrames(WB.ControlInterface.Document as IHtmlDocument2, @EnumProc,
Integer(frmMyBrowser));

    frmMyBrowser.lbFrames.Items.Delete(0);
end;

{ ***** }

procedure TfrmMyBrowser.PageControlInfosChange(Sender: TObject);
begin
    if Assigned(FPrevBrowser) then
        if Assigned(FPrevBrowser.Document) then
            begin
                case PageControlInfos.ActivePageIndex of
                    // показує код
                    0: SourceEdit.Text :=
WB_GetDocumentSourceToString(FPrevBrowser.Document);
                    // показує текст
                    1: WB_GetPlainText(FPrevBrowser, PlainTextEdit.Lines);
                    // Показує імена посилань
                    2: WB_getLinks(FPrevBrowser, LbLinks.Items);
                    // показує імена фреймів
                    3: WB_GetFrames(FPrevBrowser, LbFrames.Items);
                    // показує посилання на картинки
                    4: WB_GetImages(FPrevBrowser, LbImages.Items);
                    // показує імена полів
                    5: Wb_GetFields(FPrevBrowser, LbFields.Items);
                    // показує елементи
                    6: WB_GetObjectView(TreeView1, FPrevBrowser);
                end;
            end;
end;

{ ***** }
// додає HTML код до документу веб-браузеру

procedure TfrmMyBrowser.AddHTML1Click(Sender: TObject);
begin
    frmAddHTML := TfrmAddHTML.Create(Self);
    try
        frmAddHTML.ShowModal;
    finally
        frmAddHTML.Free;
    end;
end;

{ ***** }
// Заголовок OnNewWindow. Створює нову форму

procedure TfrmMyBrowser.WebBrowser1NewWindow2(Sender: TObject;
var ppDisp: IDispatch; var Cancel: WordBool);
var
    ts: TTabSheet;
begin

```

```

AddEventLog('OnNewWindow');
// Якщо документ веб-браузеру повністю не завантажений, то це висвічується
if TWebbrowser(Sender).ReadyState <> READYSTATE_COMPLETE then
begin
  ppDisp := nil;
  Cancel := True;
end else
begin
  ts := CreateTabBrowser('', 'New Browser');
  WaitForBrowser(TWebbrowser(ts.Controls[0]));
  ppdisp := (TWebbrowser(ts.Controls[0])).Application;
  WaitForBrowser(TWebbrowser(ts.Controls[0]));
  // створює новий приклад додатку веб-браузеру:
  // NewWindow := TfrmMyBrowser.Create(self);
  // NewWindow.Show;
  // ppDisp := NewWindow.GetCurrentWB.DefaultDispatch;
end;
end;

{ ***** }
// Встановлює документ розробки режиму

procedure TfrmMyBrowser.DesignModelClick(Sender: TObject);
var
  HTMLDocument2: IHTMLDocument2;
  DesignMode: string;
  CurrentWB: TWebbrowser;
begin
  CurrentWB := GetCurrentWB;
  if Assigned(CurrentWB) then
  begin
    DesignModel.Checked := not DesignModel.Checked;
    HTMLDocument2 := (CurrentWB.Document as IHTMLDocument2);
    if HTMLDocument2 <> nil then begin
      DesignMode := HTMLDocument2.get_designMode;
      if DesignMode = 'On' then
      begin
        // залишає документ розробки режиму
        HTMLDocument2.designMode := 'Off';
        DesignModel.Checked := False;
      end
      else
      begin
        // встановлює документ розробки режиму
        HTMLDocument2.designMode := 'On';
        DesignModel.Checked := True;
      end;
    end;
  end;
end;

{ ***** }
// Збереження документу веб-браузеру

procedure TfrmMyBrowser.Timer1Timer(Sender: TObject);
var
  MausPos: TPoint;
  Element: IHTMLDocument2;
  imgElement: IHTMLImageElement;
  scrElement: IHTMLLinkElement;
  x, y: Integer;
  Doc: IHTMLDocument2;
  CurrentWB: TWebbrowser;
begin
  if chkShowElementInfo.Checked then
  begin
    begin
      CurrentWB := GetCurrentWB;
      if Assigned(CurrentWB) then

```

```

begin
  Doc := CurrentWB.Document as IHTMLDocument2;
  if Assigned(doc) then
  begin
    // отримує поточну позицію мишки
    GetCursorPos (MausPos);
    // перетворення екраних координат в координати клієнта
    mauspos := CurrentWB.ScreenToClient (mauspos);
    x := MausPos.x;
    y := MausPos.y;
    // Отримання елементів під курсором мишки
    Element := GetElementAtPos (doc, x, y);
    if Assigned(Element) then
    begin
      // елемент властивостей
      edElementTag.Text := Element.tagName;
      edElementId.Text := Element.id;
      edElementInnerText.Text := Element.innerText;
      edElementinnerHTML.Text := Element.innerHTML;
      edElementclassName.Text := Element.className;
      Element.QueryInterface (IHTMLIMGElement, imgElement);
      if assigned(imgElement) then
        edElementhref.Text := imgElement.href;

      Element.QueryInterface (IHTMLLinkElement, scrElement);
      if assigned(scrElement) then
        edElementhref.Text := scrElement.href;
    end;
  end;
end;
end;
end;
end;
end;

{ ***** }
// створення динамічного TWebbrowser

function TfrmMyBrowser.CreateTabBrowser(sURL, sCaption: string): TTabSheet;
var
  ts: TTabSheet;
  WB: TWebbrowser;
begin
  // Створення нового TabSheet
  ts := TTabSheet.Create(PageControl);
  Result := ts;
  try
    // Додання властивостей TTabSheet
    ts.PageControl := PageControl;
    ts.Parent := PageControl;
    ts.Caption := sCaption;
    ts.PageIndex := PageControl.ActivePageIndex + 1;

    // створення прикладу TWebbrowser
    WB := TWebbrowser.Create (ts);

    // додаємо TWebbrowser в TTabSheet
    TControl(WB).Parent := ts;
    // або: Ts.InsertControl(WB);

    // додавання властивостей веб-браузера
    WB.Align := alClient;
    WB.Silent := True;
    WB.Visible := True;

    PageControl.ActivePage := ts;

    // додавання подій веб-браузера
    WB.OnProgressChange := WebBrowser1ProgressChange;
    WB.OnStatusTextChange := WebBrowser1StatusTextChange;
  
```

```

WB.OnTitleChange := WebBrowser1TitleChange;
WB.OnNewWindow2 := WebBrowser1NewWindow2;
WB.OnCommandStateChange := WebBrowser1CommandStateChange;
WB.OnDownloadComplete := WebBrowser1DownloadComplete;
WB.OnDocumentComplete := WebBrowser1DocumentComplete;
WB.OnDownloadBegin := WebBrowser1DownloadBegin;
WB.FNavForward := False;
WB.FNavBack := False;

// Навігація по URL
if Trim(sURL) <> '' then
begin
    WB.Navigate(sURL);
    WaitForBrowser(WB);
end;
except
    ts.Free;
end;
end;

{ ***** }
// показ контекстне меню для Tabs

procedure TfrmMyBrowser.PageControlMouseDown(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var
    XHitTestInfo: TTCHitTestInfo;
    HitIndex: Integer;
begin
    XHitTestInfo.pt := Classes.POINT(X, Y);
    HitIndex := TabCtrl_HitTest(PageControl.Handle, @XHitTestInfo);
    if HitIndex <> -1 then
        with PageControl do
            begin
                // показує контекстне меню, тільки, якщо натиснута права кнопка мишки
                if Button = mbRight then
                    begin
                        ActivePageIndex := HitIndex;
                        CloseTab1.Enabled := (ActivePage.Tag <> 2) and (ActivePageIndex <> 0);
                        DuplicateTab1.Enabled := ActivePage.Controls[0] is TWebbrowser;
                        if ActivePage.Tag <> 2 then
                            PCPopup.Popup(Mouse.CursorPos.X, Mouse.CursorPos.Y);
                    end;
                end;
            end;
end;

{ ***** }
// закриття поточного Tab

procedure TfrmMyBrowser.CloseTab1Click(Sender: TObject);
begin
    // не закриваємо перший tab
    if PageControl.ActivePageIndex > 0 then
        begin
            if PageControl.ActivePage.Controls[0] is TWebbrowser then
                begin
                    // звільняється Webbrowser
                    (TWebbrowser(PageControl.ActivePage.Controls[0])).Free;
                    FPrevBrowser := nil;
                    // звільняється поточний Tab
                    PageControl.ActivePage.Free;
                end;
            end;
        end;
end;

{ ***** }
// дублювання поточного Tab

procedure TfrmMyBrowser.DuplicateTab1Click(Sender: TObject);

```

```

var
  iCurrTabIndex: Integer;
  ts: TTabSheet;
  CurrentWB, NewWB: TWebbrowser;
  sCurrURL: string;
begin
  CurrentWB := GetCurrentWB;
  if CurrentWB <> nil then
  begin
    // отримуємо поточний індекс Tab
    iCurrTabIndex := PageControl.ActivePage.TabIndex;
    // отримуємо поточний URL
    sCurrURL := TWebbrowser(CurrentWB).LocationURL;
    // створюємо новий tab
    ts := CreateTabBrowser(sCurrURL, 'New Browser');
    // посилення на нову зсилку
    NewWB := TWebbrowser(ts.Controls[0]);
    // чекаємо поки завантажиться
    WaitForBrowser(NewWB);
    // дублюємо вміст
    if NewWB.Document <> nil then
      (NewWB.Document as IHTMLDocument2).body.innerHTML := (CurrentWB.Document
as IHTMLDocument2).body.innerHTML;
    // встановлюємо індекс сторінки у новий tab
    ts.PageIndex := iCurrTabIndex + 1;
    PageControl.ActivePageIndex := ts.PageIndex;
  end;
end;

{ ***** }
// створюємо подію OnDocumentComplete

procedure TfrmMyBrowser.WebBrowser1DocumentComplete(Sender: TObject;
  const pDisp: IDispatch; var URL: OleVariant);
var
  CurWebBrowser: IWebBrowser;
  TopWebBrowser: IWebBrowser;
  Document: OleVariant;
  WindowName: string;
begin
  CurWebBrowser := pDisp as IWebBrowser;
  TopWebBrowser := TWebBrowser(Sender).DefaultInterface;
  if CurWebBrowser = TopWebBrowser then
  begin
    AddEventLog('OnDocumentComplete ' + URL);
    // скидаємо позицію кнопки прогресу
    ProgressBar1.Position := 0;
    StatusBar1.Panels[0].Text := '';
    ToolBtnStop.Enabled := False;
  end else
  begin
    Document := CurWebBrowser.Document;
    WindowName := Document.ParentWindow.Name;
    AddEventLog(Format('Frame "%s" loaded', [WindowName]));
  end;
end;

{ ***** }
// Показуємо/ховаємо елемент

procedure TfrmMyBrowser.chkShowElementInfoClick(Sender: TObject);
begin
  PnlElementInfo1.Visible := chkShowElementInfo.Checked;
end;

{ ***** }
// OnDownloadBegin відбувається, коли починається навігація.

```

```

procedure TfrmMyBrowser.WebBrowser1DownloadBegin(Sender: TObject);
begin
  AddEventLog('OnDownloadBegin');
end;

{ ***** }
// OnBeforeNavigate2 відбувається, коли управління веб-браузера збирається
// переходити на URL.

procedure TfrmMyBrowser.WebBrowser1BeforeNavigate2(Sender: TObject;
  const pDisp: IDispatch; var URL, Flags, TargetFrameName, PostData,
  Headers: OleVariant; var Cancel: WordBool);
begin
  lvEventLog.Clear;
  AddEventLog('OnBeforeNavigate2 ' + URL);
  ToolBtnStop.Enabled := True;
end;

{ ***** }
// OnNavigateComplete2 відбувається після переходу на нове посилання.

procedure TfrmMyBrowser.WebBrowser1NavigateComplete2(Sender: TObject;
  const pDisp: IDispatch; var URL: OleVariant);
begin
  AddEventLog('OnNavigateComplete2 ' + URL);
  edURL.Text := TWebbrowser(Sender).LocationURL;
end;

{ ***** }
// показує пусту сторінку

procedure TfrmMyBrowser.N9Click(Sender: TObject);
begin

end;

{ ***** }
// для введення/повернення ключів

procedure TfrmMyBrowser.ApplicationEvents1Message(var Msg: tagMSG;
  var Handled: Boolean);
const
  StdKeys = [VK_TAB, VK_RETURN]; { стандартні ключі }
  ExtKeys = [VK_DELETE, VK_BACK, VK_LEFT, VK_RIGHT]; { розширені ключі }
  fExtended = $01000000; { прапор розширених ключів }
var
  CurrentWB: TWebbrowser;
begin
  CurrentWB := GetCurrentWB;
  // вихід, якщо неможливо повернути об'єкт веб-браузеру
  try
    if (CurrentWB = nil) or (CurrentWB.Document = nil) then
      begin
        Handled := False;
        Exit;
      end;
    except
    end;

    // заборона контекстне меню
    if chkDisablePopupMenu.Checked then
      if (Msg.Message = WM_RBUTTONDOWN) or (Msg.Message = WM_RBUTTONDBLCLK) then
        begin
          if IsChild(CurrentWB.Handle, Msg.hwnd) then
            begin
              // показує контекстне меню
              Handled := True;
              Exit;
            end;
          else
            begin
              Handled := False;
              Exit;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

    end;
end;

// handle Ctrl-N
if (GetKeyState(VK_CONTROL) < 0) and (Msg.Message = WM_KEYDOWN) and
    (Msg.wParam = Ord('N')) then
begin
    Handled := True;
    DuplicateTabClick(nil);
    Exit;
end;

if IsChild(CurrentWB.Handle, Msg.Hwnd) then
begin
    if (Msg.Message = WM_CLOSE) then
        msg.message := 0
    else
        if ((Msg.Message >= WM_KEYFIRST) and (Msg.Message <= WM_KEYLAST)) and
            ((Msg.wParam in StdKeys) or (GetKeyState(VK_CONTROL) < 0) or
            (Msg.wParam in ExtKeys) and ((Msg.lParam and fExtended) = fExtended))
        then
            begin
                Handled := (CurrentWB.Application as
                IOleInPlaceActiveObject).TranslateAccelerator(Msg) = S_OK;
                if not Handled then
                    begin
                        Handled := True;
                        TranslateMessage(Msg);
                        DispatchMessage(Msg);
                    end;
            end;
        end;
end;

end;

procedure TfrmMyBrowser.NllClick(Sender: TObject);
begin
    Form_about.Show;
end;

initialization
    // заборона Ctrl+C/Ctrl+V та drag&drop в веб-браузері
    OleInitialize(nil);
    // охороняє від помилок з плаваючою точкою
    Set8087CW($133F);

finalization
    OleUninitialize

end.

```

Файл uAddHTML.pas - створення та додавання html-коду

```

unit uAddHTML;

interface

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

type
  TfrmAddHTML = class(TForm)
    Memol: TMemo;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmAddHTML: TfrmAddHTML;

implementation

{$R *.dfm}

uses
  MSHTML, ActiveX, SHDocVw, Unit1;

procedure TfrmAddHTML.Button2Click(Sender: TObject);
var
  WebDoc: HTMLDocument;
  WebBody: HTMLBody;
begin
  WebDoc := frmMyBrowser.WebBrowser1.Document as HTMLDocument;
  WebBody := WebDoc.body as HTMLBody;
  WebBody.insertAdjacentHTML('BeforeEnd', Memol.Text);
end;

procedure WB_LoadHTML(WebBrowser: TWebBrowser; HTMLCode: string);
var
  sl: TStringList;
  ms: TMemoryStream;
begin
  WebBrowser.Navigate('about:blank');
  while WebBrowser.ReadyState < READYSTATE_INTERACTIVE do
    Application.ProcessMessages;

  if Assigned(WebBrowser.Document) then
  begin
    sl := TStringList.Create;
    try
      ms := TMemoryStream.Create;
      try
        sl.Text := HTMLCode;
        sl.SaveToStream(ms);
        ms.Seek(0, 0);
        (WebBrowser.Document as
IPersistStreamInit).Load(TStreamAdapter.Create(ms));
      finally
        ms.Free;
      end;
    end;
  end;
end;

```

```
        end;  
    finally  
        sl.Free;  
    end;  
end;  
  
procedure TfrmAddHTML.Button3Click(Sender: TObject);  
begin  
    WB_LoadHTML(frmMyBrowser.GetCurrentWB, Memo1.Text);  
end;  
  
end.
```

Кафедра _ КБПЗ _ 2022 рік

Файл about.pas основної програми

```
unit about;

interface

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

type
  TForm_about = class(TForm)
    Mem01: TMemo;
    Image1: TImage;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form_about: TForm_about;

implementation

{$R *.dfm}

procedure TForm_about.BitBtn1Click(Sender: TObject);
begin
  Form_about.Close;
end;

end.
```