

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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи генерації
складних двомірних матричних штрих-кодів”

Виконав здобувач вищої освіти
II курсу, групи КН-22М-2
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Батрак К.М.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук
_____ Улічев О.С.
« ____ » _____ 2023 р.
Рецензент _____

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Батраку Костянтину Миколайовичу

(прізвище, ім'я, по батькові)

- Тема роботи *Дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів*
- Керівник роботи *Улічев Олександр Сергійович, канд. техн. наук*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 33-13 від 04.08.2023 року
- Строк подання студентом роботи до захисту *10.12.2023 р.*
- Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів*
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.*
 - Перегляд аналогічних існуючих систем.*
 - Опис і обґрунтування проектних рішень.*
 - Етапи програмування системи.*
 - Впровадження системи в промислову експлуатацію*
 - Наукова новизна.*
 - Економічна ефективність розробленої програми.*
 - Заходи з охорони праці та техніки безпеки.*
 - Висновки.*
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

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

6. Консультанти розділів роботи

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

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

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

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

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Батрак К.М. Дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи генерації складних двомірних матричних штрих-кодів.

Метою розробки є дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів.

Об'єктом дослідження є процес генерації складних двомірних матричних штрих-кодів.

Предметом дослідження є методи генерації складних двомірних матричних штрих-кодів.

Методи дослідження базуються на методах обробки зображень, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи генерації складних двомірних матричних штрих-кодів.

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

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

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

Програму розроблено в середовищі Delphi 10.4 Sydney.

Ключові слова: комп'ютерні науки, штрих-коди

ABSTRACT

Batrak K.M. Research and software implementation of the complex two-dimensional matrix barcode generation system. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the system of generating complex two-dimensional matrix barcodes.

The purpose of the development is the research and software implementation of the system for generating complex two-dimensional matrix barcodes.

The object of research is the process of generating complex two-dimensional matrix barcodes.

The subject of research is methods of generating complex two-dimensional matrix barcodes.

Research methods are based on image processing methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the complex two-dimensional matrix barcode generation system.

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

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

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

The program was developed in the Delphi 10.4 Sydney environment.

Keywords: computer science, barcodes

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.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	14
2.3 Розгорнута постановка завдання	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	21
3.1 Опис функціонування системи	21
3.2 Розробка структурної схеми.....	28
3.3 Розробка функціональної схеми	35
3.4 Розробка діаграми процесів.....	37
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	39
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	39
4.2 Захист розробленого програмного забезпечення.....	48
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	51
6 НАУКОВА НОВИЗНА	54

						ВКРМ-122.23.0031.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Батрак К.М.				Дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів	Літ.	Аркуш	Аркушів
Перев.	Улічев О.С.					М	1	95
Н.контр.	Коваленко А.С.				ЦНТУ КН-22М-2			
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	55
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	55
7.2 Розрахунок трудомісткості розробки програмної продукції.....	57
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	59
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	64
7.5 Визначення собівартості розробки та ціни програмної продукції.....	68
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	71
7.7 Визначення експлуатаційних витрат.....	71
7.8 Визначення економічної ефективності програмної продукції.....	73
7.9 Висновок.....	75
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	76
8.1 Вступ.....	76
8.2 Аналіз умов праці на робочому місці ІТ-фахівця.....	78
8.3 Пропозиції щодо підвищення працездатності ІТ-фахівця.....	81
8.4 Пожежна безпека.....	82
8.5 Розрахункова частина	85
9 ОСНОВНІ ВИСНОВКИ.....	87
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	89

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

OCR	–	Optical Character Recognition
QR	–	quick response – швидкий відгук
КПК	–	кишеньковий персональний комп'ютер
ПЗ	–	програмне забезпечення

КБПЗ-2023

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. QR означає «Швидка відповідь». Хоча вони можуть виглядати просто, QR-коди здатні зберігати багато даних. Але незалежно від того, скільки вони містять, після сканування QR-код повинен дозволити користувачеві миттєво отримати доступ до інформації – тому він називається кодом швидкого реагування. QR-код – це тип штрих-коду, який легко зчитується цифровим пристроєм і який зберігає інформацію у вигляді серії пікселів у квадратній сітці. QR-коди часто використовуються для відстеження інформації про продукти в ланцюжку постачання, а оскільки багато смартфонів мають вбудовані зчитувачі QR-кодів, їх часто використовують у маркетингових і рекламних кампаніях. Зовсім недавно вони зіграли ключову роль у допомозі відстежувати зараження коронавірусом і уповільнювати поширення вірусу.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів.

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

- Огляд існуючих систем генерації складних двомірних матричних штрих-кодів.
- Дослідження системи генерації складних двомірних матричних штрих-кодів.
- Програмна реалізація системи генерації складних двомірних матричних штрих-кодів.

Об'єктом дослідження є процес генерації складних двомірних матричних штрих-кодів.

Предметом дослідження є методи генерації складних двомірних матричних штрих-кодів.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Методи дослідження базуються на методах обробки зображень, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод генерації складних двомірних матричних штрих-кодів.
- Розроблено вітчизняний продукт генерації складних двомірних матричних штрих-кодів, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі генерації складних двомірних матричних штрих-кодів.

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

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

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

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Стандартні штрих-коди можна зчитувати лише в одному напрямку – зверху вниз. Це означає, що вони можуть зберігати лише невелику кількість інформації, зазвичай у буквено-цифровому форматі. Але QR-код зчитується у двох напрямках – зверху вниз і справа наліво. Це дозволяє зберігати значно більше даних.

Команда розробників QR-коду хотіла зробити код легким для сканування, щоб оперативники не витрачали час, щоб отримати його під потрібним кутом. Вони також хотіли, щоб він мав характерний дизайн, щоб його було легко ідентифікувати. Це спонукало їх вибрати знакову квадратну форму, яка використовується досі.

Denso Wave оприлюднила свій QR-код і заявила, що не буде використовувати свої патентні права. Це означало, що кожен міг створювати та використовувати QR-коди.

Початкове сприйняття ідеї було повільним; однак у 2002 році в Японії з'явилися перші мобільні телефони з вбудованими зчитувачами QR. Використання смартфонів призвело до збільшення кількості компаній, які використовують QR-коди.

У 2020 році Denso Wave продовжила вдосконалювати свій оригінальний дизайн. Їхні нові QR-коди включають відстеження, захист бренду та засоби захисту від підробки. Є багато нових способів використання QR-коду, від переказу платежів до визначення положення об'єктів у доповненій реальності.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1.2 Область застосування

Дані, що зберігаються в QR-кодi, можуть включати URL-адреси веб-сайтів, номери телефонів або до 4000 символів тексту. QR-коди також можна використовувати для:

- Пряме посилання для завантаження програми в Apple App Store або Google Play.
- Автентифікуйте онлайн-акаунти та перевіряйте дані для входу.
- Отримайте доступ до Wi-Fi, зберігаючи дані шифрування, такі як SSID, пароль і тип шифрування.
- Надсилання та отримання платіжної інформації.
- І багато іншого – компанія у Великій Британії під назвою QR Memories навіть створює QR-коди для використання на надгробках, дозволяючи людям сканувати код, щоб дізнатися більше про життя цієї померлої людини (якщо у них є некролог чи новини, пов'язані з нею в Інтернеті).

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

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

ShopSavvy

ShopSavvy являє собою додаток для сканування QR-кодів, що дозволяє користувачам легко порівнювати ціни на товари в прилеглих магазинах у режимі онлайн, щоб вибрати найкращі пропозиції. Для цього всього лише потрібно направити лазерний сканер додатка на штрихкод товару й навіть без натискання кнопки сканер QR-коду автоматично сканує UPC-код (універсальний код продукту) і ідентифікує товар. Я використовував цей додаток для порівняння цін у режимі онлайн, однак, на жаль, деякі товари, які я сканував, не були присутні в інших магазинах у моєму районі. Проте, у відношенні інших продуктових товарів функція пошуку аналогічного товару в сусідніх магазинах спрацювала ефективно. Сканер QR-кодів Shop Savvy не зміг ідентифікувати деякі товари, які я сканував, однак мені було запропоновано додати їх в онлайнкову бібліотеку ShopSavvy для того, щоб інші користувачі могли користуватися цією інформацією. Даний додаток має додаткову функцію “Deals”, що дозволяє знайти прилеглі магазини й ознайомитися із пропонованими ними знижками, а також погасити електронні купони в Інтернет-магазині.

Незважаючи на те, що функція сканування коду UPC додатка ShopSavvy працювала без збоїв, у мене виникли невеликі складності з використанням сканери QR-кодів. Один з QR-кодів, які я намагалася сканувати, так і не завантажився. При спробі сканувати інший код, була зазначена адреса якогось веб-сайту для самостійного одержання інформації без використання додатка. При третій спробі веб-сайт легко завантажився й з'явився в спливаючому вікні.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

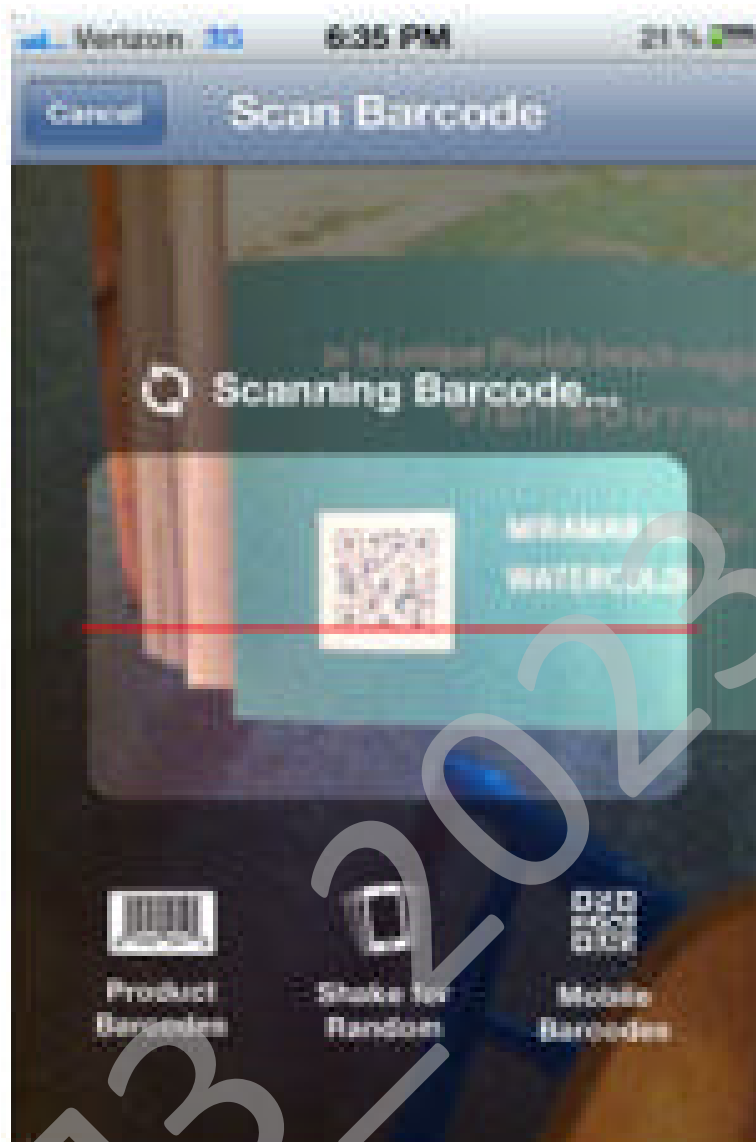


Рисунок 2.1 – Інтерфейс користувача ShopSavvy

Простим натисканням кнопки “back” у лівому верхньому куті я знову повернувся в додаток ShopSavvy, що досить зручно для користувача. У цілому, можна сказати, що функції сканування UPC у цьому додатку просто чудові, однак при використанні сканера QR-кодів, як видно, необхідно дуже точно й обережно наводити його на код і тримати його у фіксованому положенні, що може бути досить незручно для користувачів при пересуванні по магазині.

Я настійно рекомендую додаток ShopSavvy покупцям, що прагнуть робити найбільш вигідні покупки як в Інтернеті, так і в прилеглих магазинах. Функція

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

“history” спрощує систематизацію інформації про колись скановані товари й, якщо ви відкриєте рахунок ShopSavvy за допомогою цього додатка, ви зможете робити покупки за найкращими цінами через Інтернет. Сканер прекрасно функціонує при скануванні штрихкодів товарів, однак при скануванні QR-кодів він далеко не так ефективний. Проте, можна затверджувати, що цей додаток скоріше призначене для ощадливих людей, що прагнуть робити вигідні покупки, а дана функція працює безвідмовно.

QRganize

Додаток QRganize для сканування й систематизації QR-кодів має прекрасний сканер QR-кодів, що, на мою думку, відрізняється просто ідеальною чутливістю. Він легко сканував QR-коди на товарах у магазині, і запропонував або перейти на відповідний сайт по посиланню, або зберегти його у відповідній категорії в бібліотеці QRganize у додатку. Категорії вже наявні в додатку бібліотеки цілком відповідають моїм потребам відносно систематизації інформації, однак цей додаток дозволяє користувачам самим створювати категорії залежно від своїх цілей. Користувачі додатка можуть додавати обрані категорії у свої особисті бібліотеки й систематизувати скановану інформацію зі свого смаку. Пропонується просто незліченне число готових категорій. Ви також можете ділитися інформацією про QR-коди з користувачами Twitter або Facebook лише одним натисканням кнопки. Ця функція дуже зручна для всіх аматорів спілкування в соціальних мережах. Незважаючи на те, що за допомогою даного додатка мені вдалося успішно сканувати більшість QR-кодів, з одним з них сканеру QR-коду додатка QRganize не вдалося впоратися. Цілком можливо, що це відбулося з вини рекламодавця, тому що іконка коду на екрані мобільного телефону, на мій погляд, виглядала досить розпливчато і її розміри були перевернуті, у результаті чого, незалежно від відстані, з якого я намагалася неї сканувати, додаток відмовлялося зчитувати цей QR-код.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10



Рисунок 2.2 – Інтерфейс користувача QRganize

Додаток QRganize направляє користувачів безпосередньо на відповідний веб-сайт за допомогою власного браузера, що полегшує повернення в основне меню за допомогою кнопки “Preview” у верхньому лівому куті. У безкоштовній версії сканера QR-кодів, додатка QRganize, що я завантажив з Інтернет-магазину iPhone App Store, дійсно є невеликі ненав'язливі рекламні повідомлення в нижній частині екрана. Їх легко можна закрити й вони практично не заважають операціям, які робить користувач.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Мені подобається, яким образом організована систематизація інформації в сканері QR-кодів додатка QRganize. Це назва прекрасно підходить для даної прикладної програми. Мені не вдалося сканувати що не читається QR-код лише один раз і я вважаю, що даний додаток є вкрай зручним і простим у використанні. Навіть користувач, що не володіє більшим досвідом роботи з додатками, може легко освоїти сканери QR-кодів, завантажуючи й використовуючи QRganize.

QRReader

Додаток QRReader являє собою базовий сканер QR-кодів, що не володіє функцією сканування універсальних кодів продуктів або штрихкодів. Проте, відсутність функції сканування штрихкодів компенсується можливістю створення QR-кодів відповідно до власних потреб. Я успішно створив й представив на своїй сторінці в Facebook QR-код з URL-адресою свого персонального веб-сайту, щоб мої друзі могли сканувати його й заходити на сайт. Це додаток також має функцію створення QR для URLs, телефонних номерів, географічних об'єктів, адрес електронної пошти й т.д. Ця функція є справжнім подарунком для користувачів iPhone, тому що дозволяє їм здійснювати діяльність в Інтернеті з використанням самих передових цифрових технологій.

Даний додаток зберігає дані про скановані раніше коди в розділі QR Codes, що дозволяє легко переглянути історію сканування. Ви також можете зберігати певні скановані QR-коди в розділі "favorites" (Вибране). У багатьох відносинах цей сканер QR-кодів може бути ідеальним рішенням для починаючого користувача, тому що він наділений усього двома конкретними функціями: сканування й створення персональних QR-кодів. На ринку прикладних програм, заповненому додатками з безліччю функцій, додаток QRReader iPhone виділяється простотою й привабливо для користувачів завдяки своїй практичності, легкості у використанні й відсутності зайвих функцій.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12



Рисунок 2.3 – Інтерфейс користувача QRReader

Резюме

У цілому кожний із цих трьох додатків відрізняється своїми перевагами, завдяки яким вони можуть стати досить кошовним конкурентоспроможним товаром на ринку сканерів QR-кодів. Я даю високу оцінку сканеру QR-кодів ShopSavvy, тому що цей додаток забезпечує економію засобів при здійсненні покупок. Мені також сподобалася можливість систематизації інформації з категорій, що надає додаток QRganize. А QRReader пропонує базові функції користувачам, що не мають досвіду використання сканерів QR-кодів. Я з нетерпінням очікую подальшого розвитку цієї чудової технології для смартфонів.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

Delphi 10.4 Sydney

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

Основні можливості Delphi 10.4.1:

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

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

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

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

– 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-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

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

Єдине керування пам'яттю

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

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

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

Розширена підтримка бібліотек C++

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

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

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими 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-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

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 компонентів на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Поліпшена кроссплатформеність

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

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

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 генерації складних двомірних матричних штрих-кодів.

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-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Незважаючи на бурхливий ріст ринку систем електронного документообігу в різних областях діяльності, включаючи й державне керування, кількість паперових документів і їхніх обсягів не тільки не зменшуються, але й постійно збільшуються. Можна знайти безліч пояснень такому феномену, однак, основною причиною «живучості» паперового документа є його звичність, дешевина й зручність сприйняття. Інакше кажучи, паперовий документ для дуже багатьох є найбільш зрозумілим і «дружнім інтерфейсом взаємодії». До того ж, незважаючи на розвиток технологій електронно-цифрового підпису, паперовий документ як і раніше залишається юридичною основою будь-якої діяльності. Ще один немаловажний момент – це незалежність паперового документа від інформаційних систем їх програмного й технічного забезпечення, а також нехильність паперового документа зараженню комп'ютерними вірусами й проблемами, що впливають їх цього. Незважаючи на перераховані достоїнства паперового документа – він об'єктивно є бар'єром на шляху підвищення ефективності документообігу, що обумовлює, насамперед, високою складністю й затратністю процесу уведення змісту документа в інформаційну систему.

Донедавна поставлене завдання вирішувалося застосуванням технологій автоматичного оптичного розпізнавання тексту із графічних образів (Optical Character Recognition, OCR) і наступного автоматичного уведення розпізаного змісту документа в інформаційну систему. Однак в основі технологій OCR лежать процеси розпізнавання, що носять імовірнісний характер, що не повною мірою задовольняють якості (вірогідності) і швидкості розпізнавання документа.

Більшість смартфонів мають вбудовані QR-сканери, які іноді вбудовані в камеру. QR-сканер – це просто спосіб сканування QR-кодів.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

ініціювати завантаження шкідливого програмного забезпечення.

Мобільні пристрої, як правило, менш безпечні, ніж комп'ютери чи ноутбуки. Оскільки QR-коди використовуються на мобільних пристроях, це збільшує потенційні ризики.

Чи збирають QR-коди мою особисту інформацію та дані?

Програмне забезпечення для створення QR-кодів не збирає особисту інформацію.

Дані, які він збирає, і які бачать розробники коду, включають місцезнаходження, кількість і час сканування коду, а також операційну систему пристрою, який просканував код (наприклад, iPhone або Android)..

Самі QR-коди не можна зламати – ризики безпеці, пов'язані з QR-кодами, походять від місця призначення QR-кодів, а не від самих кодів.

Хакери можуть створювати шкідливі QR-коди, які спрямовують користувачів на підроблені веб-сайти, які збирають їхні особисті дані, наприклад облікові дані для входу, або навіть відстежують їх геолокацію на їхньому телефоні.

Ось чому мобільні користувачі повинні сканувати лише коди, які надходять від надійного відправника.

Візерунки в QR-кодах представляють двійкові коди, які можна інтерпретувати, щоб розкрити дані коду.

QR-зчитувач може ідентифікувати стандартний QR-код на основі трьох великих квадратів поза межами QR-коду. Коли він ідентифікує ці три фігури, він дізнається, що все, що міститься всередині квадрата, є QR-кодом.

Потім QR-зчитувач аналізує QR-код, розбиваючи все на сітку. Він розглядає окремі квадрати сітки та призначає кожному значення залежно від того, чорний чи білий він. Потім він групує квадрати сітки для створення більших візерунків.

Стандартний QR-код можна ідентифікувати на основі шести компонентів:

1. Тиха зона – це порожня біла рамка навколо QR-коду. Без цієї межі QR-

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

зчитувач не зможе визначити, що міститься в QR-кодi, а що ні (через втручання зовнішніх елементів).

2. Шаблон шукача – QR-коди зазвичай містять три чорні квадрати в нижньому лівому, верхньому лівому та верхньому правому кутах. Ці квадрати повідомляють QR-зчитувача, що він дивиться на QR-код і де пролягають зовнішні межі коду.

3. Шаблон вирівнювання - це ще один менший квадрат, який міститься десь біля нижнього правого кута. Це гарантує, що QR-код можна прочитати, навіть якщо він перекошений або під кутом.

4. Шаблон синхронізації - це L-подібна лінія, яка проходить між трьома квадратами в шаблоні шукача. Шаблон синхронізації допомагає зчитувача ідентифікувати окремі квадрати в усьому кодi та дає змогу зчитувати пошкоджений QR-код.

5. Інформація про версію – це невелике поле інформації, яке міститься біля верхньої правої клітинки шаблону шукача. Це визначає, яка версія QR-коду зчитується (див. «Типи QR-коду» нижче).

6. Комірки даних – решта QR-коду передає фактичну інформацію, тобто URL-адресу, номер телефону або повідомлення, яке містить.

Програмне забезпечення для створення QR-кодів не збирає особисту інформацію.

Дані, які він збирає, і які бачать розробники коду, включають місцезнаходження, кількість і час сканування коду, а також операційну систему пристрою, який просканував код (наприклад, iPhone або Android)..

Види QR-коду

QR-коди можна використовувати для багатьох цілей, але існує чотири загальноприйняті версії QR-кодів. Використана версія визначає спосіб зберігання даних і називається «режимом введення». Це може бути числове, буквено-цифрове, двійкове або кандзі. Тип режиму повідомляється через поле інформації про версію в QR-кодi.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

3. Максi-код – цей тип QR-коду використовується поштовою службою США. Він схожий на код Aztec тим, що він розміщує візерунок шукача посередині, але в ньому використовується стільниковий візерунок замість квадратів.

4. PDF417. Винайдений у 1991 році Ynjiun Wang із Symbol Technologies, PDF417 із дивною назвою виник на три роки раніше QR-коду. Він виглядає як суміш QR-коду та штрих-коду, і його легко впізнати за прямокутною формою.

5. Semacode. Розроблений однойменною програмною компанією Semacode – це матриця даних, яка дуже схожа на звичайний QR-код, але не має розпізнаваних шаблонів шукача.

QR-коди використовуються в багатьох контекстах, наприклад:

– QR-коди в продажах і маркетингу. Багато рекламодавців використовують QR-коди у своїх кампаніях, оскільки це забезпечує швидший та інтуїтивно зрозуміліший спосіб спрямовувати людей на веб-сайти, ніж введення URL-адрес вручну. Їх також можна використовувати для прямого посилання на сторінки продукту в Інтернеті. Наприклад, якщо ви шукали саме ту сукню, яку одягла модель на плакаті, QR-код міг би напяму перевести вас на веб-сторінку, де її можна було придбати.

– QR-коди для відстеження коронавірусу. Пандемія коронавірусу посилила використання QR-кодів. Наприклад, у Великій Британії відвідувачам закладів гостинності, таких як бари та ресторани, пропонується сканувати QR-код після прибуття за допомогою програми відстеження Covid-19 NHS. Це допоможе відстежити та зупинити поширення вірусу. Якщо хтось виявляє позитивний результат тесту на Covid-19 у цьому закладі, інші відвідувачі цього місця сповіщаються за допомогою програми завдяки даним, зібраним зі сканованих QR-кодів.

– QR-коди на упаковці товару. Ви також можете знайти QR-коди на упаковках для деяких ваших улюблених продуктів. Ці QR-коди можуть розкривати інформацію про продукт, наприклад інформацію про харчову цінність або спеціальні пропозиції, якими ви можете скористатися під час наступної покупки.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

– QR-коди в промисловості. Спочатку QR-коди були винайдені, щоб допомогти відстежувати деталі при виробництві транспортних засобів, і вони досі використовуються у всій промисловості. Ви також знайдете QR-коди, які використовуються іншими підприємствами, яким потрібно уважно стежити за продуктами та витратними матеріалами, як-от будівництво, машинобудування та роздрібна торгівля.

– QR-коди в поштових службах. Поштові служби в усьому світі також використовують їх. Оскільки вони можуть містити великий обсяг інформації, на них часто покладаються для відстеження посилок. Наприклад, світовий модний бренд ASOS повністю перейшов на QR-коди для відстеження повернення коштів.

– QR-коди в освіті. QR-коди також використовуються в школах і коледжах, щоб допомогти спілкуватися зі студентами. Вони з'являлися скрізь, від класу до бібліотеки, для таких завдань, як допомога учням у пошуку книжок, які вони шукають.

Невідомо, де і коли ви можете натрапити на шкідливий QR-код. Ось чому важливо використовувати QR-сканер, якому можна довіряти, а не завантажувати випадковий сканер із магазину додатків чи онлайн.

QR Scanner миттєво перевіряє безпечність сканованого посилання, перш ніж надіслати вам будь-яку інформацію.

Сканер забезпечує автентифікацію за QR-кодом і попереджає про потенційну небезпеку, що криється за QR-кодом, наприклад:

1. Фішингове шахрайство.
2. Примусове завантаження програми або шахрайство з платними текстовими повідомленнями.
3. Небезпечні посилання.

QR Scanner як і раніше надає все, що вам потрібно від QR-сканера, наприклад додавання контактів у ваш телефон. Він також створює журнал минулих сканувань, щоб у разі потреби ви могли відстежити, коли та де вас могли зламати.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

3.2 Розробка структурної схеми

На рисунку 3.1 зображено розроблену структурну схему системи генерації складних двомірних матричних штрих-кодів типу QR.

Розглянемо схему згори до низу. Спочатку проходить введення даних користувача, а саме даних які будуть вбудовані у QR код. При необхідності проводиться зміна налаштувань генерації QR коду по замовчанню.



Рисунок 3.1 – Структурна схема роботи системи

Далі проводиться кодування даних користувача з використання існуючих налаштувань кодування. Після цього проводиться формування кодів корегування тобто дублювання даних для поміхо стійкості з використанням модулю генерації кодів корегування.

Після цих дій проводиться застосування маски та додавання версії штрих-коду. Версії штрих-коду дозволяє при розпізнаванні QR коду користувачем більш точно провести декодування.

Та на останньому етапі проводиться генерація двомірного матричного штрих-коду QR з використанням модуля створення JPG файлів, що у кінцевому результаті дає повноцінний згенерований QR код.

Ось кілька креативних способів використання QR-кодів для окремих осіб і компаній.

1. Автобусні зупинки та вокзали

QR-коди використовуються на автобусних зупинках і вокзалах у містах по всьому світу. У Лондоні, наприклад, Transport for London розмістив QR-коди на автобусних зупинках, щоб люди могли відсканувати код і отримати інформацію про автобусний маршрут і розклад.

QR-коди також використовуються на вокзалах Японії, щоб надати пасажиром інформацію про їхню подорож. Крім того, країна також використовує QR-коди для забезпечення доступу до квиткових воріт.

2. Візитки

Ще один спосіб, у який компанії використовують QR-коди, – це вставляти їх у свої візитні картки. Це дозволяє людям сканувати код і миттєво додавати контактну інформацію власника бізнесу на свій телефон.

Так само QR-код на візитній картці може перевести людей на веб-сайт компанії або сторінку контактів. За допомогою таких візитних карток компанії можуть зробити свої картки простими, елегантними та інноваційними.

3. Про упаковану їжу

Супермаркети можуть розміщувати QR-коди на упакованих продуктах

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

харчування, що дозволяє клієнтам отримати більше інформації про інгредієнти та рецепти. Покупцеві достатньо піднести свій телефон до QR-коду на упаковці, і він перейде на веб-сайт або додаток із додатковою інформацією.

Це чудовий спосіб для компаній надати клієнтам більше прозорості щодо своїх продуктів харчування. Крім того, це корисний спосіб для покупців дізнатися більше про те, що вони купують і як цим користуватися.

Згідно зі звітом Statista, 57% покупців сканували продукти харчування, щоб отримати більше інформації про них.

4. Опитування клієнтів

Отримання аналітичних даних про клієнтів є важливим для кожного бізнесу, оскільки це допомагає оцінити рівень задоволеності клієнтів і визначити сфери, які потрібно вдосконалити. Традиційно опитування клієнтів проводилися на паперових носіях і вимагали від клієнтів заповнити форму та опустити її в ящик або надіслати назад поштою.

З появою QR-кодів компанії тепер можуть створювати цифрові опитування клієнтів, які легко заповнювати та не вимагають паперу. Ви повинні розмістити QR-код на квитанції, яку ви надаєте клієнту після покупки.

Клієнт сканує цей код і потрапляє на сайт опитування. Потім вони можуть заповнити опитування онлайн.

5. Купони

Компанії роками використовують QR-коди на купонах, щоб відстежувати ефективність своїх маркетингових кампаній. QR-коди полегшують клієнтам активацію купонів, а компаніям – відстеження коефіцієнтів використання купонів.

Щоб використовувати QR-коди для купонів, компаніям потрібно просто створити унікальний код для кожного купона та додати його до самого купону. Потім клієнт може відсканувати код за допомогою свого телефону, щоб скористатися знижкою.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

6. Збільште кількість завантажень програми

Малі підприємства можуть збільшити завантаження додатків за допомогою QR-кодів. Ви можете розмістити QR-код на своєму веб-сайті, у профілях у соціальних мережах або навіть на візитних картках, які посилаються на сторінку вашого додатка в магазині програм. Дослідження показують, що люди сканують QR-коди майже з усіх носіїв.

Коли хтось просканує код, він перейде безпосередньо на сторінку, де зможе завантажити вашу програму. Це швидкий і простий спосіб залучити людей до App Store замість того, щоб розраховувати на те, що вони самі пройдуть через проблеми.

7. Безконтактне меню

Під час пандемії багато ресторанів перейшли на безконтактне меню, щоб зменшити поширення мікробів. Натомість клієнти можуть відсканувати QR-код своїм смартфоном, щоб переглянути меню.

Це чудовий спосіб для ресторанів оновлювати свої меню в режимі реального часу та уникати друку нових меню щоразу, коли відбуваються зміни.

Цікаво те, що експерти галузі не вважають це примхою. Натомість кажуть, що це буде довгострокова зміна. Крім того, звіт про готовність ресторанів показав, що 33% підприємств бачать позитивну відповідь, коли вони використовують QR-коди.

8. Сканувати для оплати

Компанії можуть використовувати QR-коди, щоб дозволити клієнтам здійснювати безконтактні платежі. Клієнти можуть сканувати QR-код за допомогою свого смартфона, а потім використовувати свій мобільний гаманець для завершення платежу.

Це чудовий спосіб для підприємств зменшити ризик розповсюдження мікробів і зробити платіж зручнішим для клієнтів. Крім того, це може допомогти підприємствам відійти від традиційних методів і застосувати інноваційні підходи.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

9. Приєднайтеся до WiFi

Замість того, щоб вводити пароль, клієнти можуть відсканувати QR-код, щоб підключитися до мережі WiFi компанії. Це вигідно для компаній із великою кількістю відвідувачів, таких як кафе, готелі та роздрібні магазини.

Це чудовий спосіб зробити клієнтам зручнішим підключення до Інтернету, не вивчаючи пароль.

10. Інформація про курс

Університети можуть друкувати QR-коди на інформаційних листках курсу, щоб студенти могли швидко та легко отримати доступ до опису курсу, попередніх вимог та іншої важливої інформації.

Це допоможе зменшити роботу та клопоти для адміністративного відділу, особливо під час прийому, коли потрібно багато документів.

Дослідження вже показують, що молоді люди сканують більше QR-кодів, ніж їхні старші колеги. Тому такі ініціативи принесуть величезну користь молодому поколінню.

11. Подарунки

Як бренд, якщо ви хочете влаштувати роздачу, отримати QR-коди в суміші – чудова ідея. Ви можете використовувати QR-код, щоб направити учасників на цільову сторінку, де вони можуть ввести свою інформацію та отримати участь у виграші.

Це чудовий спосіб підвищити впізнаваність бренду та змусити людей говорити про ваш продукт чи послугу. Крім того, ви можете заохочувати людей ділитися QR-кодом зі своїми друзями для додаткових записів.

12. Сорочки та ділові товари

Ви шукаєте інноваційний і дешевий спосіб рекламувати свою останню послугу, щоб збільшити кількість відвідувачів на вашому веб-сайті? Ви можете використовувати QR-коди. 45% покупців кажуть, що сканували маркетинговий код у США.

Додайте на свою футболку QR-код із посиланням на ваш веб-сайт,

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

ваші заходи.

15. Створіть список передплатників електронної пошти

Компанії також можуть використовувати QR-коди для створення списку підписників електронної пошти. Ви можете додати QR-код на свій веб-сайт, профілі в соціальних мережах або навіть фізичні маркетингові матеріали, як-от візитні картки чи листівки. Коли люди сканують QR-код, вони переходять на цільову сторінку, де вони можуть ввести свою електронну адресу, щоб зареєструватися у вашому списку розсилки.

Таким чином ви можете дозволити клієнтам приходити до вас, а не відстежувати їх і запитувати їх контактну інформацію. Ви можете використовувати це на YouTube під час створення аудиторії. Додайте QR-код у свій відеоблог, який веде до провідного магніту, щоб спрямувати ваших глядачів до більшого способу взаємодії з вашим брендом.

16. Дозволи на будівництво

Чи знаєте ви, що Нью-Йорк використовує QR-коди для дозволів на будівництво? Інші міста також починають це робити.

QR-код є посиланням на веб-сайт, де громадяни можуть дізнатися більше про проект будівництва, наприклад, графік, очікуваний рівень шуму та контактну інформацію керівника проекту.

Це інноваційний спосіб для міст забезпечити більшу прозорість і інформувати мешканців про те, що відбувається

17. Резюме та CV

Окремі особи також можуть розміщувати QR-коди у своїх резюме та CV. Коли потенційні роботодавці сканують QR-код, їх можна перевести на веб-сайт, де вони зможуть дізнатися більше про кваліфікацію та досвід людини.

Це чудовий спосіб виділитися серед конкурентів і допомогти роботодавцям дізнатися про вас більше. Наприклад, якщо ви графічний дизайнер, ви можете додати QR-код до свого резюме, який переведе роботодавця на ваш профіль або веб-сайт Behance.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Поради щодо використання QR-кодів

Незалежно від того, чи використовуєте ви QR-коди для бізнесу чи для особистих цілей, ви можете зробити кілька речей, щоб переконатися, що ваш досвід буде позитивним. Ось кілька порад:

- Виберіть генератор QR-кодів, який має репутацію та хороші відгуки.
- Переконайтеся, що QR-код видимий і його можна легко відсканувати.

Якщо він занадто маленький або розмитий, він не працюватиме.

- Перевірте QR-код, щоб переконатися, що він працює, перш ніж використовувати його.

- Обов'язково додайте заклик до дії разом із своїм QR-кодом. Скажіть людям, що їм потрібно зробити після сканування.

- Переконайтеся, що вміст, на який ви посилаетесь, зручний для мобільних пристроїв. Якщо це не так, люди не зможуть належним чином переглядати його на своїх телефонах. У результаті це зруйнує всю мету використання QR-коду.

Підводячи підсумок, ви можете використовувати QR-коди для багатьох цілей, включаючи контент-маркетинг, обмін контактною інформацією та посилання на веб-сайти. QR-коди – це чудовий спосіб поділитися інформацією чи просувати свій бізнес.

Просто не забудьте скористатись надійним генератором QR-кодів і перевірте свій код перед його використанням. І не забудьте включити заклик до дії. Найважливіше те, що ви можете придумати власні інноваційні способи використання QR-кодів

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. На схемі детально розглянуто результат роботи розробленої програми – згенерований QR код. На наведеному прикладі закодовано назву учбового закладу – текст

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

«Центральноукраїнський національний технічний університет» (докладніше у розділі 5).

Для розпізнавання створеного QR коду існує велика кількість сторонніх програм (розглянуті у розділі 2). При отриманні цього коду вони шукають створені області детектування та центрування. Після цього зчитують системну інформацію. Яка представляє собою графічно представлений бітовий масив. Та усікаючи все зайве за допомогою маски отримують кінцеві дані – текст «Центральноукраїнський національний технічний університет».



Рисунок 3.2 – Функціональна схема системи

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

3.4 Розробка діаграми процесів

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

Після початку роботи ми потрапляємо до інтерфейсу ПЗ, звідки можемо перейти до довідкової системи, налаштування ПЗ, кодування даних користувача.

Далі проводиться формування кодів корегування з використанням налаштувань параметрів генерації та модуля генерації кодів корегування.

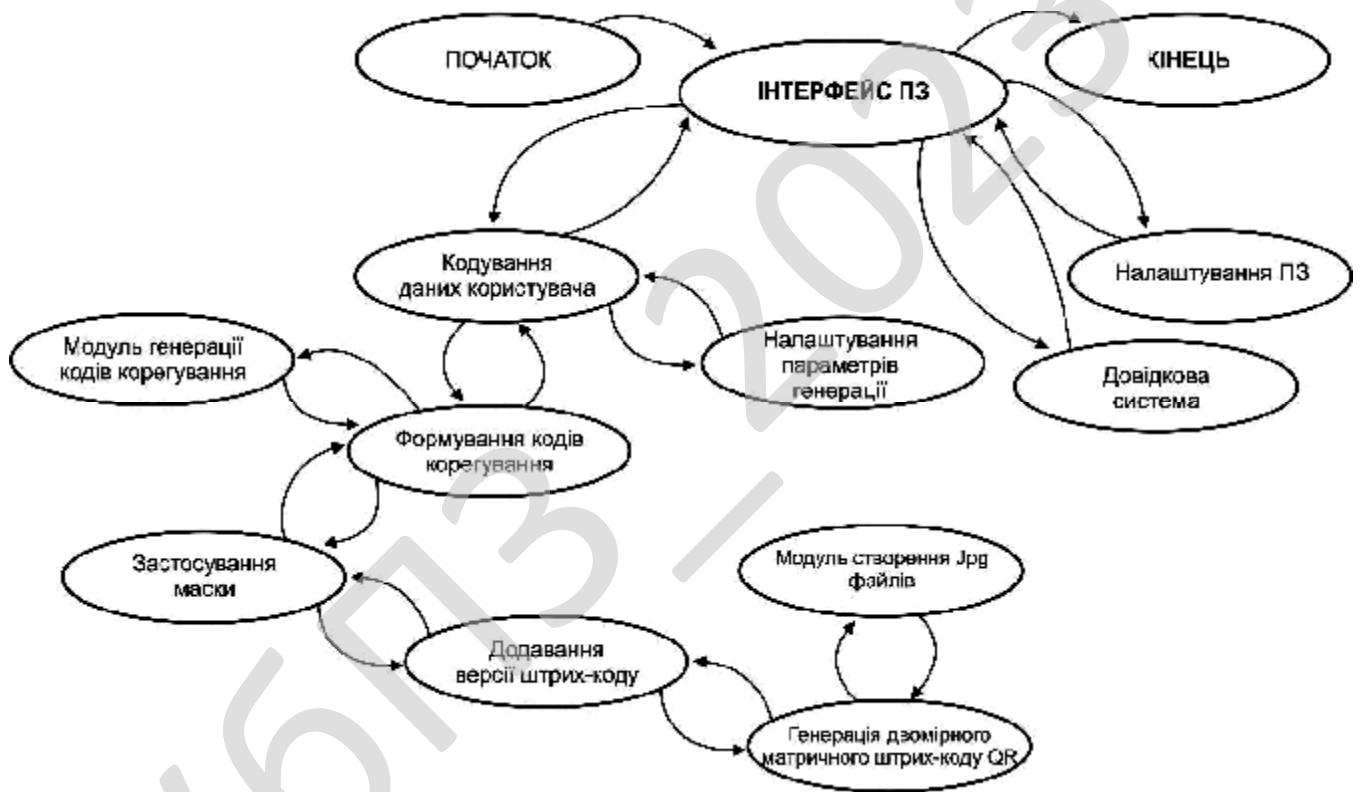


Рисунок 3.3 – Діаграма взаємодії процесів

Далі застосовуючи маску, додавання версії штрих-коду, проводиться генерація двомірного матричного штрих-коду QR з використанням модуля створення JPG файлів.

На відміну від старого штрих-коду, що сканують тонким променем, QR-код визначається сенсором як двовимірне зображення. Три квадрати в кутах

зображення дозволяють нормалізувати розмір зображення і його орієнтацію, а також кут, під яким сенсор ставиться до поверхні зображення. Точки переводяться у двійкові числа з перевіркою за контрольною сумою.

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

КБПЗ-2023

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми. Її робота складається з виконання наступних кроків.

Спочатку проходить ініціалізація програми:

- виділення пам'яті ПЗ;
- ініціалізація черги повідомлень ПЗ;
- ініціалізація призначених для користувача типів даних;
- завантаження налаштувань ПЗ;
- виділення динамічних ресурсів користувача;
- сканування ресурсів ПЗ;

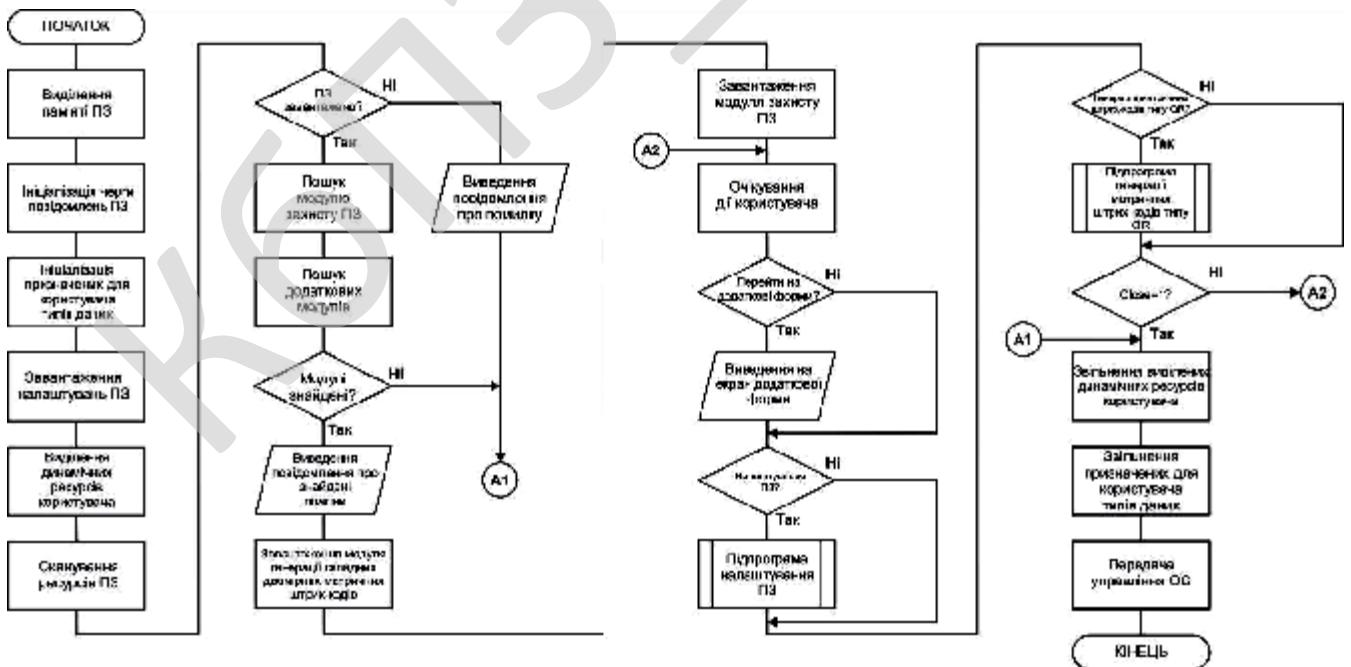


Рисунок 4.1 – Блок-схема основної програми

- перевірка завантаження ПЗ;
- пошук модулю захисту ПЗ;
- пошук додаткових модулів;
- перевірка знаходження модулів;
- виведення повідомлення про знайдені плагіни;
- завантаження модуля генерації складних двомірних матричних штрих-кодів;
- завантаження модуля захисту ПЗ.

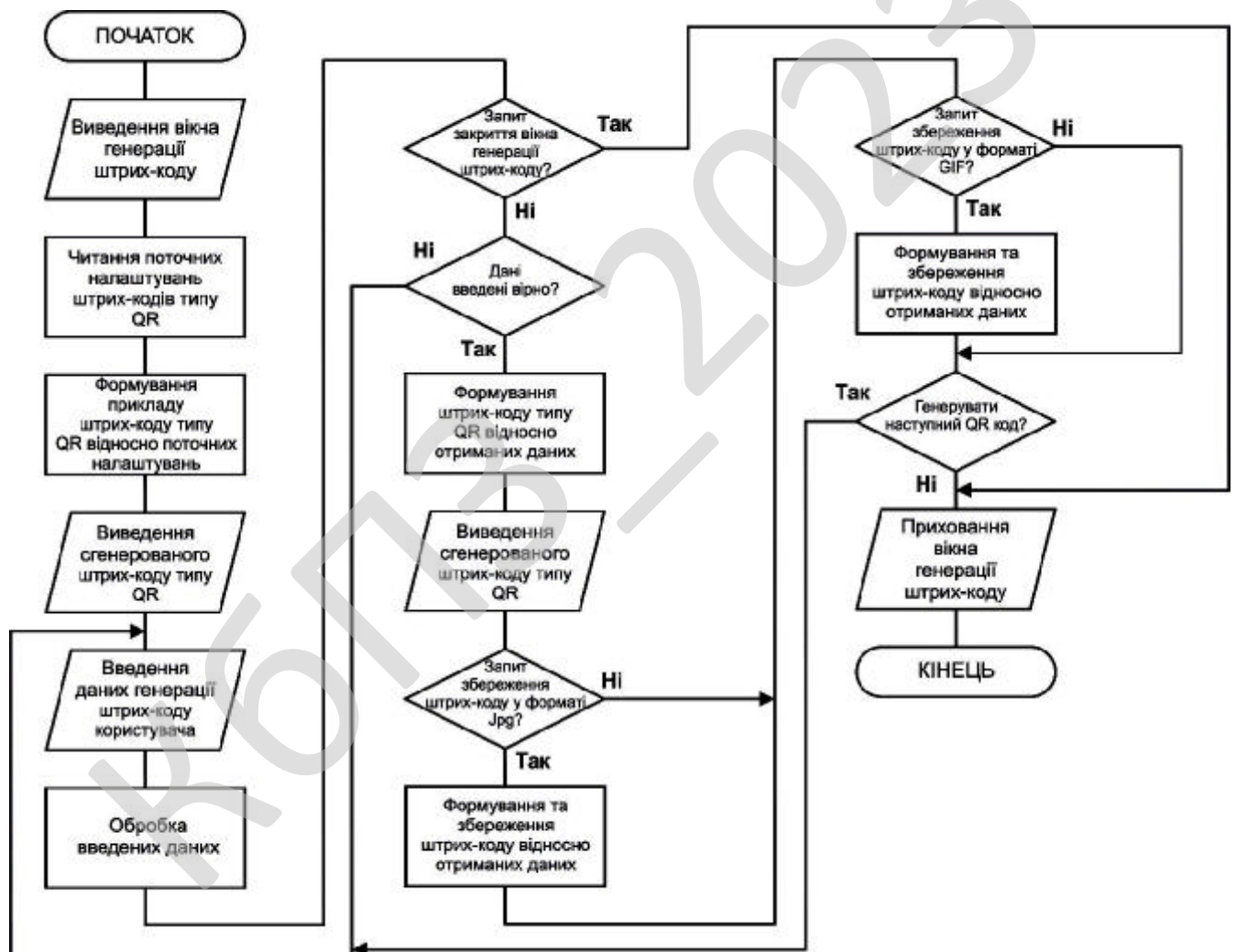


Рисунок 4.2 – Блок-схема роботи підпрограми генерації матричних штрих-кодів типу QR

Далі йдуть блоки основного функціонала ПЗ:

- очікування дії користувача;
- запит переходу на додаткові форми;
- виведення на екран додаткової форми;
- запит на налаштування ПЗ;
- виклик підпрограми налаштування ПЗ;
- запит генерації матричних штрих-кодів типу QR;
- виклик підпрограми генерації матричних штрих-кодів типу QR.

Дії підпрограми генерації матричних штрих-кодів:

- виведення вікна генерації штрих-коду;
- читання поточних налаштувань штрих-кодів типу QR;
- формування прикладу штрих-коду типу QR відносно поточних налаштувань;
- виведення згенерованого штрих-коду типу QR;
- введення даних генерації штрих-коду користувача;
- обробка введених даних;
- запит на закриття вікна генерації штрих-коду;
- запит введених даних;
- формування штрих-коду типу QR відносно отриманих даних;
- виведення згенерованого штрих-коду типу QR;
- запит збереження штрих-коду у форматі Jpg?;
- формування та збереження штрих-коду відносно отриманих даних;
- запит на збереження штрих-коду у форматі GIF;
- формування та збереження штрих-коду відносно отриманих даних;
- запит на генерування наступного QR коду?;
- приховання вікна генерації штрих-коду.

Дії завершення роботи ПЗ:

- якщо поступи сигнал Close=1 завершуємо роботу ПЗ;
- звільнення виділених динамічних ресурсів користувача;

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

- звільнення призначених для користувача типів даних;
- передача управління ОС.

Розроблене ПЗ підтримує три основні типи QR кодів.

1. QRCode (код швидкого відгуку) в основному використовується в Азії.
2. QR-Code 2005 (код швидкого відгуку) варіант QR-Коду, який використовує ISO-5589-1 / Latin-1 як кодування символів за замовчуванням.
3. Micro QR-Code мікро код швидкого відгуку.

Розроблене ПЗ підтримує генерацію QR кодів у наступних форматах.

1. Поточковий формат зображення. Створення штрихкоду у форматі зображення на диску або як бітовий масив у пам'яті. Оптимальне використання у Веб додатках (потік даних http) і у Веб сервісах.

2. Формати растрових зображень. Підтримуються формати GIF, BMP, PNG, JPEG / JPG, TIFF, Підтримка BMP Monochrome (1 біт на піксель).

3. Формати векторних зображень. Підтримуються EPS (Encapsulated PostScript), RGB або CMYK колірний простір, PCL-5/HPGL2 або PCL-3, EMF (Enhanced Windows Metafile), PDF.

Було розроблено прикладний програмний інтерфейс тобто API (Application Programming Interface). Розглянемо основні розроблені функції Barcode Api:

- AboutBox відкриває діалогове вікно з інформацією о відповідному елементі (авторське право та версія);
- CopyToClipboard копіює поточний код в буфер обміну;
- CopyToClipboardEx як і вище описана функція, але із зазначенням контексту пристроїв і можливістю збереження у форматі EMF (при необхідності);
- Licensing відкриває ліцензійну інформація;
- LicenseMe використання ActiveX ® Control в реальному часі;
- BCDraw отрисовка штрихкоду в зазначеному контекстному пристрої;
- Refresh оновлення штрихкоду з поточними параметрами;
- SaveImage збереження штрихкоду в форматі зображення (растрове зображення), з можливістю вибору формату зображення;

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

- SaveImageEx як і вище описана функція, але із зазначенням контексту пристроїв (можливість вибору стиснення);
- ConvertToStream повернення штрихкоду у форматі DataStream зображення (растровий формат) для HTTP (ASP, PHP, ...);
- ConvertToStreamEx як і вище описана функція, але із зазначенням контексту пристроїв (можливість вибору стиснення).

На рисунку 4.3 та 4.4 зображено всі існуючі властивості розробленого класу який відповідає за генерацію QR кодів.

General	Font	Advanced	Behavior
BackStyle	Font	CDMethod	DrawStatus
BarCode	FontName	CountModules	Enabled
MustFit	FontSize	EscapeSequences	OptResolution
Orientation		Format	
PrintDataText	Appearance	GuardWidth	Misc
PrintTextAbove	BackColor	InterpretInputAs	CheckDigits
Text	ForeColor	ModulWidth	CountCheckDigits
TextDistance	TextColor	NotchHeight	CountRows
		Ratio	Height
		RatioDefault	LastError
		RatioHint	LastErrorNo
		SuppressErrorMsg	Width

Рисунок 4.3 – Основні властивості створеного класу генерації QR кодів

Параметризація штрихкодів і виведення даних.

1. Ширина модуля. Регульована ширина модуля (величина X, мінімальна ширина штриха) і відношення ширини штриха.
2. Pixel Shaving. Точне зменшення ширини штриха для поліпшення якості штрихкоду.
3. Підгрупи коду. Повна підтримка різних варіантів штрихкодів (наприклад підгрупи А / В / С, режим двійкового рахунку, режими GS1 та ін.).
4. Контрольні цифри. Можливість вибору розрахунку контрольної цифри. Контрольні цифри певних штрихкодів обчислюються автоматично.

5. Керуюча послідовність. Керуючі послідовності для недрукованих символів ("Test \ 13" == "Test" CR), двійкових даних або для спеціальних штрихкодових функціональних символів (FNC1, ECI ...).

6. Види кодувань. Підтримка Unicode і Single / Multi-Byte набору символів з регульованими кодовими сторінками. Символіка спеціальних кодових сторінок приймається до уваги автоматично. Підтримується UTF-8.

7. Орієнтація. Регульована орієнтація штрихкодів: 0°, 90°, 180°, 270°.

8. Додаткові тексти. Додаткові тексти або коментарі можуть бути додані в штрихкод. Ці текстові елементи можуть бути повернені, вирівняні, переміщені та ін.

9. Індивідуальна отрисовка. Інтерфейс Callback для індивідуальних режимів відтворення (наприклад, для систем лазерного гравіювання або DPM).

QR Code
QR_AppendCount
QR_AppendIndex
QR_AppendParity
QR_ECLLevel
QR_FmtAppIndicator
QR_Format
QR_Mask
QR_Version

Рисунок 4.4 – Додаткові розширені властивості класу генерації QR

Google API вже підготував готову схему для генерації QR-code. Потрібно лише скористатися запропонованими функціями для отримання зображення QR-Code у форматі PNG. За замовчуванням генерується зображення у форматі png, але додавши до запиту параметр `chof = gif` можна згенерувати в форматі GIF.

```
unit uMain; // об'ява
interface // інтерфейс
uses // стандартні бібліотеки
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls;
```



```

    UrlHandle := InternetOpenUrl(hInter, PChar(Url), nil, 0,
INTERNET_FLAG_RELOAD, 0);
    if Assigned(UrlHandle) then
    begin
        repeat
            InternetReadFile(UrlHandle, Buffer, BuffSize, BytesRead);
            if BytesRead>0 then
            Stream.WriteBuffer(Buffer^,BytesRead);
            until BytesRead = 0;
            InternetCloseHandle(UrlHandle);
        end;
        finally
            FreeMem(Buffer);
        end;
        InternetCloseHandle(hInter);
    end
end;
// основна функція яка повертає Stream (Png Image) з QR-кодом.
procedure GetQrCode(Width,Height:Word;Correction_Level:TQrImage_ErrCorrLevel;
    const Data:string;StreamImage : TStream);
Var
    EncodedURL : string;
begin
    EncodedURL:=Format (UrlGoogleQrCode, [Width,Height,QrImgCorrStr[Correction_Lev
el],HTTPEncode(Data)]);
    WinInet_HttpGet(EncodedURL,StreamImage);
end;
procedure TForm1.BtnGetQRClick(Sender: TObject);
var
    ImageStream : TMemoryStream;
    GifImage : TGIFImage;
    ddd : real;
begin
    Image1.Picture:=nil;
    ImageStream:=TMemoryStream.Create;
    GifImage :=TGIFImage.Create;
    try
        try
            ddd:=strtof(float(Edit1.text));
            GetQrCode(300,300,L,floattostr(ddd),ImageStream);
            if ImageStream.Size>0 then
            begin

```

						ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			46

```

        ImageStream.Position:=0;
        GifImage.LoadFromStream(ImageStream);
        Image1.Picture.Assign(GifImage);
    end;
except
    on E: exception do
        ShowMessage(E.Message);
    end;
finally
    ImageStream.Free;
    GIFImage.Free;
end;
end;
end.

```

Можна використовувати TIdHTTP для спрощення завантаження.

```

function TForm1.Qr2Stream(Correction_Level: TQrImage_ErrCorrLevel;
    Width, Height: Integer;
    const DataString: string): TStream;
const
    UrlGoogleQrCode='http://chart.apis.google.com/chart?chs=%dx%d&cht=qr&chld=%s&chl=%s&chof=gif';
    QrImgCorrStr: array[TQrImage_ErrCorrLevel] of char = ('L', 'M', 'Q', 'H');
begin
    Result := TMemoryStream.Create;
    // запит
    IdHTTP1.Get(Format(UrlGoogleQrCode, [Width, Height,
        QrImgCorrStr[Correction_Level],
        HTTPEncode(DataString)]), Result);
    Result.Seek(0, 0);
end;

procedure TForm1.Button1Click(Sender: TObject);
var
    ms: TStream;
begin
    ms := Qr2Stream(H, 300, 300, 'Test'); // створення
    Image1.Picture.Graphic := TGIFImage.Create;
    Image1.Picture.Graphic.LoadFromStream(ms);
    ms.Free;
end;

```

						ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			47

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм LOKI_91. Механізм алгоритму LOKI_91 подібний DES (рисунок 4.4). Блок даних розщеплюється на ліву й праву половини й проходить 16 раундів, що досить нагадує DES. У кожному раунді права половина спочатку піддається операції XOR із частиною ключа, а потім розширювальній перестановці (таблиця 4.1).

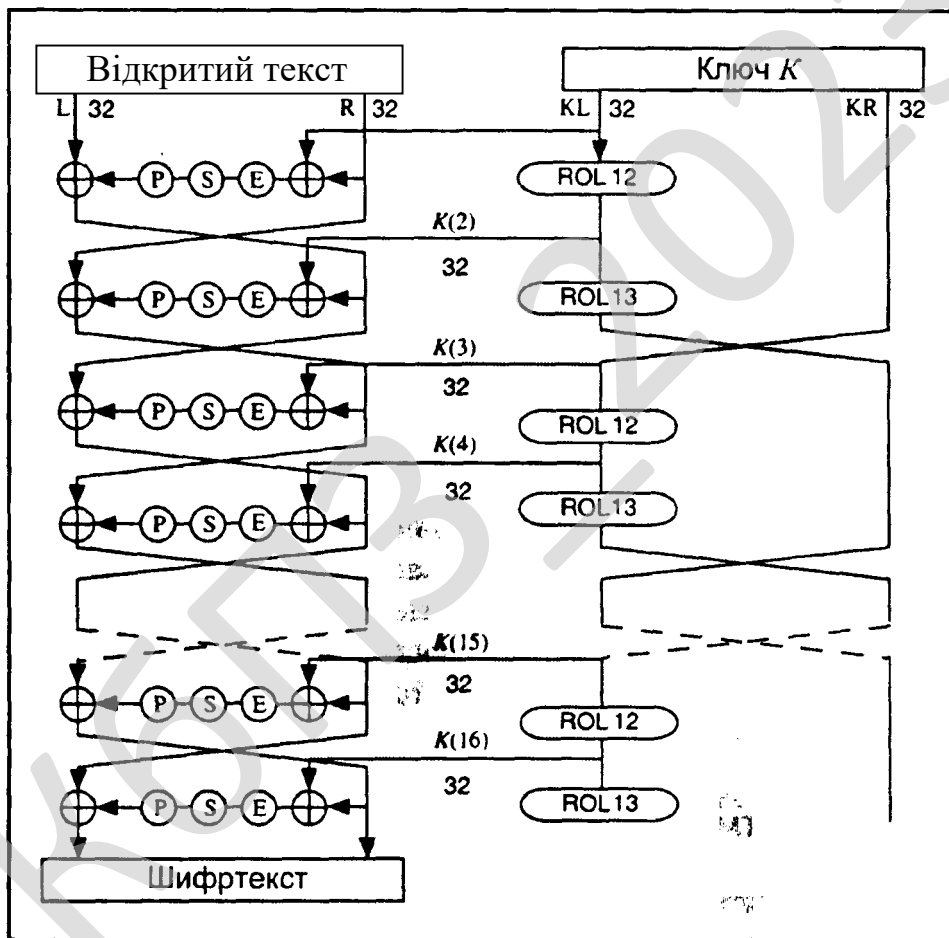


Рисунок 4.4 – Алгоритм LOKI91

Таблиця 4.1 – Перестановка з розширенням

4	3	2	1	32	31	30	29	28	27	26	25
28	27	26	25	24	23	22	21	20	19	18	17
20	19	18	17	16	15	14	13	12	11	10	9
12	11	10	9	8	7	6	5	4	3	2	1

48-бітовий вихід розділяється на чотири 12-бітових блоки. У кожному блоці виконується така підстановка з використанням S-блоку: береться кожний 12-бітовий вхід, 2 старших і 2 молодших біти використовуються для утворення номера r , а вісім внутрішніх біт утворюють номер s . Вихід S-блоку, O , має наступне значення: $O(r,s) = (s + ((r*17) \oplus 0xff) \& 0xff)^{31} \bmod P_r$.

Таблиця 4.2 – Значення P_r

r	1	2	3	4	5	6	7	8
P_r	375	379	391	395	397	415	419	425
r	9	10	11	12	13	14	15	16
P_r	433	445	451	463	471	477	487	499

Після цього чотири 8-бітових результати знову поєднуються, утворюючи 32-бітове число, що піддається операції перестановки, описаній в таблиці 3. Нарешті, для одержання нової лівої половини виконується операція XOR правої половини з колишньою лівою половиною, а ліва половина стає новою правою половиною. Після 16 раундів для одержання остаточного шифртексту знову виконується операція XOR над блоком і ключем.

Таблиця 4.3 – Перестановка за допомогою P-блоку

32	24	16	8	31	23	15	7	30	22	14	6	29	21	13	5
2	20	12	4	27	19	11	3	26	18	10	2	25	17	9	1

Підключи генеруються із ключа досить прямолінійно. 64-бітовий ключ розбивається на ліву й праву половини. На кожному раунді підключем служить ліва половина. Далі вона циклічно зрушується вліво на 12 або 13 біт, потім після кожних двох раундів ліва й права половина міняються місцями. Як і в DES, для зашифрування й розшифрування використовується один й той самий алгоритм із деякими змінами у використанні підключів.

КБГПЗ - 2023

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ, головне вікно якого зображено на рисунку 5.1. Основну частину вікна займає поле згенерованого QR коду. Для налаштування генерації існує меню, яке складається з наступних розділів:

- файл;
- дані;
- налаштування;
- БД кодів QR;
- довідка.



Рисунок 5.1 – Головне вікно генерації QR коду

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Крім розділів меню існують функціональні кнопки:

- генерувати;
- зберегти;
- додати;
- видалити;
- встановити поле;
- очистити QR код;
- налаштування;
- довідка;
- авторське право.

При натисненні кнопки «Генерувати» на екрані з'являється вікно налаштування коду та введення даних, приклад яких зображено на рисунку 5.2.

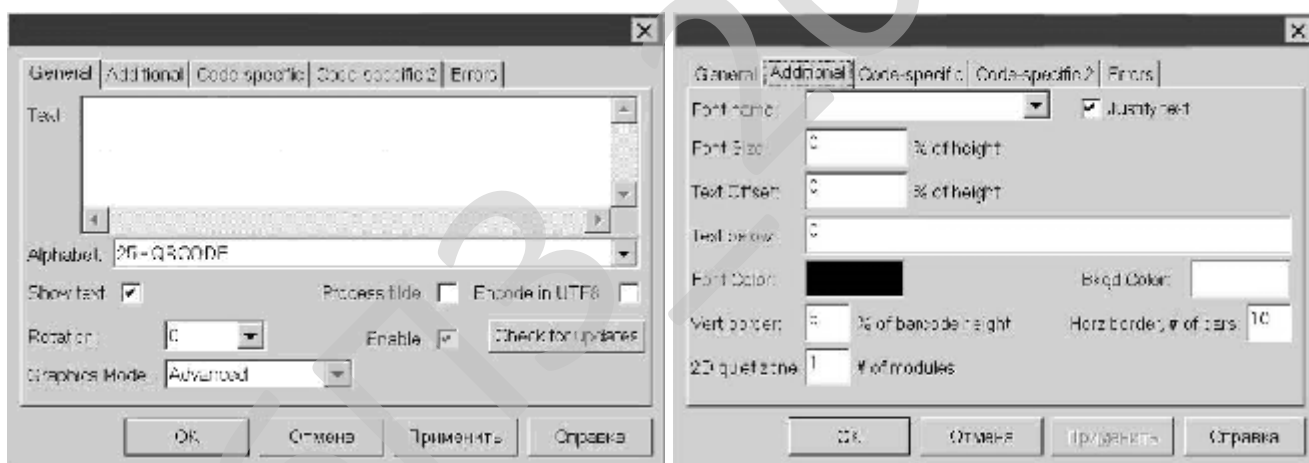


Рисунок 5.2 – Вікна налаштування роботи розробленого ПЗ

На рисунку 5.3 зображена форма авторського права, в якій вказані авторські дані. Для розробленої системи обрано тип ліцензії – безкоштовне програмне забезпечення (Freeware). Тобто програмне забезпечення, що надається безкоштовно, навіть, якщо автор зберігає авторське право на програму.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Центральноукраїнський національний технічний
університет
Кафедра кібербезпеки та програмного забезпечення
ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему " Дослідження та програмна реалізація системи
генерації складних двомірних матричних штрих-кодів "
ОПП «Комп'ютерні науки»
спеціальності 122 «Комп'ютерні науки»
Виконав: Батрак К.М.
Науковий керівник: Улічев О.С.
Кропивницький - 2023

ок

Рисунок 5.3 – Вікно розробника

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		53

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи генерації складних двомірних матричних штрих-кодів.

Метою розробки є дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів.

Об'єктом дослідження є процес генерації складних двомірних матричних штрих-кодів.

Предметом дослідження є методи генерації складних двомірних матричних штрих-кодів.

Методи дослідження базуються на методах обробки зображень, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод генерації складних двомірних матричних штрих-кодів.
- Розроблено вітчизняний продукт генерації складних двомірних матричних штрих-кодів, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 24 днів (один місяць).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи генерації складних двомірних матричних штрих-кодів.

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

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

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	30
3. Запланований термін розробки, днів	Frq	24 (1 місяць)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Г
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	8
8. Кількість форм вихідної інформації.	–	6
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)	–	1
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	3
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	4
17. Складність кінцевого програмного продукту (1-6)	–	5
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	3
20. Вимоги до швидкодії ПП (1-6)	–	3
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	4
23. Професійний рівень аналітиків (1-6)	–	3
24. Професійний рівень програмістів (1-6)	–	4
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	1
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	3
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	30000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	40
38. Ставка податку на додану вартість, %	Н _{дв}	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.:

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

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

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.

$$B = 1,01 + 0,001(2,43 + 3,64 + 4,22 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,2^{1,027} = 5,5 \text{ люд-міс.}$$

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

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

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

$$T_{уточн} = 5,5 \cdot (1 \cdot 1,09 \cdot 1,30 \cdot 0,91 \cdot 1 \cdot 1 \cdot 1 \cdot 1,15 \cdot 1 \cdot 0,87 \cdot 1,10 \cdot 1,22 \cdot 1,12 \cdot 1,10 \cdot 1 \cdot 1 \cdot 1,10) = 12,9 \text{ люд-міс.}$$

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

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

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

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

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

$$T_{РП} = 0,3 \cdot 2,66 \cdot 12,9^{0,33+0,2(1,027-1,01)} \cdot 60 = 112 \text{ люд/день.}$$

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

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	15	Д7
Робочий проект	112	Ф 7.1-7.4
Впровадження	15	Д13
Всього	161	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{161 \cdot 1}{24 \cdot 3} = 7,7 \text{ ставки.}$$

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	0	0	0,0
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	2	60	1
Кабельні господарства ЛОМ на 1 м.п.	2,5	200	500	8,33
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	51,99

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:

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

$$\Phi_{\text{др}}^c = \frac{52 \cdot 1}{1,2} = 43 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 43/(24 \cdot 8) = 0,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

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

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2016, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	1	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	1	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	1	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	1	
	Контроль взаєморозрахунків з постачальниками	1	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,5
	Створення графічних і стилістичних елементів сайту	1	
	Оформлення банерів і промо-сторінок	1	
	Розміщення графіки і контенту на Інтернет сторінках	1	
Всього		4	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,5
	Верстка друкованих видань	1	
	Додрукова підготовка макетів	1	
	Розміщення графіки і контенту на Інтернет сторінках	1	
Всього		4	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	0,5	11112	5556
Продакт-менеджер	0,5	8500	4250
Інженер-програміст	7,7	8500	65450
Інженер-електронщик	0,2	8000	1600
Інженер-системотехнік	0,5	8000	4000
Адміністратор мережі	0,5	8000	4000
Дизайнер WEB	0,5	10000	5000
Всього за період розробки	$R_{cn} = 10,4$	-	$\Phi_{роб} = 89856$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{cd} = \frac{89856}{10,4 \cdot 24} = 360 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

$$B_{y0} = R_{cn}^1 S_y \Pi_{nl}, \quad (7.9)$$

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

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

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Системна плата	Asus H81M2, 6x USB 2.0, 2x USB 3.0, 5 Audio, 2x PS/2, 2x DVI, 1x HDMI, 1x LAN (RJ-45)	-
Відеокарта	nVidia GeForce GTX 550 Ti, 1 GB GDDR5 192-bit	-
Жорсткий диск	SSD: 480 Gb	-
Оперативна пам'ять	Kingston DDR3 8GB Intel/AMD – 2 шт	-
DVD-привод	-	-
Корпус	Azza 727-06U Tower GZ-X4 Silver 500W (GZ-X4 Silver)	-
Кулер	-	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-Ext. 3.5", 1*USB2.0+AUDIO+1394, multi: A Type Cards, black	-
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1 170/160, D-SUB, Wide)	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37 Photo	2970
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	UPS APC BACK-UPS ES 525VA 230V RUSSIA (BE525-RS)	1348

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	13	11457	14894,1	163835,1
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2970	297	3267
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	185653,6

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	2080000	-	-
2. Передавальні пристрої	208000	-	-
Всього по групі	2288000	5	114400

Продовження таблиці 7.8

1	2	3	4
Група 4			
3. Обчислювальна техніка	185654	-	-
Всього по групі	185654	50	92827
Група 5, 6			
4. Вимірювальні пристрої	3999	25	-
5. Транспортні засоби	0	20	-
6. Господарський інвентар	45500	25	-
Всього по групі	49499	-	12374,75
7. Нематеріальні активи	30000	10	3000
Разом	$K_p = 2553153$		$A_p = 222602$

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.

$$Z_o = 360 \cdot 161 / 30 = 1932 \text{ грн.}$$

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, %.

$$Z_d = 1932 \cdot 10 \cdot 0,01 = 193 \text{ грн.}$$

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_3, \quad (7.18)$$

де: C_3 – 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.

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

$$Z_M = (40 + 93 + 1702) / 30 = 61 \text{ грн.}$$

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, %.

$$O_n = 1932 \cdot 15 \cdot 0,01 = 290 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 30$ прим.):

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

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

$$A_m = 222602 \cdot 1 / (30 \cdot 12) = 618 \text{ грн.}$$

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

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

$$C_n = 1932 + 193 + 468 + 290 + 61 + 290 + 618 = 3852 \text{ грн.}$$

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

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	1932
2. Додаткова зарплата виконавців	Z_d	193
3. Відрахування на соціальні потреби	C_{oc}	468
4. Загальногосподарські витрати	G_{ocn}	290
5. Витрати на матеріали	Z_M	61
6. Освоєння нових операційних систем, мов програмування	O_n	290
7. Амортизація основних фондів	A_M	618
8. Повна собівартість програмного забезпечення	C_n	3852
9. Плановий прибуток	P_p	1541
10. Ціна підприємства $C_n = C_n + P_p$	C_n	5393
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{об} \cdot C_n$	$ПДВ$	1077
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	6470

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

Для даного програмного забезпечення рівень рентабельності складає 40%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 40 \cdot 3852 = 1541 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

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

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	6470
Всього капітальних витрат	–	6470

7.7 Визначення експлуатаційних витрат

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

Витрати на профілактичні роботи:

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

де: T_p – кількість годин обслуговування кожного комп'ютера за рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування	Z_p	32208	20130
2. Витрати на електроенергію	$Z_{ел}$	277	116
3. Витрати на амортизацію	$Z_{ам}$	0	3235
Всього витрат за рік	I	32485	23481

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 150 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 100 \cdot 1,1 \cdot 1,22 = 32208 \text{ грн},$$

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 ($C_{ел}$):

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

$$Z_{ел \text{ баз}} = 0,55 \cdot 240 \cdot 2,1 = 277 \text{ грн}.$$

$$Z_{ел \text{ нов}} = 0,55 \cdot 100 \cdot 2,1 = 116 \text{ грн}.$$

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

$$E_{cn} = (32485 - 23481) - 0,5 \cdot 6470 = 5769 \text{ грн.}$$

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

$$T_{cn} = \frac{K_n - K_0}{I_0 - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{6470}{32485 - 23481} = 0,72 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

7.9 Висновки

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

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

- Забезпечувати безпечні і нешкідливі умови праці для працівників, використовуючи сучасні засоби техніки безпеки, санітарно-гігієнічні умови, засоби клектинго та індивідуального захисту, оптимальні режими праці та відпочинку.

- Проводити атестацію робочих місць на відповідність нормативно-правовим актам з охорони праці та аудит з охорони праці.

- Проводити навчання та інструктаж з питань охорони праці, з надання першої медичної допомоги потерпілим від нещасних випадків і правил поведінки у разі виникнення аварії⁵.

- Забезпечувати лікувально-профілактичне обслуговування працюючих, санітарно-побутове обслуговування, пільги і компенсації для працівників, які працюють у важких і шкідливих умовах.

- Нести відповідальність за порушення законодавства про охорону праці та зподіяння шкоди життю і здоров'ю працівників.

8.2 Аналіз умов праці на робочому місці ІТ-фахівця

На робочому місці ІТ-фахівця (або програміста) виникають небезпечні та шкідливі для безпечної життєдіяльності фактори:

- підвищений рівень шуму;
- несприятливі мікрокліматичні умови;
- недостатній рівень освітленості;
- шкідливі речовини;
- підвищений рівень електромагнітних випромінювань радіочастот;
- висока напруга електричної мережі;
- статична електрика та інші.

Робота програміста супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці програміста є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, щодо простору, форми, розміру забезпечує йому зручне положення при роботі і високу продуктивність праці при найменшому фізичному і психічному напруженні.

Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця ІТ-фахівців, встановлені оптимальні параметри мікроклімату, а за неможливості їх дотримання використовують допустимі параметри. Робота ІТ-фахівця за важкістю відноситься до Іа (роботи, що виконуються сидячи і не потребують фізичного напруження) та Іб (роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням) категорій. В таблиці 8.1. наведені оптимальні параметри мікроклімату в приміщеннях.

Таблиця 8.1 – Параметри мікроклімату для приміщень з ПК

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	22...24°C; 40... 60%; до 0,1 м/с
Теплий	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	23...25 °С 40...60% 0,1...0,2 м/с

Виміряні за допомогою приладів температура та вологість у приміщеннях праці ІТ-фахівців повинні відповідати зазначеним у таблиці для теплового періоду

освітленості; надмірна освітленість; неправильний напрям світла. Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань. [21]

8.3 Пропозиції щодо підвищення працездатності ІТ-фахівця

Практичне значення заходів щодо підвищення працездатності впливає із закономірностей її динаміки і зводиться ось до чого:

- збільшення фази стійкого стану у фонді робочого часу;
- прискорення процесу впрацювання;
- віддалення фази розвитку втоми;
- забезпечення високої продуктивності праці за нормальних фізіологічних затрат.

Комплекс заходів щодо підвищення і збереження працездатності працівників на оптимальному рівні реалізується на техніко-організаційному, соціально-економічному, санітарно-гігієнічному, медико-біологічному, психологічному напрямках.

Вагомим фактором високої працездатності і продуктивності праці є оптимізація трудових навантажень на основі механізації і автоматизації виробничих процесів, удосконалення технології, скорочення і ліквідації важкої ручної праці. Доведено, що при правильній організації праці на легких роботах спостерігається найбільша тривалість фази стійкого стану, а на важких роботах вона нетривала.

Високий рівень працездатності безпосередньо залежить від умов праці, оскільки поліпшення їх супроводжується зменшенням енергетичних затрат організму на подолання несприятливого впливу факторів виробничого

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

середовища.

Важливим напрямком підвищення працездатності працюючих є ритмізація трудових процесів, оптимізація темпу роботи, а також раціоналізація трудових рухів на фізіологічній основі, що сприяє формуванню і закріпленню робочих динамічних стереотипів, а отже зменшенню м'язових і вольових зусиль. Ритмічна робота підвищує функціональні можливості організму, сприяє його тренуваності і забезпечує економізацію енергетичних затрат. [1]

Багатьом програмістам постійно доводиться працювати з великою кількістю програм одночасно. Часте перемикання туди-сюди між IDE та довідкою суттєво зменшує продуктивність фахівця. Однак вирішення цієї проблеми досить просте та очевидне: встановлення більшої кількості моніторів.

Оптимальним варіантом є два монітори. Все ж таки це найпростіший з апаратної точки зору варіант. Крім того, якби їх було більше, то ними було б важче керувати, та й столі просто не вистачить місця на ще один монітор. Але тут ще залежить розміру моніторів. Є системи із 4 або 6 відносно невеликими екранами, які кріпляться на кронштейні. Але оптимальним є два 27-дюймові монітори, на яких все добре видно, особливо коли працювати доводиться в основному з текстом [3].

8.4 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів;
- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;
- засобів пожежогасіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами.

Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів,

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань.

Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [24]

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення, ширина якого складає 6 м, довжина – 7 м, висота – 3,4 м.

У зазначеному приміщенні працює 7 людей.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою [1]:

$$F=ESKZ/n,$$

де: F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; $E = 300$ Лк;

S – площа освітлюваного приміщення (у нашому випадку $S=6 \times 7 = 42$ м²);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку $K = 1,5$);

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку $Z = 1,1$);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці [8]; залежить від характеристик світильника,

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$.

Обчислимо індекс приміщення за формулою:

$$i = S / (h(A+B)),$$

де: S – площа приміщення, $S = 42 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 3,4 \text{ м}$ (співпадає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 6 \text{ м}$;

B – довжина приміщення, $B = 7 \text{ м}$.

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$i = 1,4.$$

Знаючи індекс приміщення, за знаходимо $n = 0,23$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом ламп) [8]. Підставимо всі значення у формулу, визначимо світловий потік: $F = 90391 \text{ Лм}$.

Для розрахунку дудемо використовувати світлодіодні стельові панелі Призма-72 6400К, світловий потік яких $F_{л} = 7200 \text{ Лм}$.

Число ламп визначається по формулі:

$$N = F / F_{л}$$

де: F – світловий потік,

$F_{л}$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$N = 90391 / 7200 = 12,5 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 13 шт.

Висновки до розділу

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

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

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

КБПЗ - 2023

					VKPM-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи генерації складних двомірних матричних штрих-кодів.

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 генерації складних двомірних матричних штрих-кодів.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем генерації складних двомірних матричних штрих-кодів.
- Досліджена система генерації складних двомірних матричних штрих-кодів.
- На основі отриманих результатів досліджень створена програмна реалізація системи генерації складних двомірних матричних штрих-кодів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання генерації складних двомірних матричних штрих-кодів.

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.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

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

Програма реалізована на мові високого рівня Delphi 10.4 Sydney. 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 LOKI_91.

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 5769 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,72 роки.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Батрак К.М. Дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Adam Freeman. Pro Go The Complete Guide to Programming Reliable and Efficient Software Using Golang. Apress Media. 2022. 1078 p.
3. Fernando Doglio. Skills of a Successful Software Engineer. Manning. 2022. 182 с.
4. M. Holmes He. Creating Apps with React Native. Apress Media. 2022. 445 p.
5. Maurício Aniche. Effective Software Testing. Manning Publications. 2021. 372 p
6. Priscila Heller. Automating Workflows with GitHub Actions. Packt Publishing. 2021. 216 p.
7. JJ Geewax. API Design Patterns. Manning Publications Co. 2021. 481 p.
8. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
9. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
10. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
11. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
12. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

13. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
14. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.
15. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
16. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.
17. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.
18. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.
19. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.
20. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143
21. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.
22. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-

feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

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

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

25. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43.

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

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

28. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

29. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of

Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

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

31. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.

32. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.

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

34. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

35. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». Системи управління, навігації та зв'язку, 2023, вип. 2(72), С. 170-178.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

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

37. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.

38. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

39. Смірнов О.А., Коваленко О.В. Використання псевдобулевих методів бівалентного програмування для управління ризиками розробки програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 1 (37). – Полтава: ПолтНТУ. – 2016. – С. 98-103.

40. Смірнов О.А., Лисенко І.А. Формалізація процесу проектування тестових наборів. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 3 (48). – Харків: ХУПС. – 2016. – С.96-100.

41. Смірнов О.А., Лисенко І.А. Удосконалення методу перевірки коректності таблиць рішень для подання тестових наборів. Збірник наукових праць "Системи обробки інформації". – Випуск 8 (145). – Х.: ХУПС – 2016. – С. 77-80.

42. Смірнов О.А., Лисенко І.А. Розробка впорядкованих каскадних таблиць рішень із використанням матриць слідування. Збірник наукових праць "Системи обробки інформації". – Випуск 6 (143). – Х.: ХУПС – 2016. – С. 216-220.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

43. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Метод кількісної оцінки ризиків розроблення програмного забезпечення. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 128-133.

44. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Метод якісного аналізу ризиків розроблення програмного забезпечення. Наука і техніка Збройних Сил України. – Випуск 2(23). – Харків: ХУПС. – 2016. – С. 150-158.

45. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Проблеми аналізу та оцінки ризиків інформаційної діяльності. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 40-42.

46. Смірнов О.А., Коваленко А.С., Коваленко О.В., Доренський О.П. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи. Системи озброєння і військова техніка. – Випуск 2(46) – Х.: ХУПС – 2016. – С. 103-107.

47. Smirnov A.A., Kovalenko A.V., Kovalenko A.S., Dorensky A.P. Information model and its element for displaying information on technical condition of objects of integrated information system. International Journal of Computational Engineering Research (IJCER). – Volume 6, Issue 1. – India. Delhi. – 2016. – P. 21-27.

48. Смірнов О.А., Євсєєв С.П., Король О.Г., Коваленко О.В., Коваленко А.С., Смірнов С.А. Архітектура мікропроцесорів та компонентів ЕОМ. Навчальний посібник – Кіровоград: Вид. Лисенко В.Ф., 2015. – 550 с.

49. Смірнов О.А., Коваленко О.В., Мелешко Є.В., Константинова Л.В., Кожанова А.С. Інженерія програмного забезпечення. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.А. Смірнова Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки України від 18.03.2013 року № 1/11-5584. – Кіровоград: КНТУ 2013. – 409с.

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

50. Смірнов О.А., Осадчий С.І., Мелешко Є.В., Іванов С.Г., Павленко М.А., Усачов О.М. Основи технічної експлуатації АСУ. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.А. Смірнова, Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки України від 27.02.2013 року № 1/11-4487 . – Кіровоград: КНТУ 2013. – 322с.

51. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лешко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. Коваленка О.В., Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки України від 26.02.2013 року № 1/11-4368. – Кіровоград: КНТУ 2013. – 257с.

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

					ВКРМ-122.23.0031.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

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-122.23.0031.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Батрак К.М.				<i>Дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів</i>	Літ.	Аркуш	Аркушів
Перевірів	Улічев О.С.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22М-2			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи генерації складних двомірних матричних штрих-кодів.

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. № 33-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи генерації складних двомірних матричних штрих-кодів.

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-122.23.0031.00.00.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 генерації складних двовірних матричних штрих-кодів;
- 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-122.23.0031.00.00.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 Delphi 10.4 Sydney.

					ВКРМ-122.23.0031.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті пропозиції щодо підвищення працездатності ІТ-фахівця.

					ВКРМ-122.23.0031.00.00.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 – 95 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

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

					ВКРМ-122.23.0031.00.00.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

_____ Улічев О.С.

*Дослідження та програмна реалізація
системи генерації складних двомірних матричних штрих-кодів*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 40

Літера: РП

Кропивницький – 2023 року

ОСНОВНИЙ ПРОЕКТ ПЗ

```
program QRcode_DIPLOM;
{
Батрак Костянтин Миколайович
Дослідження та програмна реалізація системи генерації складних двомірних
матричних штрих-кодів
Центральноукраїнський національний технічний університет
2023, каф. КБПЗ
}

Uses // бібліотеки
  Forms,
  main in 'main.pas' {Form1},
  QRcode1 in 'QRcode1.pas', // перша бібліотека, налаштування QR коду
  QRcode2 in 'QRcode2.pas'; // друга бібліотека, генерування QR коду

{$R *.RES} // ресурси

begin
  Application.Initialize; // Ініціалізація
  Application.CreateForm(TForm1, Form1); // створення форми
  Application.Run; // запуск
end.
```

ФАЙЛ ГОЛОВНОГО ВІКНА ПЗ – MAIN.PAS

```

unit main;
{
Батрак Костянтин Миколайович
Дослідження та програмна реалізація системи генерації складних двомірних
матричних штрих-кодів
Центральноукраїнський національний технічний університет
2023, каф. КБПЗ
}
Interface // інтерфейс

Uses // бібліотеки
  SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  QRcode, ExtCtrls, StdCtrls;

type
  TForm1 = class(TForm)
    Image1: TImage;
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Label3: TLabel;
    ComboBox1: TComboBox;
    Edit2: TEdit;
    Label4: TLabel;
    Button1: TButton;
    ComboBox2: TComboBox;
    CBCheckSum: TCheckBox;
    Button2: TButton;
    EditWidth: TEdit;
    Label5: TLabel;
    FontDialog1: TFontDialog;
    BtnFont: TButton;
    ComboBox3: TComboBox;
    procedure FormCreate(Sender: TObject);
    procedure Label2Click(Sender: TObject);
    procedure ComboBox1Click(Sender: TObject);
    procedure Edit2Change(Sender: TObject);
    procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Button1Click(Sender: TObject);
    procedure ComboBox2Click(Sender: TObject);
    procedure CBCheckSumClick(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure EditWidthEnter(Sender: TObject);
    procedure EditWidthExit(Sender: TObject);
    procedure QRcode1Change(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure BtnFontClick(Sender: TObject);
    procedure ComboBox3Change(Sender: TObject);
  private
    { Декларация-Private }
    QRcode1 : TAsQRcode;
    procedure print_demo(bc:TAsQRcode);
  public
    { Декларация-Public }
  end;

var
  Form1: TForm1;

Implementation

```

```

Uses // бібліотеки
WinTypes, WinProcs, Printers, QRcode2;

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  {
  Створити QRcode об'єкта.
  це об'єкт звільняється, коли форма буде зруйнована
  }
  QRcode1 := TQRcode.Create(self);
  QRcode1.Top := 50;
  QRcode1.Left := 30;
  QRcode1.Type := bcCodePostNet;
  QRcode1.Modul := 2;
  QRcode1.Ratio := 2.0;
  QRcode1.Height := 50;
  QRcode1.OnChange := Self.QRcode1Change;
  ComboBox2.ItemIndex := integer(QRcode1.ShowText);
end;

procedure TForm1.ComboBox1Click(Sender: TObject);
begin
  QRcode1.Type := TQRcodeType(ComboBox1.ItemIndex);
end;

procedure TForm1.Edit2Change(Sender: TObject);
begin
  try
    QRcode1.Angle := StrToFloat(Edit2.Text);
  except
    QRcode1.Angle := 0.0;
  end;
end;

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  QRcode1.Top := Y;
  QRcode1.Left := X;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  Print;
end;

procedure TForm1.ComboBox2Click(Sender: TObject);
begin
  QRcode1.ShowText := TQRcodeOption(ComboBox2.ItemIndex);
end;

procedure TForm1.CBChecksumClick(Sender: TObject);
begin
  QRcode1.Checksum := CBChecksum.Checked;
end;

procedure TForm1.print_demo(bc:TAsQRcode);
var
  tmpQRcode : TAsQRcode;
begin

```

```

{
створити об'єкт QRcode, тому що ми хочемо змінити деякі ластивості
}
tmpQRcode := TQRcode.Create(nil);
{ copy the object }
tmpQRcode.Assign(bc);
try
  with printer do
  begin
    BeginDoc;
    Canvas.TextOut(10, 10, 'QRcode'+DateTimeToStr(Now));
{ висота QRcode: 40mm }
    tmpQRcode.Height := ConvertMmToPixelsY(40.0);
    tmpQRcode.Height := ConvertInchToPixelsY(1.5);
{ ширина: 0.5mm }
    tmpQRcode.Modul := ConvertMmToPixelsX(0.5);
    tmpQRcode.Top := ConvertMmToPixelsY(100.0);
    tmpQRcode.Left := ConvertMmToPixelsX(35.0);
    tmpQRcode.DrawQRcode(Canvas);
    EndDoc;
  end;
finally
  tmpQRcode.Free;
end;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  print_demo(QRcode1);
end;

procedure TForm1.EditWidthEnter(Sender: TObject);
begin
  EditWidth.Text := IntToStr(QRcode1.Width);
end;

procedure TForm1.EditWidthExit(Sender: TObject);
begin
  QRcode1.Width := StrToInt(EditWidth.Text);
  EditWidth.Text := IntToStr(QRcode1.Width);
end;

procedure TForm1.QRcode1Change(Sender: TObject);
begin
  Image1.Picture := nil;
  QRcode1.DrawQRcode(Image1.Canvas);
end;

procedure TForm1.Edit1Change(Sender: TObject);
begin
  QRcode1.Text := Edit1.Text;
end;

procedure TForm1.BtnFontClick(Sender: TObject);
begin
  with FontDialog1 do
  begin
    Font := QRcode1.ShowTextFont;
    if Execute then
      QRcode1.ShowTextFont := Font;
  end;
end;
end;

```

```
procedure TForm1.ComboBox3Change(Sender: TObject);  
begin  
    QRcode1.ShowTextPosition := TShowTextPosition(ComboBox3.ItemIndex);  
end;  
end.
```

К6П3 - 2023

ФАЙЛ БІБЛІОТЕКИ - QRCODE1.PAS

```

unit QRcode1;
{
Батрак Костянтин Миколайович
Дослідження та програмна реалізація системи генерації складних двомірних
матричних штрих-кодів
Центральноукраїнський національний технічний університет
2023, каф. КБПЗ
}
interface

Uses // бібліотеки
  Classes, SysUtils, Graphics, QRcode, QRCTRLS;

type
  TQRasQRcode = class(TQRImage)
  private
    FQRcode :TASQRcode;
    FIgnoreOnChange :Boolean;
    procedure FixupSize;
    procedure QRcodeToCanvas;
    procedure SetQRcode(const Value: TASQRcode);
    function GetQRcode: TASQRcode;
    function GetAngle: double;
    function GetChecksum: boolean;
    function GetChecksumMethod: TChecksumMethod;
    function GetColor: TColor;
    function GetColorBar: TColor;
    function GetModul: integer;
    function GetRatio: double;
    function GetShowText: TQRcodeOption;
    function GetText: string;
    function GetType: TQRcodeType;
    procedure SetAngle(const Value: double);
    procedure SetChecksum(const Value: boolean);
    procedure SetChecksumMethod(const Value: TChecksumMethod);
    procedure SetColor(const Value: TColor);
    procedure SetColorBar(const Value: TColor);
    procedure SetModul(const Value: integer);
    procedure SetRatio(const Value: double);
    procedure SetShowText(const Value: TQRcodeOption);
    procedure SetText(const Value: string);
    procedure SetType(const Value: TQRcodeType);
    function GetQRcodeHeight: Integer;
    function GetQRcodeWidth: Integer;
    procedure SetQRcodeHeight(const Value: Integer);
    procedure SetQRcodeWidth(const Value: Integer);
  protected
    procedure Loaded; override;
    procedure Resize; override;
    procedure OnQRcodeChange(Sender :TObject);
  public
    constructor Create(AOwner :TComponent); override;
    property QRcode :TASQRcode read GetQRcode write SetQRcode;
  published
    property Angle :double read GetAngle write SetAngle;
    property QRcodeHeight :Integer read GetQRcodeHeight write SetQRcodeHeight;
    property QRcodeWidth :Integer read GetQRcodeWidth write SetQRcodeWidth;
    property Checksum :boolean read GetChecksum write SetChecksum
  default FALSE;
    property ChecksumMethod :TChecksumMethod read GetChecksumMethod
  write SetChecksumMethod default csmModulo10;
    property Color :TColor read GetColor write SetColor default clWhite;

```

```

        property ColorBar :TColor read GetColorBar write SetColorBar default
c1Black;
        property Modul :integer read GetModul write SetModul;
        property Ratio :double read GetRatio write SetRatio;
        property ShowText :TQRcodeOption read GetShowText write SetShowText
default bcoNone;
        property Text :string read GetText write SetText;
        property Typ :TQRcodeType read GetTyp write SetTyp default
bcCode_2_5_interleaved;
        property Picture stored False;
    end;

implementation

Uses // бібліотеки
    Math;

end;

{ реалізація коду TQRASQRcode }
procedure TQRASQRcode.QRcodeToCanvas;
begin
    //очищення канви
    Picture.Assign(nil);
    //рисування QRcode
    FixupSize;
    QRcode.DrawQRcode(Self.Canvas);
end;
constructor TQRASQRcode.Create(AOwner :TComponent);
begin
    inherited;
    FQRcode := TQRcode.Create(nil);
    FQRcode := TQRcode.Create(Self);
    FQRcode.OnChange := OnQRcodeChange;
    FQRcode.Height := 50; // по замовчанню
    FQRcode := TQRcode.Create(Self);
    FQRcode.OnChange := OnQRcodeChange;
    FQRcode.Height := 50; // по замовчанню
end;
{
destructor TQRASQRcode.Destroy;
begin
    inherited;
end;
}
procedure TQRASQRcode.FixupSize;
begin
    if (not (csLoading in ComponentState)) then
        begin
            FIgnoreOnChange := True;
            try
                // Налаштування ширини QRcode
                Height := QRcode.CanvasHeight;
                Width := QRcode.CanvasWidth;
            finally
                FIgnoreOnChange := False;
            end;
        end;
end;

function TQRASQRcode.GetAngle: double;
begin
    Result := FQRcode.Angle;
end;

```

```
function TQRAsQRcode.GetQRcode: TASQRcode;
begin
  Result := FQRcode;
end;

function TQRAsQRcode.GetQRcodeHeight: Integer;
begin
  Result := QRcode.Height;
end;

function TQRAsQRcode.GetQRcodeWidth: Integer;
begin
  Result := QRcode.Width;
end;

function TQRAsQRcode.GetChecksum: boolean;
begin
  Result := QRcode.CheckSum;
end;

function TQRAsQRcode.GetChecksumMethod: TChecksumMethod;
begin
  Result := QRcode.CheckSumMethod;
end;

function TQRAsQRcode.GetColor: TColor;
begin
  Result := QRcode.Color;
end;

function TQRAsQRcode.GetColorBar: TColor;
begin
  Result := QRcode.ColorBar;
end;

function TQRAsQRcode.GetModul: integer;
begin
  Result := QRcode.Modul;
end;

function TQRAsQRcode.GetRatio: double;
begin
  Result := QRcode.Ratio;
end;

function TQRAsQRcode.GetShowText: TQRcodeOption;
begin
  Result := QRcode.ShowText;
end;

function TQRAsQRcode.GetText: string;
begin
  Result := QRcode.Text;
end;

function TQRAsQRcode.GetTyp: TQRcodeType;
begin
  Result := QRcode.Typ;
end;

procedure TQRAsQRcode.Loaded;
begin
```

```
    inherited;
    QRcodeToCanvas;
end;

procedure TQRAsQRcode.OnQRcodeChange(Sender: TObject);
begin
    if FIgnoreOnChange then EXIT;
    if (not (csLoading in ComponentState)) then
        QRcodeToCanvas;
end;

procedure TQRAsQRcode.Resize;
begin
    inherited;
    FixupSize;
end;

procedure TQRAsQRcode.SetAngle(const Value: double);
begin
    QRcode.Angle := Value;
end;

procedure TQRAsQRcode.SetQRcode(const Value: TASQRcode);
begin
    FQRcode.Assign(Value);
end;

procedure TQRAsQRcode.SetQRcodeHeight(const Value: Integer);
begin
    QRcode.Height := Value;
end;

procedure TQRAsQRcode.SetQRcodeWidth(const Value: Integer);
begin
    QRcode.Width := Value;
end;

procedure TQRAsQRcode.SetChecksum(const Value: boolean);
begin
    QRcode.CheckSum := Value;
end;

procedure TQRAsQRcode.SetChecksumMethod(const Value: TChecksumMethod);
begin
    QRcode.CheckSumMethod := Value;
end;

procedure TQRAsQRcode.SetColor(const Value: TColor);
begin
    QRcode.Color := Value;
end;

procedure TQRAsQRcode.SetColorBar(const Value: TColor);
begin
    QRcode.ColorBar := Value;
end;

procedure TQRAsQRcode.SetModul(const Value: integer);
begin
    QRcode.Modul := Value;
end;

procedure TQRAsQRcode.SetRatio(const Value: double);
```

```
begin
  QRcode.Ratio := Value;
end;

procedure TQRAsQRcode.SetShowText(const Value: TQRcodeOption);
begin
  QRcode.ShowText := Value;
end;

procedure TQRAsQRcode.SetText(const Value: string);
begin
  QRcode.Text := Value;
end;

procedure TQRAsQRcode.SetTyp(const Value: TQRcodeType);
begin
  QRcode.Typ := Value;
end;
end.
```

K6П3-2023

ФАЙЛ БІБЛІОТЕКИ - QRCODE2.PAS

```

unit QRCODE2;
{
Батрак Костянтин Миколайович
Дослідження та програмна реалізація системи генерації складних двомірних
матричних штрих-кодів
Центральноукраїнський національний технічний університет
2023, каф. КБПЗ
}
interface

Uses // бібліотеки
  Classes, Graphics;

type
  TQRcodeType =
  (
    bcCode_2_5_interleaved,
    bcCode_2_5_industrial,
    bcCode_2_5_matrix,
    bcCode39,
    bcCode39Extended,
    bcCode128A,
    bcCode128B,
    bcCode128C,
    bcCode93,
    bcCode93Extended,
    bcCodeMSI,
    bcCodePostNet,
    bcCodeCodabar,
    bcCodeEAN8,
    bcCodeEAN13,
    bcCodeUPC_A,
    bcCodeUPC_E0,
    bcCodeUPC_E1,
    bcCodeUPC_Supp2,   { UPC 2 цифри }
    bcCodeUPC_Supp5,   { UPC 5 довідкова цифра }
    bcCodeEAN128A,
    bcCodeEAN128B,
    bcCodeEAN128C
  );
  TBarLineType = (white, black, black_half);
{
тільки для внутрішнього використання
}
{
black_half означає чорна лінія 2/5
висоти (використовується для PostNet)
}
  TQRcodeOption = (bcoNone, bcoCode, bcoTyp, bcoBoth);
{ Вид тексту, щоб показати }
  TShowTextPosition =
  (
    stpTopLeft,
    stpTopRight,
    stpTopCenter,
    stpBottomLeft,
    stpBottomRight,
    stpBottomCenter
  );
  TChecksumMethod =
  (
    csmNone,

```

```

csmModulo10
);
TAsQRcode = class(TComponent)
private
  { Декларация-Private }
  FHeight : integer;
  FText : string;
  FTop : integer;
  FLeft : integer;
  FModul : integer;
  FRatio : double;
  FTyp : TQRcodeType;
  FCheckSum:boolean;
  FShowText:TQRcodeOption;
  FAngle : double;
  FColor : TColor;
  FColorBar:TColor;
  FCheckSumMethod : TCheckSumMethod;
  FOnChange : TNotifyEvent;
  modules:array[0..3] of shortint;
  FShowTextFont: TFont;
  FShowTextPosition: TShowTextPosition;
  procedure OneBarProps(code:char; var Width:integer; var lt:TBarLineType);
  procedure DoLines(data:string; Canvas:TCanvas);
  function SetLen(pI:byte):string;
  function Code_2_5_interleaved:string;
  function Code_2_5_industrial:string;
  function Code_2_5_matrix:string;
  function Code_39:string;
  function Code_39Extended:string;
  function Code_128:string;
  function Code_93:string;
  function Code_93Extended:string;
  function Code_MSI:string;
  function Code_PostNet:string;
  function Code_Codabar:string;
  function Code_EAN8:string;
  function Code_EAN13:string;
  function Code_UPC_A:string;
  function Code_UPC_E0:string;
  function Code_UPC_E1:string;
  function Code_Supp5:string;
  function Code_Supp2:string;
  function GetTypText:string;
  procedure MakeModules;
  procedure SetModul(v:integer);
  function GetWidth : integer;
  procedure SetWidth(Value :integer);
  function DoCheckSumming(const data : string):string;
  procedure SetRatio(const Value: Double);
  procedure SetTyp(const Value: TQRcodeType);
  procedure SetAngle(const Value: Double);
  procedure SetText(const Value: string);
  procedure SetShowText(const Value: TQRcodeOption);
  procedure SetTop(const Value: Integer);
  procedure SetLeft(const Value: Integer);
  procedure SetCheckSum(const Value: Boolean);
  procedure SetHeight(const Value: integer);
  function GetCanvasHeight: Integer;
  function GetCanvasWidth: Integer;
  procedure SetShowTextFont(const Value: TFont);
  procedure SetShowTextPosition(const Value: TShowTextPosition);
protected

```

```

    { Protected-декларація }
    function MakeData : string;
    procedure DoChange; virtual;
public
    { Декларація-Public }
    constructor Create(Owner:TComponent); override;
    destructor Destroy; override;
    procedure Assign(Source: TPersistent);override;
    procedure DrawQRcode(Canvas:TCanvas);
    procedure DrawText(Canvas:TCanvas);
    property CanvasHeight :Integer read GetCanvasHeight;
    property CanvasWidth :Integer read GetCanvasWidth;
published
    { Published- Декларація }
{
висота QRcode (Pixel)
}
    property Height : integer read FHeight write SetHeight;
    property Text : string read FText write SetText;
    property Top : Integer read FTop write SetTop;
    property Left : Integer read FLeft write SetLeft;
{
Ширина найменшої лінії в QRcode
}
    property Modul : integer read FModul write SetModul;
    property Ratio : Double read FRatio write SetRatio;
    property Typ : TQRcodeType read FTyp write SetTyp default
bcCode_2_5_interleaved;
{ CheckSum }
    property Checksum:boolean read FChecksum write SetChecksum default FALSE;
    property CheckSumMethod:TChecksumMethod read FChecksumMethod write
FChecksumMethod default csmModulo10;
{ 0 - 360 градуси }
    property Angle :double read FAngle write SetAngle;
    property ShowText:TQRcodeOption read FShowText write SetShowText default
bcNone;
    property ShowTextFont: TFont read FShowTextFont write SetShowTextFont;
    property ShowTextPosition: TShowTextPosition read FShowTextPosition write
SetShowTextPosition default stpTopLeft;
    property Width : integer read GetWidth write SetWidth stored False;
    property Color:TColor read FColor write FColor default clWhite;
    property ColorBar:TColor read FColorBar write FColorBar default clBlack;
    property OnChange:TNotifyEvent read FOnChange write FOnChange;
end;

implementation // реалізація

{$ifdef WIN32} // компіляторні умови
    {$R QRcode.d32}
{$else}
    {$R QRcode.d16}
{$endif}

// використані
Uses // бібліотеки
    WinProcs, WinTypes, SysUtils, bcchksm, math;

{$DEFINE ASSERT_SUPPORTED}
{$IFDEF VER80}
{$UNDEF ASSERT_SUPPORTED}
{$ENDIF}
{$IFDEF VER90}
{$UNDEF ASSERT_SUPPORTED}
{$ENDIF}

```

```

{$IFDEF VER100}
{$UNDEF ASSERT_SUPPORTED}
{$ENDIF}
{$IFDEF VER110}
{$UNDEF ASSERT_SUPPORTED}
{$ENDIF}

type
  TBCdata = record
    Name:string;          { Ім'я QRcode }
    num :Boolean;        { цифрове значення }
  end;
const BCdata:array[bcCode_2_5_interleaved..bcCodeEAN128C] of TBCdata =
  (
    (Name:'2_5_interleaved'; num:True),
    (Name:'2_5_industrial'; num:True),
    (Name:'2_5_matrix'; num:True),
    (Name:'Code39'; num:False),
    (Name:'Code39 Extended'; num:False),
    (Name:'Code128A'; num:False),
    (Name:'Code128B'; num:False),
    (Name:'Code128C'; num:True),
    (Name:'Code93'; num:False),
    (Name:'Code93 Extended'; num:False),
    (Name:'MSI'; num:True),
    (Name:'PostNet'; num:True),
    (Name:'Codebar'; num:False),
    (Name:'EAN8'; num:True),
    (Name:'EAN13'; num:True),
    (Name:'UPC_A'; num:True),
    (Name:'UPC_E0'; num:True),
    (Name:'UPC_E1'; num:True),
    (Name:'UPC Supp2'; num:True),
    (Name:'UPC Supp5'; num:True),
    (Name:'EAN128A'; num:False),
    (Name:'EAN128B'; num:False),
    (Name:'EAN128C'; num:True)
  );
{$ifndef WIN32}

function Trim(const S: string): string; export;
{ Видаляє початкові і кінцеві пробіли з S}
var
  I, L: Integer;
begin
  L := Length(S);
  I := 1;
  while (I <= L) and (S[I] <= ' ') do Inc(I);
  if I > L then Result := '' else
  begin
    while S[L] <= ' ' do Dec(L);
    Result := Copy(S, I, L - I + 1);
  end;
end;
{$endif}

{
перетворить рядок з '321 'до внутрішнього уявлення '715'
потрібна ця функція, тому що деякі моделі
таблиці мають різні формати: '00111 '
перетворює в '05161'
}

```

```

function Convert(const s:string):string;
var
  i, v : integer;
begin
  Result := s;
  for i:=1 to Length(s) do
  begin
    v := ord(s[i]) - 1;
    if odd(i) then
      Inc(v, 5);
    Result[i] := Chr(v);
  end;
end;

function quersumme(x:integer):integer;
var
  sum:integer;
begin
  sum := 0;
  while x > 0 do
  begin
    sum := sum + (x mod 10);
    x := x div 10;
  end;
  result := sum;
end;
{
Поверніть точку, "альфа", Кут повороту
}

function Rotate2D(p:TPoint; alpha:double): TPoint;
var
  sinus, cosinus : Extended;
begin
  // sinus := sin(alpha);
  // cosinus := cos(alpha);
  {
Працює в два рази швидше, ніж грек sin() і cos()
Які за коментовані вище
  SinCos(alpha, sinus, cosinus);
  result.x := Round(p.x*cosinus + p.y*sinus);
  result.y := Round(-p.x*sinus + p.y*cosinus);
end;
{
Перемістити точку "a" на вектор "b"
}

function Translate2D(a, b:TPoint): TPoint;
begin
  result.x := a.x + b.x;
  result.y := a.y + b.y;
end;

procedure Rotate2Darray(p:array of TPoint; alpha:double);
var
  i : Integer;
begin
  for i:=Low(p) to High(p) do
    p[i] := Rotate2D(p[i], alpha);
end;

procedure Translate2Darray(p:array of TPoint; shift:TPoint);
var

```

```

    i : Integer;
begin
    for i:=Low(p) to High(p) do
        p[i] := Translate2D(p[i], shift);
    end;

    {
    Перемістити картинку так що, коли точка повертається, все залишається у видимому
    квадранті
    }

function TranslateQuad2D(const alpha :double; const orgin, point :TPoint):
TPoint;
var
    alphacos: Extended;
    alphasin: Extended;
    moveby: TPoint;
begin
    SinCos(alpha, alphasin, alphacos);
    {
    alphasin := sin(alpha);
    alphacos := cos(alpha);
    }
    if alphasin >= 0 then
    begin
        if alphacos >= 0 then
        begin
            { 1. квадрант }
            moveby.x := 0;
            moveby.y := Round(alphasin*point.x);
        end
        else
        begin
            { 2. квадрант }
            moveby.x := -Round(alphacos*point.x);
            moveby.y := Round(alphasin*point.x - alphacos*point.y);
        end;
    end
    else
    begin
        if alphacos >= 0 then
        begin
            { 4. квадрант }
            moveby.x := -Round(alphasin*point.y);
            moveby.y := 0;
        end
        else
        begin
            { 3. квадрант }
            moveby.x := -Round(alphacos*point.x) - Round(alphasin*point.y);
            moveby.y := -Round(alphacos*point.y);
        end;
    end;
    Result := Translate2D(orgin, moveby);
end;

constructor TQRcode.Create (Owner:TComponent) ;
begin
    inherited Create (owner);
    FAngle := 0.0;
    FRatio := 2.0;
    FModul := 1;
    FTyp := bcCodeEAN13;

```

```

FChecksum := FALSE;
FChecksumMethod := csmModulo10;
FShowText := bcoNone;
FColor := clWhite;
FColorBar := clBlack;
FShowTextFont := TFont.Create;
FShowTextPosition := stpTopLeft;
end;
destructor TQRcode.Destroy;
begin
  FShowTextFont.Free;
  inherited;
end;

procedure TQRcode.Assign(Source: TPersistent);
var
  BSource : TAsQRcode;
begin
  if Source is TAsQRcode then
    begin
      BSource := TAsQRcode(Source);
      FHeight := BSource.FHeight;
      FText := BSource.FText;
      FTop := BSource.FTop;
      FLeft := BSource.FLeft;
      FModul := BSource.FModul;
      FRatio := BSource.FRatio;
      FTyp := BSource.FTyp;
      FChecksum := BSource.FChecksum;
      FShowText := BSource.FShowText;
      FShowTextPosition := BSource.FShowTextPosition;
      FAngle := BSource.FAngle;
      FColor := BSource.FColor;
      FColorBar := BSource.FColorBar;
      FChecksumMethod := BSource.FChecksumMethod;
      FOnChange := BSource.FOnChange;
    end
  else
    end;

function TQRcode.GetTypText:string;
begin
  result := BCdata[FTyp].Name;
end;

{ встановлення Ширини }
procedure TQRcode.SetModul(v:integer);
begin
  if (v >= 1) and (v < 50) then
    begin
      FModul := v;
      DoChange;
    end;
end;

{
Розрахована ширина і тип ліній

```

Код	Колір-лінії	Ширина	Висота
'0'	white	100%	full
'1'	white	100%*Ratio	full
'2'	white	150%*Ratio	full
'3'	white	200%*Ratio	full
'5'	black	100%	full

```

'6'   black           100%*Ratio         full
'7'   black           150%*Ratio         full
'8'   black           200%*Ratio         full
'A'   black           100%                2/5 (used for PostNet)
'B'   black           100%*Ratio         2/5 (used for PostNet)
'C'   black           150%*Ratio         2/5 (used for PostNet)
'D'   black           200%*Ratio         2/5 (used for PostNet)
}
procedure TQRcode.OneBarProps(code:char; var Width:integer; var
lt:TBarLineType);
begin
  case code of
    '0': begin width := modules[0]; lt := white; end;
    '1': begin width := modules[1]; lt := white; end;
    '2': begin width := modules[2]; lt := white; end;
    '3': begin width := modules[3]; lt := white; end;
    '5': begin width := modules[0]; lt := black; end;
    '6': begin width := modules[1]; lt := black; end;
    '7': begin width := modules[2]; lt := black; end;
    '8': begin width := modules[3]; lt := black; end;
    'A': begin width := modules[0]; lt := black_half; end;
    'B': begin width := modules[1]; lt := black_half; end;
    'C': begin width := modules[2]; lt := black_half; end;
    'D': begin width := modules[3]; lt := black_half; end;
  else
    begin
      raise Exception.CreateFmt('%s: внутрішня помилка', [self.ClassName]);
    end;
  end;
end;

function TQRcode.MakeData : string;
var
  i : integer;
begin
  {
  Розрахунок ліній
  }
  MakeModules;
  { числовий тип QRcode }
  if BCdata[Typ].num then
    begin
      FText := Trim(FText);
      { ліквідувати прогалини }
      for i := 1 to Length(Ftext) do
        if (FText[i] > '9') or (FText[i] < '0') then
          raise Exception.Create('QRcode повинен бути числовий');
      end;
      {Отримати структуру QRcode}
      case Typ of
        bcCode_2_5_interleaved: Result := Code_2_5_interleaved;
        bcCode_2_5_industrial:  Result := Code_2_5_industrial;
        bcCode_2_5_matrix:      Result := Code_2_5_matrix;
        bcCode39:                Result := Code_39;
        bcCode39Extended:        Result := Code_39Extended;
        bcCode128A,
        bcCode128B,
        bcCode128C,
        bcCodeEAN128A,
        bcCodeEAN128B,
        bcCodeEAN128C:           Result := Code_128;
        bcCode93:                Result := Code_93;
        bcCode93Extended:        Result := Code_93Extended;
      end;
    end;
  end;
end;

```

```

bcCodeMSI:           Result := Code_MSI;
bcCodePostNet:      Result := Code_PostNet;
bcCodeCodabar:      Result := Code_Codabar;
bcCodeEAN8:         Result := Code_EAN8;
bcCodeEAN13:        Result := Code_EAN13;
bcCodeUPC_A:        Result := Code_UPC_A;
bcCodeUPC_E0:       Result := Code_UPC_E0;
bcCodeUPC_E1:       Result := Code_UPC_E1;
bcCodeUPC_Supp2:    Result := Code_Supp2;
bcCodeUPC_Supp5:    Result := Code_Supp5;
else
  raise Exception.CreateFmt('%s: wrong QRcodeType', [self.ClassName]);
end;
Showmessage (Format ('Data <%s>', [Result]));

end;

function TQRcode.GetWidth:integer;
var
  data : string;
  i : integer;
  w : integer;
  lt : TBarLineType;
begin
  Result := 0;
  { отримати QRcode візерунок}
  data := MakeData;
  for i:=1 to Length(data) do
  begin
    OneBarProps (data[i], w, lt);
    Inc (Result, w);
  end;
end;

procedure TQRcode.SetWidth(Value :integer);
var
  data : string;
  i : integer;
  w, wtotal : integer;
  lt : TBarLineType;
begin
  wtotal := 0;
  { отримати QRcode візерунку }
  data := MakeData;
  for i:=1 to Length(data) do
  begin
    OneBarProps (data[i], w, lt);
    Inc (wtotal, w);
  end;
  {
  wtotal: поточна ширина QRcode
  Value : new нова ширина QRcode
  }
  if wtotal > 0 then
    SetModul ((FModul * Value) div wtotal);
end;

function TQRcode.DoCheckSumming(const data : string):string;
begin
  case FCheckSumMethod of
    csmNone:
      Result := data;
  end;
end;

```

```

    csmModulo10:
        Result := CheckSumModulo10(data);
    end;
end;
{L1  S1  L2  S2}
const tabelle_EAN_A:array['0'..'9'] of string =
(
('2605'),    { 0 }
('1615'),    { 1 }
('1516'),    { 2 }
('0805'),    { 3 }
('0526'),    { 4 }
('0625'),    { 5 }
('0508'),    { 6 }
('0706'),    { 7 }
('0607'),    { 8 }
('2506')     { 9 }
);
{S1  L1  S2  L2}
const tabelle_EAN_C:array['0'..'9'] of string =
(
('7150' ),   { 0 }
('6160' ),   { 1 }
('6061' ),   { 2 }
('5350' ),   { 3 }
('5071' ),   { 4 }
('5170' ),   { 5 }
('5053' ),   { 6 }
('5251' ),   { 7 }
('5152' ),   { 8 }
('7051' )    { 9 }
);

function TQRcode.Code_EAN8:string;
var
    i : integer;
    tmp : String;
begin
    if FCheckSum then
        begin
            tmp := SetLen(7);
            tmp := DoCheckSumming(copy(tmp,length(tmp)-6,7));
        end
    else
        tmp := SetLen(8);
    {$IFDEF ASSERT_SUPPORTED}
        Assert(Length(tmp)=8, 'Invalid Text len (EAN8)');
    {$ENDIF}
    result := '505';    {Початковийкод}
    for i:=1 to 4 do
        result := result + tabelle_EAN_A[tmp[i]] ;
        result := result + '05050';
    for i:=5 to 8 do
        result := result + tabelle_EAN_C[tmp[i]] ;
        result := result + '505';    {Стопкод}
    end;
{L1  S1  L2  S2}
const tabelle_EAN_B:array['0'..'9'] of string =
(
('0517'),    { 0 }
('0616'),    { 1 }
('1606'),    { 2 }
('0535'),    { 3 }

```

```

('1705'),    { 4 }
('0715'),    { 5 }
('3505'),    { 6 }
('1525'),    { 7 }
('2515'),    { 8 }
('1507')     { 9 }
);
const tabelle_ParityEAN13:array[0..9, 1..6] of char =
(
('A', 'A', 'A', 'A', 'A', 'A'),    { 0 }
('A', 'A', 'B', 'A', 'B', 'B'),    { 1 }
('A', 'A', 'B', 'B', 'A', 'B'),    { 2 }
('A', 'A', 'B', 'B', 'B', 'A'),    { 3 }
('A', 'B', 'A', 'A', 'B', 'B'),    { 4 }
('A', 'B', 'B', 'A', 'A', 'B'),    { 5 }
('A', 'B', 'B', 'B', 'A', 'A'),    { 6 }
('A', 'B', 'A', 'B', 'A', 'B'),    { 7 }
('A', 'B', 'A', 'B', 'B', 'A'),    { 8 }
('A', 'B', 'B', 'A', 'B', 'A')     { 9 }
);

function TQRcode.Code_EAN13:string;
var
  i, LK: integer;
  tmp : String;
begin
  if FCheckSum then
    begin
      tmp := SetLen(12);
      tmp := DoCheckSumming(tmp);
    end
  else
    tmp := SetLen(13);
  {$IFDEF ASSERT_SUPPORTED}
  Assert(Length(tmp) = 13, 'Invalid Text len (EAN13)');
  {$ENDIF}
  LK := StrToInt(tmp[1]);
  tmp := copy(tmp,2,12);
  result := '505';    {Startcode}
  for i:=1 to 6 do
    begin
      case tabelle_ParityEAN13[LK,i] of
        'A' : result := result + tabelle_EAN_A[tmp[i]];
        'B' : result := result + tabelle_EAN_B[tmp[i]] ;
        'C' : result := result + tabelle_EAN_C[tmp[i]] ;
      end;
    end;
  result := result + '05050';    {Center Guard Pattern}
  for i:=7 to 12 do
    result := result + tabelle_EAN_C[tmp[i]] ;
    result := result + '505';    {Stopcode}
  end;
  {Pattern for QRcode 2 of 5}
  const tabelle_2_5:array['0'..'9', 1..5] of char =
  (
('0', '0', '1', '1', '0'),    {'0'}
('1', '0', '0', '0', '1'),    {'1'}
('0', '1', '0', '0', '1'),    {'2'}
('1', '1', '0', '0', '0'),    {'3'}
('0', '0', '1', '0', '1'),    {'4'}
('1', '0', '1', '0', '0'),    {'5'}
('0', '1', '1', '0', '0'),    {'6'}
('0', '0', '0', '1', '1'),    {'7'}

```

```

('1', '0', '0', '1', '0'),    {'8'}
('0', '1', '0', '1', '0')    {'9'}
);

function TQRcode.Code_2_5_interleaved:string;
var
  i, j: integer;
  c : char;
begin
  result := '5050';    {Startcode}
  for i:=1 to Length(FText) div 2 do
  begin
    for j:= 1 to 5 do
    begin
      if tabelle_2_5[FText[i*2-1], j] = '1' then
        c := '6'
      else
        c := '5';
      result := result + c;
      if tabelle_2_5[FText[i*2], j] = '1' then
        c := '1'
      else
        c := '0';
      result := result + c;
    end;
  end;
  result := result + '605';    {Stopcode}
end;

function TQRcode.Code_2_5_industrial:string;
var
  i, j: integer;
begin
  result := '606050';    {Startcode}
  for i:=1 to Length(FText) do
  begin
    for j:= 1 to 5 do
    begin
      if tabelle_2_5[FText[i], j] = '1' then
        result := result + '60'
      else
        result := result + '50';
    end;
  end;
  result := result + '605060';    {Stopcode}
end;

function TQRcode.Code_2_5_matrix:string;
var
  i, j: integer;
  c :char;
begin
  result := '705050';    {Startcode}
  for i:=1 to Length(FText) do
  begin
    for j:= 1 to 5 do
    begin
      if tabelle_2_5[FText[i], j] = '1' then
        c := '1'
      else
        c := '0';
      { Якщо непарний, то зробити заборону }
      if odd(j) then

```

```

        c := chr(ord(c)+5);
        result := result + c;
    end;
    result := result + '0';
end;
result := result + '70505';    {Стопкод}
end;

```

```
function TQRcode.Code_39:string;
```

```
type TCode39 =
```

```
    record
```

```
        c : char;
```

```
        data : array[0..9] of char;
```

```
        chk: shortint;
```

```
    end;
```

```
const tabelle_39: array[0..43] of TCode39 = (
```

```

    ( c:'0'; data:'505160605'; chk:0 ),
    ( c:'1'; data:'605150506'; chk:1 ),
    ( c:'2'; data:'506150506'; chk:2 ),
    ( c:'3'; data:'606150505'; chk:3 ),
    ( c:'4'; data:'505160506'; chk:4 ),
    ( c:'5'; data:'605160505'; chk:5 ),
    ( c:'6'; data:'506160505'; chk:6 ),
    ( c:'7'; data:'505150606'; chk:7 ),
    ( c:'8'; data:'605150605'; chk:8 ),
    ( c:'9'; data:'506150605'; chk:9 ),
    ( c:'A'; data:'605051506'; chk:10),
    ( c:'B'; data:'506051506'; chk:11),
    ( c:'C'; data:'606051505'; chk:12),
    ( c:'D'; data:'505061506'; chk:13),
    ( c:'E'; data:'605061505'; chk:14),
    ( c:'F'; data:'506061505'; chk:15),
    ( c:'G'; data:'505051606'; chk:16),
    ( c:'H'; data:'605051605'; chk:17),
    ( c:'I'; data:'506051605'; chk:18),
    ( c:'J'; data:'505061605'; chk:19),
    ( c:'K'; data:'605050516'; chk:20),
    ( c:'L'; data:'506050516'; chk:21),
    ( c:'M'; data:'606050515'; chk:22),
    ( c:'N'; data:'505060516'; chk:23),
    ( c:'O'; data:'605060515'; chk:24),
    ( c:'P'; data:'506060515'; chk:25),
    ( c:'Q'; data:'505050616'; chk:26),
    ( c:'R'; data:'605050615'; chk:27),
    ( c:'S'; data:'506050615'; chk:28),
    ( c:'T'; data:'505060615'; chk:29),
    ( c:'U'; data:'615050506'; chk:30),
    ( c:'V'; data:'516050506'; chk:31),
    ( c:'W'; data:'616050505'; chk:32),
    ( c:'X'; data:'515060506'; chk:33),
    ( c:'Y'; data:'615060505'; chk:34),
    ( c:'Z'; data:'516060505'; chk:35),
    ( c:'-'; data:'515050606'; chk:36),
    ( c:'.'; data:'615050605'; chk:37),
    ( c:' '; data:'516050605'; chk:38),
    ( c:'*'; data:'515060605'; chk:0 ),
    ( c:'$'; data:'515151505'; chk:39),
    ( c:'/'; data:'515150515'; chk:40),
    ( c:'+'; data:'515051515'; chk:41),
    ( c:'%'; data:'505151515'; chk:42)
);

```

```
function FindIdx(z:char):integer;
```

```

var
  i:integer;
begin
  for i:=0 to High(tabelle_39) do
  begin
    if z = tabelle_39[i].c then
    begin
      result := i;
      exit;
    end;
  end;
  result := -1;
end;
var
  i, idx : integer;
  checksum:integer;
begin
  checksum := 0;
  {Старткод}
  result := tabelle_39[FindIdx('*')].data + '0';
  for i:=1 to Length(FText) do
  begin
    idx := FindIdx(FText[i]);
    if idx < 0 then
      continue;
    result := result + tabelle_39[idx].data + '0';
    Inc(checksum, tabelle_39[idx].chk);
  end;

  {Калькуляція перевірконої суми даних - Checksum Data}
  if FCheckSum then
  begin
    checksum := checksum mod 43;
    for i:=0 to High(tabelle_39) do
      if checksum = tabelle_39[i].chk then
      begin
        result := result + tabelle_39[i].data + '0';
        break;
      end;
    end;
  end;
  {Стрпкод}
  result := result + tabelle_39[FindIdx('*')].data;
end;

```

```

function TQRcode.Code_39Extended:string;
const code39x : array[0..127] of string[2] =
(
  ('%U'), ('$A'), ('$B'), ('$C'), ('$D'), ('$E'), ('$F'), ('$G'),
  ('$H'), ('$I'), ('$J'), ('$K'), ('$L'), ('$M'), ('$N'), ('$O'),
  ('$P'), ('$Q'), ('$R'), ('$S'), ('$T'), ('$U'), ('$V'), ('$W'),
  ('$X'), ('$Y'), ('$Z'), ('%A'), ('%B'), ('%C'), ('%D'), ('%E'),
  (' '), ('/A'), ('/B'), ('/C'), ('/D'), ('/E'), ('/F'), ('/G'),
  ('/H'), ('/I'), ('/J'), ('/K'), ('/L'), ('/M'), ('/N'), ('/O'),
  ('0'), ('1'), ('2'), ('3'), ('4'), ('5'), ('6'), ('7'),
  ('8'), ('9'), ('/Z'), ('%F'), ('%G'), ('%H'), ('%I'), ('%J'),
  ('%V'), ('A'), ('B'), ('C'), ('D'), ('E'), ('F'), ('G'),
  ('H'), ('I'), ('J'), ('K'), ('L'), ('M'), ('N'), ('O'),
  ('P'), ('Q'), ('R'), ('S'), ('T'), ('U'), ('V'), ('W'),
  ('X'), ('Y'), ('Z'), ('%K'), ('%L'), ('%M'), ('%N'), ('%O'),
  ('%W'), ('+A'), ('+B'), ('+C'), ('+D'), ('+E'), ('+F'), ('+G'),
  ('+H'), ('+I'), ('+J'), ('+K'), ('+L'), ('+M'), ('+N'), ('+O'),

```

```

    ('+P'), ('+Q'), ('+R'), ('+S'), ('+T'), ('+U'), ('+V'), ('+W'),
    ('+X'), ('+Y'), ('+Z'), ('%P'), ('%Q'), ('%R'), ('%S'), ('%T')
  );
var
  save:string;
  i : integer;
begin
  save := FText;
  FText := '';
  for i:=1 to Length(save) do
  begin
    if ord(save[i]) <= 127 then
      FText := FText + code39x[ord(save[i])];
    end;
  result := Code_39;
  FText := save;
end;
function TQRcode.Code_128:string;
type TCode128 =
  record
    a, b : char;
    c : string[2];
    data : string[6];
  end;
const tabelle_128: array[0..102] of TCode128 = (
  ( a:' ' ; b:' ' ; c:'00' ; data:'212222' ),
  ( a:'!' ; b:'!' ; c:'01' ; data:'222122' ),
  ( a:'"' ; b:'"' ; c:'02' ; data:'222221' ),
  ( a:'#' ; b:'#' ; c:'03' ; data:'121223' ),
  ( a:'$' ; b:'$' ; c:'04' ; data:'121322' ),
  ( a:'%' ; b:'%' ; c:'05' ; data:'131222' ),
  ( a:'&' ; b:'&' ; c:'06' ; data:'122213' ),
  ( a:'"' ; b:'"' ; c:'07' ; data:'122312' ),
  ( a:'(' ; b:'(' ; c:'08' ; data:'132212' ),
  ( a:')' ; b:')' ; c:'09' ; data:'221213' ),
  ( a:'*' ; b:'*' ; c:'10' ; data:'221312' ),
  ( a:'+' ; b:'+' ; c:'11' ; data:'231212' ),
  ( a:',' ; b:',' ; c:'12' ; data:'112232' ),
  ( a:'-' ; b:'-' ; c:'13' ; data:'122132' ),
  ( a:'.' ; b:'.' ; c:'14' ; data:'122231' ),
  ( a:'/' ; b:'/' ; c:'15' ; data:'113222' ),
  ( a:'0' ; b:'0' ; c:'16' ; data:'123122' ),
  ( a:'1' ; b:'1' ; c:'17' ; data:'123221' ),
  ( a:'2' ; b:'2' ; c:'18' ; data:'223211' ),
  ( a:'3' ; b:'3' ; c:'19' ; data:'221132' ),
  ( a:'4' ; b:'4' ; c:'20' ; data:'221231' ),
  ( a:'5' ; b:'5' ; c:'21' ; data:'213212' ),
  ( a:'6' ; b:'6' ; c:'22' ; data:'223112' ),
  ( a:'7' ; b:'7' ; c:'23' ; data:'312131' ),
  ( a:'8' ; b:'8' ; c:'24' ; data:'311222' ),
  ( a:'9' ; b:'9' ; c:'25' ; data:'321122' ),
  ( a':' ; b':' ; c:'26' ; data:'321221' ),
  ( a:';' ; b:';' ; c:'27' ; data:'312212' ),
  ( a:'<' ; b:'<' ; c:'28' ; data:'322112' ),
  ( a:'=' ; b:'=' ; c:'29' ; data:'322211' ),
  ( a:'>' ; b:'>' ; c:'30' ; data:'212123' ),
  ( a:'?' ; b:'?' ; c:'31' ; data:'212321' ),
  ( a:'@' ; b:'@' ; c:'32' ; data:'232121' ),
  ( a:'A' ; b:'A' ; c:'33' ; data:'111323' ),
  ( a:'B' ; b:'B' ; c:'34' ; data:'131123' ),
  ( a:'C' ; b:'C' ; c:'35' ; data:'131321' ),
  ( a:'D' ; b:'D' ; c:'36' ; data:'112313' ),
  ( a:'E' ; b:'E' ; c:'37' ; data:'132113' ),

```

```
( a:'F'; b:'F'; c:'38'; data:'132311' ),
( a:'G'; b:'G'; c:'39'; data:'211313' ),
( a:'H'; b:'H'; c:'40'; data:'231113' ),
( a:'I'; b:'I'; c:'41'; data:'231311' ),
( a:'J'; b:'J'; c:'42'; data:'112133' ),
( a:'K'; b:'K'; c:'43'; data:'112331' ),
( a:'L'; b:'L'; c:'44'; data:'132131' ),
( a:'M'; b:'M'; c:'45'; data:'113123' ),
( a:'N'; b:'N'; c:'46'; data:'113321' ),
( a:'O'; b:'O'; c:'47'; data:'133121' ),
( a:'P'; b:'P'; c:'48'; data:'313121' ),
( a:'Q'; b:'Q'; c:'49'; data:'211331' ),
( a:'R'; b:'R'; c:'50'; data:'231131' ),
( a:'S'; b:'S'; c:'51'; data:'213113' ),
( a:'T'; b:'T'; c:'52'; data:'213311' ),
( a:'U'; b:'U'; c:'53'; data:'213131' ),
( a:'V'; b:'V'; c:'54'; data:'311123' ),
( a:'W'; b:'W'; c:'55'; data:'311321' ),
( a:'X'; b:'X'; c:'56'; data:'331121' ),
( a:'Y'; b:'Y'; c:'57'; data:'312113' ),
( a:'Z'; b:'Z'; c:'58'; data:'312311' ),
( a:'['; b:'['; c:'59'; data:'332111' ),
( a:'\'; b:'\'; c:'60'; data:'314111' ),
( a:']'; b:']'; c:'61'; data:'221411' ),
( a:'^'; b:'^'; c:'62'; data:'431111' ),
( a:'_'; b:'_'; c:'63'; data:'111224' ),
( a:'#0'; b:'0'; c:'64'; data:'111422' ),
( a:'#1'; b:'a'; c:'65'; data:'121124' ),
( a:'#2'; b:'b'; c:'66'; data:'121421' ),
( a:'#3'; b:'c'; c:'67'; data:'141122' ),
( a:'#4'; b:'d'; c:'68'; data:'141221' ),
( a:'#5'; b:'e'; c:'69'; data:'112214' ),
( a:'#6'; b:'f'; c:'70'; data:'112412' ),
( a:'#7'; b:'g'; c:'71'; data:'122114' ),
( a:'#8'; b:'h'; c:'72'; data:'122411' ),
( a:'#9'; b:'i'; c:'73'; data:'142112' ),
( a:'#10'; b:'j'; c:'74'; data:'142211' ),
( a:'#11'; b:'k'; c:'75'; data:'241211' ),
( a:'#12'; b:'l'; c:'76'; data:'221114' ),
( a:'#13'; b:'m'; c:'77'; data:'413111' ),
( a:'#14'; b:'n'; c:'78'; data:'241112' ),
( a:'#15'; b:'o'; c:'79'; data:'134111' ),
( a:'#16'; b:'p'; c:'80'; data:'111242' ),
( a:'#17'; b:'q'; c:'81'; data:'121142' ),
( a:'#18'; b:'r'; c:'82'; data:'121241' ),
( a:'#19'; b:'s'; c:'83'; data:'114212' ),
( a:'#20'; b:'t'; c:'84'; data:'124112' ),
( a:'#21'; b:'u'; c:'85'; data:'124211' ),
( a:'#22'; b:'v'; c:'86'; data:'411212' ),
( a:'#23'; b:'w'; c:'87'; data:'421112' ),
( a:'#24'; b:'x'; c:'88'; data:'421211' ),
( a:'#25'; b:'y'; c:'89'; data:'212141' ),
( a:'#26'; b:'z'; c:'90'; data:'214121' ),
( a:'#27'; b:'{'; c:'91'; data:'412121' ),
( a:'#28'; b:'|'; c:'92'; data:'111143' ),
( a:'#29'; b:'}'; c:'93'; data:'111341' ),
( a:'#30'; b:'~'; c:'94'; data:'131141' ),
( a:'#31'; b:' '; c:'95'; data:'114113' ),
( a:' ' ; b:' ' ; c:'96'; data:'114311' ),
( a:' ' ; b:' ' ; c:'97'; data:'411113' ),
( a:' ' ; b:' ' ; c:'98'; data:'411311' ),
( a:' ' ; b:' ' ; c:'99'; data:'113141' ),
( a:' ' ; b:' ' ; c:' ' ; data:'114131' ),
```

```

    ( a:' '; b:' '; c:' '; data:'311141' ),
    ( a:' '; b:' '; c:' '; data:'411131' )
  );
StartA = '211412';
StartB = '211214';
StartC = '211232';
Stop   = '2331112';

function Find_Code128AB(c:char):integer;
var
  i:integer;
  v:char;
begin
  for i:=0 to High(tabelle_128) do
  begin
    if FTyp = bcCode128A then
      v := tabelle_128[i].a
    else
      v := tabelle_128[i].b;
    if c = v then
      begin
        result := i;
        exit;
      end;
    end;
  result := -1;
end;

function Find_Code128C(c:string):integer;
var i:integer;
begin
  for i:=0 to High(tabelle_128) do begin
    if tabelle_128[i].c = c then begin
      result := i;
      exit;
    end;
  end;
  result := -1;
end;
var i, j, idx: integer;
startcode:string;
checksum : integer;
codeword_pos : integer;
begin
  case FTyp of
    bcCode128A, bcCodeEAN128A:
      begin checksum := 103; startcode:= StartA; end;
    bcCode128B, bcCodeEAN128B:
      begin checksum := 104; startcode:= StartB; end;
    bcCode128C, bcCodeEAN128C:
      begin checksum := 105; startcode:= StartC; end;
    else
      raise Exception.CreateFmt('%s: wrong QRcodeType in Code_128',
[Self.ClassName]);
  end;
  result := startcode;      {Старткод}
  codeword_pos := 1;
  case FTyp of
    bcCodeEAN128A,
    bcCodeEAN128B,
    bcCodeEAN128C:
      begin

```

```

        result := result + tabelle_128[102].data;
        Inc(checksum, 102*codeword_pos);
        Inc(codeword_pos);
    }
    Перевірка контрольної суми в кінці рядка
}

    if FCheckSum then FText:=DoCheckSumming(FTEXT);
    end;
end;
if (FTyp = bcCode128C) or (FTyp = bccodeEAN128C) then
begin
    if (Length(FText) mod 2<>0) then FText:='0'+FText;
    for i:=1 to (Length(FText) div 2) do
    begin
        j:=(i-1)*2+1;
        idx:=Find_Code128C(copy(Ftext,j,2));
        if idx < 0 then idx := Find_Code128C('00');
        result := result + tabelle_128[idx].data;
        Inc(checksum, idx*codeword_pos);
        Inc(codeword_pos);
    end;
end
else
    for i:=1 to Length(FText) do
    begin
        idx := Find_Code128AB(FText[i]);
        if idx < 0 then
            idx := Find_Code128AB(' ');
        result := result + tabelle_128[idx].data;
        Inc(checksum, idx*codeword_pos);
        Inc(codeword_pos);
    end;
checksum := checksum mod 103;
result := result + tabelle_128[checksum].data;
result := result + Stop;      {Стопкод}
Result := Convert(Result);
end;

function TQRcode.Code_93:string;
type TCode93 =
    record
        c : char;
        data : array[0..5] of char;
    end;

const tabelle_93: array[0..46] of TCode93 = (
    ( c:'0'; data:'131112' ),
    ( c:'1'; data:'111213' ),
    ( c:'2'; data:'111312' ),
    ( c:'3'; data:'111411' ),
    ( c:'4'; data:'121113' ),
    ( c:'5'; data:'121212' ),
    ( c:'6'; data:'121311' ),
    ( c:'7'; data:'111114' ),
    ( c:'8'; data:'131211' ),
    ( c:'9'; data:'141111' ),
    ( c:'A'; data:'211113' ),
    ( c:'B'; data:'211212' ),
    ( c:'C'; data:'211311' ),
    ( c:'D'; data:'221112' ),
    ( c:'E'; data:'221211' ),
    ( c:'F'; data:'231111' ),

```

```

( c:'G'; data:'112113' ),
( c:'H'; data:'112212' ),
( c:'I'; data:'112311' ),
( c:'J'; data:'122112' ),
( c:'K'; data:'132111' ),
( c:'L'; data:'111123' ),
( c:'M'; data:'111222' ),
( c:'N'; data:'111321' ),
( c:'O'; data:'121122' ),
( c:'P'; data:'131121' ),
( c:'Q'; data:'212112' ),
( c:'R'; data:'212211' ),
( c:'S'; data:'211122' ),
( c:'T'; data:'211221' ),
( c:'U'; data:'221121' ),
( c:'V'; data:'222111' ),
( c:'W'; data:'112122' ),
( c:'X'; data:'112221' ),
( c:'Y'; data:'122121' ),
( c:'Z'; data:'123111' ),
( c:'-'; data:'121131' ),
( c:'.'; data:'311112' ),
( c:' '; data:'311211' ),
( c:'$'; data:'321111' ),
( c:'/'; data:'112131' ),
( c:'+'; data:'113121' ),
( c:'%'; data:'211131' ),
( c:'['; data:'121221' ), { використовується тільки для розширеного коду 93}
( c:']'; data:'312111' ),
( c:'{'; data:'311121' ),
( c:'}'; data:'122211' )
);

function Find_Code93(c:char):integer;
var
  i:integer;
begin
  for i:=0 to High(tabelle_93) do
  begin
    if c = tabelle_93[i].c then
    begin
      result := i;
      exit;
    end;
  end;
  result := -1;
end;

var
  i, idx : integer;
  checkC, checkK, {контрольна сума}
  weightC, weightK : integer;
begin
  result := '111141'; {старткод}
  for i:=1 to Length(FText) do
  begin
    idx := Find_Code93(FText[i]);
    if idx < 0 then
      raise Exception.CreateFmt('%s: bad Data <%s>', [self.ClassName,FText]);
    result := result + tabelle_93[idx].data;
  end;
  checkC := 0;
  checkK := 0;

```

```

weightC := 1;
weightK := 2;
for i:=Length(FText) downto 1 do
begin
  idx := Find_Code93(FText[i]);
  Inc(checkC, idx*weightC);
  Inc(checkK, idx*weightK);
  Inc(weightC);
  if weightC > 20 then weightC := 1;
  Inc(weightK);
  if weightK > 15 then weightK:= 1;
end;
Inc(checkK, checkC);
checkC := checkC mod 47;
checkK := checkK mod 47;
result := result + tabelle_93[checkC].data +
  tabelle_93[checkK].data;
result := result + '1111411'; {стопкод}
Result := Convert(Result);
end;

```

```

function TQRcode.Code_93Extended:string;
const code93x : array[0..127] of string[2] =
(
  (']U'), ('[A'), ('[B'), ('[C'), ('[D'), ('[E'), ('[F'), ('[G'),
  ('[H'), ('[I'), ('[J'), ('[K'), ('[L'), ('[M'), ('[N'), ('[O'),
  ('[P'), ('[Q'), ('[R'), ('[S'), ('[T'), ('[U'), ('[V'), ('[W'),
  ('[X'), ('[Y'), ('[Z'), (']A'), (']B'), (']C'), (']D'), (']E'),
  (' '), ('{A'), ('{B'), ('{C'), ('{D'), ('{E'), ('{F'), ('{G'),
  ('{H'), ('{I'), ('{J'), ('{K'), ('{L'), ('{M'), ('{N'), ('{O'),
  ('0'), ('1'), ('2'), ('3'), ('4'), ('5'), ('6'), ('7'),
  ('8'), ('9'), ('{Z'), (']F'), (']G'), (']H'), (']I'), (']J'),
  (']V'), ('A'), ('B'), ('C'), ('D'), ('E'), ('F'), ('G'),
  ('H'), ('I'), ('J'), ('K'), ('L'), ('M'), ('N'), ('O'),
  ('P'), ('Q'), ('R'), ('S'), ('T'), ('U'), ('V'), ('W'),
  ('X'), ('Y'), ('Z'), (']K'), (']L'), (']M'), (']N'), (']O'),
  (']W'), ('}A'), ('}B'), ('}C'), ('}D'), ('}E'), ('}F'), ('}G'),
  ('}H'), ('}I'), ('}J'), ('}K'), ('}L'), ('}M'), ('}N'), ('}O'),
  ('}P'), ('}Q'), ('}R'), ('}S'), ('}T'), ('}U'), ('}V'), ('}W'),
  ('}X'), ('}Y'), ('}Z'), (']P'), (']Q'), (']R'), (']S'), (']T')
);
var
  save : string;
  i : integer;
begin
  save := FText;
  FText := '';
  for i:=1 to Length(save) do
  begin
    if ord(save[i]) <= 127 then
      FText := FText + code93x[ord(save[i])];
    end;
  Showmessage(Format('Text: <%s>', [FText]));
  result := Code_93;
  FText := save;
end;

```

```

function TQRcode.Code_MSI:string;
const tabelle_MSI:array['0'..'9'] of string[8] =
(
  ('51515151'), {'0'}
  ('51515160'), {'1'}
  ('51516051'), {'2'}

```

```

( '51516060' ),    {'3'}
( '51605151' ),    {'4'}
( '51605160' ),    {'5'}
( '51606051' ),    {'6'}
( '51606060' ),    {'7'}
( '60515151' ),    {'8'}
( '60515160' )     {'9'}
);

var
  i:integer;
  check_even, check_odd, checksum:integer;
begin
  result := '60';    {Startcode}
  check_even := 0;
  check_odd := 0;
  for i:=1 to Length(FText) do
  begin
    if odd(i-1) then
      check_odd := check_odd*10+ord(FText[i])
    else
      check_even := check_even+ord(FText[i]);
      result := result + tabelle_MSI[FText[i]];
    end;
  checksum := quersumme(check_odd*2) + check_even;
  checksum := checksum mod 10;
  if checksum > 0 then
    checksum := 10-checksum;
  result := result + tabelle_MSI[chr(ord('0')+checksum)];
  result := result + '515'; {Stopcode}
end;

function TQRcode.Code_PostNet:string;
const tabelle_PostNet:array['0'..'9'] of string[10] =
(
( '5151A1A1A1' ),    {'0'}
( 'A1A1A15151' ),    {'1'}
( 'A1A151A151' ),    {'2'}
( 'A1A15151A1' ),    {'3'}
( 'A151A1A151' ),    {'4'}
( 'A151A151A1' ),    {'5'}
( 'A15151A1A1' ),    {'6'}
( '51A1A1A151' ),    {'7'}
( '51A1A151A1' ),    {'8'}
( '51A151A1A1' )     {'9'}
);
var
  i:integer;
begin
  result := '51';
  for i:=1 to Length(FText) do
  begin
    result := result + tabelle_PostNet[FText[i]];
  end;
  result := result + '5';
end;

function TQRcode.Code_Codabar:string;
type TCodabar =
  record
    c : char;
    data : array[0..6] of char;
  end;
end;

```

```

const tabelle_cb: array[0..19] of TCodabar = (
  ( c:'1'; data:'5050615' ),
  ( c:'2'; data:'5051506' ),
  ( c:'3'; data:'6150505' ),
  ( c:'4'; data:'5060515' ),
  ( c:'5'; data:'6050515' ),
  ( c:'6'; data:'5150506' ),
  ( c:'7'; data:'5150605' ),
  ( c:'8'; data:'5160505' ),
  ( c:'9'; data:'6051505' ),
  ( c:'0'; data:'5050516' ),
  ( c:'-'; data:'5051605' ),
  ( c:'$'; data:'5061505' ),
  ( c:':'; data:'6050606' ),
  ( c:'/'; data:'6060506' ),
  ( c:'.'; data:'6060605' ),
  ( c:'+'; data:'5060606' ),
  ( c:'A'; data:'5061515' ),
  ( c:'B'; data:'5151506' ),
  ( c:'C'; data:'5051516' ),
  ( c:'D'; data:'5051615' )
);

function Find_Codabar(c:char):integer;
var
  i:integer;
begin
  for i:=0 to High(tabelle_cb) do
  begin
    if c = tabelle_cb[i].c then
    begin
      result := i;
      exit;
    end;
  end;
  result := -1;
end;
var
  i, idx : integer;
begin
  result := tabelle_cb[Find_Codabar('A')].data + '0';
  for i:=1 to Length(FText) do
  begin
    idx := Find_Codabar(FText[i]);
    result := result + tabelle_cb[idx].data + '0';
  end;
  result := result + tabelle_cb[Find_Codabar('B')].data;
end;

{допоміжна функція }
function TQRcode.SetLen(pI:byte):string;
begin
  Result := StringOfChar('0', pI-Length(FText)) + FText;
end;

function TQRcode.Code_UPC_A:string;
var
  i : integer;
  tmp : String;
begin
  FText := SetLen(12);
  if FCheckSum then tmp:=DoCheckSumming(copy(FText,1,11));
  if FCheckSum then FText:=tmp else tmp:=FText;

```

```

    result := '505';    {Startcode}
    for i:=1 to 6 do
        result := result + tabelle_EAN_A[tmp[i]];
    result := result + '05050';    { розділювач }
    for i:=7 to 12 do
        result := result + tabelle_EAN_C[tmp[i]];
    result := result + '505';    {Stopcode}
end;
const tabelle_UPC_E0:array['0'..'9', 1..6] of char =
(
('E', 'E', 'E', 'o', 'o', 'o' ),    { 0 }
('E', 'E', 'o', 'E', 'o', 'o' ),    { 1 }
('E', 'E', 'o', 'o', 'E', 'o' ),    { 2 }
('E', 'E', 'o', 'o', 'o', 'E' ),    { 3 }
('E', 'o', 'E', 'E', 'o', 'o' ),    { 4 }
('E', 'o', 'o', 'E', 'E', 'o' ),    { 5 }
('E', 'o', 'o', 'o', 'E', 'E' ),    { 6 }
('E', 'o', 'E', 'o', 'E', 'o' ),    { 7 }
('E', 'o', 'E', 'o', 'o', 'E' ),    { 8 }
('E', 'o', 'o', 'E', 'o', 'E' )    { 9 }
);

function TQRcode.Code_UPC_E0:string;
var i,j : integer;
    tmp : String;
    c : char;
begin
    FText := SetLen(7);
    tmp:=DoCheckSumming(copy(FText,1,6));
    c:=tmp[7];
    if FCheckSum then FText:=tmp else tmp := FText;
    result := '505';    {Startcode}
    for i:=1 to 6 do
    begin
        if tabelle_UPC_E0[c,i]='E' then
        begin
            for j:= 1 to 4 do result := result + tabelle_EAN_C[tmp[i],5-j];
        end
        else
        begin
            result := result + tabelle_EAN_A[tmp[i]];
        end;
    end;
    result := result + '050505';    {Stopcode}
end;

function TQRcode.Code_UPC_E1:string;
var i,j : integer;
    tmp : String;
    c : char;
begin
    FText := SetLen(7);
    tmp:=DoCheckSumming(copy(FText,1,6));
    c:=tmp[7];
    if FCheckSum then FText:=tmp else tmp := FText;
    result := '505';    {Startcode}
    for i:=1 to 6 do
    begin
        if tabelle_UPC_E0[c,i]='E' then
        begin
            result := result + tabelle_EAN_A[tmp[i]];
        end
        else

```

```

begin
  for j:= 1 to 4 do result := result + tabelle_EAN_C[tmp[i],5-j];
end;
end;
result := result + '050505';    {Stopcode}
end;

{ функція допомоги }
function getSupp(Nr : String) : String;
var i,fak,sum : Integer;
    tmp      : String;
begin
  sum := 0;
  tmp := copy(nr,1,Length(Nr)-1);
  fak := Length(tmp);
  for i:=1 to length(tmp) do
begin
  if (fak mod 2) = 0 then
    sum := sum + (StrToInt(tmp[i])*9)
  else
    sum := sum + (StrToInt(tmp[i])*3);
  dec(fak);
end;
sum:=(sum mod 10) mod 10) mod 10;
result := tmp+IntToStr(sum);
end;

function TQRcode.Code_Supp5:string;
var
  i,j : integer;
  tmp : String;
  c   : char;
begin
  FText := SetLen(5);
  tmp:=getSupp(copy(FText,1,5)+'0');
  c:=tmp[6];
  if FCheckSum then FText:=tmp else tmp := FText;
  result := '506';    {Startcode}
  for i:=1 to 5 do
begin
  if tabelle_UPC_E0[c,(6-5)+i]='E' then
begin
  for j:= 1 to 4 do result := result + tabelle_EAN_C[tmp[i],5-j];
end
else
begin
  result := result + tabelle_EAN_A[tmp[i]];
end;
  if i<5 then result:=result+'05';
end;
end;

function TQRcode.Code_Supp2:string;
var i,j : integer;
    tmp,mS : String;
begin
  FText := SetLen(2);
  i:=StrToInt(Ftext);
  case i mod 4 of
    3: mS:='EE';
    2: mS:='Eo';
    1: mS:='oE';
    0: mS:='oo';

```

```

end;
tmp:=getSupp(`(FText,1,5)+'0');
if FCheckSum then FText:=tmp else tmp := FText;
result := '506'; {Startcode}
for i:=1 to 2 do
begin
  if mS[i]='E' then
  begin
    for j:= 1 to 4 do result := result + tabelle_EAN_C[tmp[i],5-j];
    end
  else
  begin
    result := result + tabelle_EAN_A[tmp[i]];
    end;
  if i<2 then result:=result+'05';
  end;
end;
{-----}

procedure TQRcode.MakeModules;
begin
  case Typ of
    bcCode_2_5_interleaved,
    bcCode_2_5_industrial,
    bcCode39,
    bcCodeEAN8,
    bcCodeEAN13,
    bcCode39Extended,
    bcCodeCodabar,
    bcCodeUPC_A,
    bcCodeUPC_E0,
    bcCodeUPC_E1,
    bcCodeUPC_Supp2,
    bcCodeUPC_Supp5:
  begin
    if Ratio < 2.0 then Ratio := 2.0;
    if Ratio > 3.0 then Ratio := 3.0;
  end;
  bcCode_2_5_matrix:
  begin
    if Ratio < 2.25 then Ratio := 2.25;
    if Ratio > 3.0 then Ratio := 3.0;
  end;
  bcCode128A,
  bcCode128B,
  bcCode128C,
  bcCode93,
  bcCode93Extended,
  bcCodeMSI,
  bcCodePostNet:    ;
  end;
  modules[0] := FModul;
  modules[1] := Round(FModul*FRatio);
  modules[2] := modules[1] * 3 div 2;
  modules[3] := modules[1] * 2;
end;

procedure TQRcode.DoLines(data:string; Canvas:TCanvas);
var i:integer;
    lt : TBarLineType;
    xadd:integer;
    width, height:integer;

```

```

    a,b,c,d,
// Краю лінії нам потрібно 4 точки, тому що лінія прямокутник
    origin : TPoint;
    alpha:double;
begin
    xadd := 0;
    origin.x := FLeft;
    origin.y := FTop;
    alpha := FAngle/180.0*pi;
    { весь QRcode потрапляє в видиму область}
    origin := TranslateQuad2D(alpha,origin,Point(Self.Width,Self.Height));
    with Canvas do begin
        Pen.Width := 1;
        for i:=1 to Length(data) do
            begin
                {
                вхід: шаблон коду
                вихід: ширина і тип лінії
                }
                OneBarProps(data[i], width, lt);
                if (lt = black) or (lt = black_half) then
                    begin
                        Pen.Color := FColorBar;
                    end
                else
                    begin
                        Pen.Color := FColor;
                    end;
                Brush.Color := Pen.Color;
                if lt = black_half then
                    height := FHeight * 2 div 5
                else
                    height := FHeight;
                a.x := xadd;
                a.y := 0;
                b.x := xadd;
                b.y := height;
                c.x := xadd+width-1;
                c.y := Height;
                d.x := xadd+width-1;
                d.y := 0;
                {A, B, C, D будується прямокутник, який ми хочемо повернути}
                {Обертати прямокутник}
                a := Translate2D(Rotate2D(a, alpha), origin);
                b := Translate2D(Rotate2D(b, alpha), origin);
                c := Translate2D(Rotate2D(c, alpha), origin);
                d := Translate2D(Rotate2D(d, alpha), origin);
                {побудова прямокутника}
                Polygon([a,b,c,d]);
                xadd := xadd + width;
            end;
        end;
    end;

procedure TQRcode.DrawQRcode(Canvas:TCanvas);
var
    data : string;
    SaveFont: TFont;
    SavePen: TPen;
    SaveBrush: TBrush;
begin
    Savefont := TFont.Create;

```

```

SavePen := TPen.Create;
SaveBrush := TBrush.Create;
{отримання QRcode }
data := MakeData;
try
  Savefont.Assign(Canvas.Font);
  SavePen.Assign(Canvas.Pen);
  SaveBrush.Assign(Canvas.Brush);
  DoLines(data, Canvas); {від рисовка QRcode}
  if FShowText <> bcoNone then
    DrawText(Canvas); {показати текст }
  Canvas.Font.Assign(savefont);
  Canvas.Pen.Assign(SavePen);
  Canvas.Brush.Assign(SaveBrush);
finally
  Savefont.Free;
  SavePen.Free;
  SaveBrush.Free;
end;
end;

{
Основне застосування для цієї процедури тестування.
Примітка: Ця процедура змінює кисть поточного полотна.
}

procedure TQRcode.DrawText(Canvas:TCanvas);
var
  PosX, PosY: Integer;
  SaveFont: TFont;
begin
  with Canvas do
  begin
    Font.Size := 5;
    SaveFont := TFont.Create;
    try
      Font.Assign(ShowTextFont);
      try
        Pen.Color := Font.Color;
        Brush.Color := clWhite;
        PosX := FLeft;
        PosY := FTop;
        if ShowTextPosition in [stpTopLeft, stpBottomLeft] then
          PosX := FLeft
        else
          if ShowTextPosition in [stpTopRight, stpBottomRight] then
            PosX := FLeft + Width - TextWidth(Text)
          else
            if ShowTextPosition in [stpTopCenter, stpBottomCenter] then
              PosX := FLeft + Trunc((Width - TextWidth(Text))/2);
            if ShowTextPosition in [stpTopLeft, stpTopCenter, stpTopRight] then
              PosY := FTop
            else
            if ShowTextPosition in [stpBottomLeft, stpBottomCenter, stpBottomRight] then
              PosY := FTop + Height - TextHeight(Text);
            if FShowText in [bcoCode, bcoBoth] then
              TextOut(FLeft, FTop, FText); { контекст QRcode}
              TextOut(PosX, PosY, FText); { контекст QRcode }
            if FShowText in [bcoTyp, bcoBoth] then
              TextOut(FLeft, FTop+Round(Font.Height*2.5), GetTypText);
            {тип та ім'я QRcode}
            finally
              Font.Assign(SaveFont);

```

```
        end;
    finally
        SaveFont.Free;
    end;
end;
end;

procedure TQRcode.DoChange;
begin
    if Assigned(FOnChange) then
        FOnChange(Self);
end;

procedure TQRcode.SetRatio(const Value: Double);
begin
    if Value <> FRatio then
    begin
        FRatio := Value;
        DoChange;
    end;
end;

procedure TQRcode.SetType(const Value: TQRcodeType);
begin
    if Value <> FTyp then
    begin
        FTyp := Value;
        DoChange;
    end;
end;

procedure TQRcode.SetAngle(const Value: Double);
begin
    if Value <> FAngle then
    begin
        FAngle := Value;
        DoChange;
    end;
end;

procedure TQRcode.SetText(const Value: string);
begin
    if Value <> FText then
    begin
        FText := Value;
        DoChange;
    end;
end;

procedure TQRcode.SetShowText(const Value: TQRcodeOption);
begin
    if Value <> FShowText then
    begin
        FShowText := Value;
        DoChange;
    end;
end;

procedure TQRcode.SetTop(const Value: Integer);
begin
    if Value <> FTop then
    begin
        FTop := Value;
    end;
end;
```

```

        DoChange;
    end;
end;

procedure TQRcode.SetLeft(const Value: Integer);
begin
    if Value <> FLeft then
    begin
        FLeft := Value;
        DoChange;
    end;
end;

procedure TQRcode.SetChecksum(const Value: Boolean);
begin
    if Value <> FChecksum then
    begin
        FChecksum := Value;
        DoChange;
    end;
end;

procedure TQRcode.SetHeight(const Value: integer);
begin
    if Value <> FHeight then
    begin
        FHeight := Value;
        DoChange;
    end;
end;

function TQRcode.GetCanvasHeight: Integer;
var
    alpha :Extended;
begin
    alpha := FAngle/180.0*pi;
    Result := Round(abs(sin(alpha))*Self.Width + abs(cos(alpha))*Self.Height + 0.5);
    {округления }
end;

function TQRcode.GetCanvasWidth: Integer;
var
    alpha :Extended;
begin
    alpha := FAngle/180.0*pi;
    Result := Round(abs(cos(alpha))*Self.Width + abs(sin(alpha))*Self.Height +
    0.5);
end;
procedure TQRcode.SetShowTextFont(const Value: TFont);
begin
    FShowTextFont.Assign(Value);
    DoChange;
end;

procedure TQRcode.SetShowTextPosition(const Value: TShowTextPosition);
begin
    if Value <> FShowTextPosition then
    begin
        FShowTextPosition := Value;
        DoChange;
    end;
end;
end.

```