

Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
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 синтезу та
аналізу QR-кодів на мобільних пристроях під ОС Android”**

КБПЗ - 2023

Виконав здобувач вищої освіти
II курсу, групи КН-22МЗ
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Якубенко Я.І.
« ____ » _____ 2023 р.

Керівник проекту
доктор технічних наук, професор
_____ Смірнов О.А.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет

Центр *Заочної та дистанційної освіти*

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**

Якубенку Ярославу Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи

Дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android

2. Керівник роботи

Смірнов Олексій Анатолійович, докт. техн. наук, професор

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

затверджені наказом вищого навчального закладу № 37-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту 10.12.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android*

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	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

Якубенко Я.І. Дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android. 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 синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Метою розробки є дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Об'єктом дослідження є процес синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Предметом дослідження є методи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

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

Результат роботи – програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

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 мобільних пристроях під ОС Android.

Програму розроблено в середовищі Delphi 10.4.

Ключові слова: комп'ютерні науки, QR-код

ABSTRACT

Yakubenko Ya.I. Research and software implementation of the system of synthesis and analysis of QR codes on mobile devices with Android OS. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software was developed, which is intended for the system of synthesis and analysis of QR codes on mobile devices with the Android OS.

The purpose of the development is the research and software implementation of the system of synthesis and analysis of QR codes on mobile devices with the Android OS.

The object of the study is the process of synthesis and analysis of QR codes on mobile devices with Android OS.

The subject of the research is methods of synthesis and analysis of QR codes on mobile devices with Android OS.

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

The result of the work is the software implementation of the system of synthesis and analysis of QR codes on mobile devices with the Android OS.

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 mobile devices with Android OS.

The program was developed in the Delphi 10.4 environment.

Keywords: computer science, QR code

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.....	6
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.....	13
2.3 Розгорнута постановка завдання	19
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	21
3.1 Опис функціонування системи	21
3.2 Розробка структурної схеми.....	30
3.3 Розробка функціональної схеми	36
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.....	47
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	50
6 НАУКОВА НОВИЗНА	52

						ВКРМ-122.23.0066.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Якубенко Я.І.				Дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android	Літ.	Аркуш	Аркушів
Перев.	Смірнов О.А.					М	1	94
Н.контр.	Коваленко А.С.				ЦНТУ КН-22МЗ			
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	53
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	53
7.2 Розрахунок трудомісткості розробки програмної продукції.....	55
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	57
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	62
7.5 Визначення собівартості розробки та ціни програмної продукції.....	66
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	70
7.7 Визначення експлуатаційних витрат.....	70
7.8 Визначення економічної ефективності програмної продукції.....	72
7.9 Висновок.....	74
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	75
8.1 Вступ.....	75
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	76
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	77
8.4 Техніка безпеки та протипожежна профілактика.....	80
8.5 Розрахункова частина	82
9 ОСНОВНІ ВИСНОВКИ.....	86
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	88

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-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. QR-код – квадратна картинка в яку закодована якась інформація.

Є багато речей, які можна сказати про QR-коди, а саме:

- кілька режимів кодування;
- режим кодування ECI;
- кодування символів кандзі;
- оптимізація даних;
- мікро QR-коди;
- читання QR-кодів у браузері.

Все це розглянемо далі у даній роботі.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

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

- Огляд існуючих систем синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.
- Дослідження системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.
- Програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Об'єктом дослідження є процес синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Предметом дослідження є методи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

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

– Удосконалено метод синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

– Розроблено вітчизняний продукт синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

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 синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

QR-код – це винайдений десять років тому в Японії штрих-код у квадраті. Зведення в ступінь дало йому дві головних переваги: більша місткість інформації й швидке зчитування будь-якою камерою, звідки й розшифровка абрєвіатури – quick-response код.

Першими технологію стали використовувати рекламщики для залучення уваги до комерційних компаній. У просвітительських цілях QR-коди почали застосовувати порівняно недавно, зате з розмахом: можна згадати хоча б інфоцентр, що представляє собою засіяний QR-кодами купол. Можна сказати, що, так чи інакше, коди спонукають до більше глибокого вивчення теми, відправляючи вас у цифровий простір.

Звичайно, якщо судити здраво, то QR-коди – усього лише міні-віха в технологіях, і через якийсь час вони будуть не менш смішні, чим дискети. Але поки цей невеликий візерунок у куті роздруківки або сайту може заощадити час і папір, позбавити від передруку довгих посилань або хоча б різноманітиту рутину.

1.2 Область застосування

Областю застосування є можливість кодувати відео з YouTube, локацію на Google картах, файл із Dropbox, профіль у соцмережах, телефонний номер або e-mail. Словом усе, що можна відправити поштою або через соціальні мережі, QR-коди дозволяють прикріпити до будь-якої поверхні, будь те стіна або аркуш паперу.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Екскурсії

По тім же принципі можна запросто організувати екскурсію по школі. У кожного кабінету або закутка потрібно розмістити роздрукований QR-код з посиланням на відео, аудіо або цілий фотоальбом. Багато музеїв уже давно використовують цей прийом, тим самим розширюючи експозицію не тільки поясненнями, але й додатковими матеріалами начебто міні-фільму або навіть гри за мотивами виставки. Все це не говорячи вже про те, що QR-код – прекрасний інструмент навігації, що багато університетів використовують, наприклад, для навігації в бібліотеках.

QR-код – це можливість прибити онлайн-версію словника або підручника прямо на бібліотечну полицю

Інтерактивна розповідь

QR-коди вміють зберігати в собі тексти невеликого обсягу, які можна вважати без підключення до Інтернету. Для творчих занять це надає чималі можливості. Приклад гри: потрібно придумати оповідання, що складається з декількох частин – наприкінці кожної, як у комп'ютерній грі, буде кілька варіантів продовження. Можна створити інтерактивну версію казки про червону шапочку: з першого QR-коду зчитується зав'язка, а продовження можна вибрати самостійно («Скануйте код під номером 2, щоб героїня пішла прямо через ліс, або під номером 3, щоб вона пішла в обхід по безпечній дорозі»). Єдина проблема полягає в тому, що при кодуванні тексту значки виходять більші й складносурядні, а це може зробити гру громіздкою у виконанні.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0066.00.00.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

Slackmath

Slackmath – значна база роздруківок по геометрії й алгебрі, у які вже включені QR-коди з відеоінструкцією за рішенням кожного завдання. Якщо стандартні матеріали вчителя не влаштують, він може закодувати власні підказки, а потім роздрукувати що вийшов QR-код разом із завданням. Кожний, кому знадобиться консультація по темі, одержить її в будь-якому зручному місці, просто зчитавши код.

QR Treasure Hunt Generator

QR Treasure Hunt Generator автоматично створює QR-вікторину із запропонованих питань. Потім роздруковані QR-коди можна розосередити по класі або по всьому шкільному дворі. Учні, можливо, підтягнуть не тільки знання, але й свою фізпідготовку.

Квест із QR-кодами здатний об'єднати використання технологій у класі з фізичною активністю

QrStuff

Першою справою необхідно закодувати своє послання. Для створення власного QR-коду знадобляться тільки Інтернет, принтер і камера. Перший сайт, що приходить на розум – це елементарний qrcoder.ru. Якщо потрібно щось більше, то безкоштовний сервіс QrStuff за пару кліків допоможе створити коди різних квітів (у чорно-білих квадратах легко заплутатися).

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Kaiwa

Програма Kaiwa перетворює в QR-коди не тільки гіперпосилання, але й самий звичайний текст, у тому числі, набраний кирилицею. У такому випадку не потрібний навіть Інтернет, щоб уважати закодоване послання, що відкриває ціле поле для експериментів.

Reader Kaiwa

Майже універсальним додатком можна назвати Reader Kaiwa, що працює на різних моделях смартфонів – на сайті сервісу є тест на сумісність і російськомовну інструкцію.

QR Droid

Більше серйозна програма QR Droid дозволяє зчитувати коди, створювати власні й відправляти друзям. Все зчитане зберігається в історії – при бажанні додаток може навіть запам'ятати місце розташування того або іншого коду. Neoreader, сервіс зі схожими функціями, працює на iOS і управляється з усіма типами кодів, включаючи ті, які ви навряд чи зустрінете.

QRreader

Тим, у кого немає смартфона, допоможе проста програма QRreader: у ній лише одна функція – досить піднести код до веб-камери, і додаток його зчитує. А якщо у вас немає й веб-камери, урятує розширення для Google Chrome, що розкусить будь-який QR-код, що зустрівся в Інтернеті.

QR Code Reader

У цього додатка дуже багато позитивних оцінок. При першому включенні вам пропонують дозволити програмі запам'ятовувати геодані місця, у якому був отриманий QR-код, щоб легше було згадати, де був він був прочитаний.

Головний екран зустрічає нас простим інтерфейсом камери й червоним квадратиком по центрі. При наведенні камери на QR-код він автоматично розпізнається (навіть не потрібно жати на кнопку спуска) і зверху відображається отримана інформація (наприклад, посилання). При її натисканні вас переносить у браузер.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

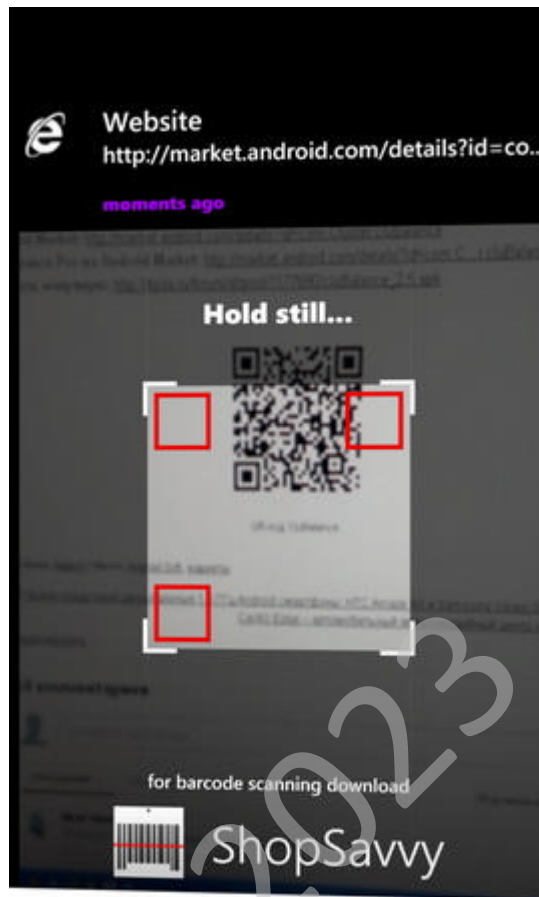


Рисунок 2.1 – Інтерфейс користувача QR Code Reader

У додатку є список, у якому фіксуються всі прочитані коди, разом з геоданими, якщо ви включили відповідну опцію.

Проста, але функціональна програма з усіма потрібними налаштуваннями.

Scan

А цей додаток саме популярне із всіх програм такого роду.

При першому запуску у вас також просять доступ до ваших геоданих, після чого відкривається панель із налаштуваннями. Особливо цікава «фішка» з реєстрацією аккаунту. Для його створення досить ввести адресу електронної пошти й пароль, підтвердження не потрібно. Після цього програма створює вам аккаунт, що буде синхронізувати інформацію про прочитані коди.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Вкладка Scan відповідає за читання кодів. Для їхньої розшифровки досить просто навести камеру на потрібний код. Якщо код містить інтернет-посилання, то вона відразу відкривається в браузері.

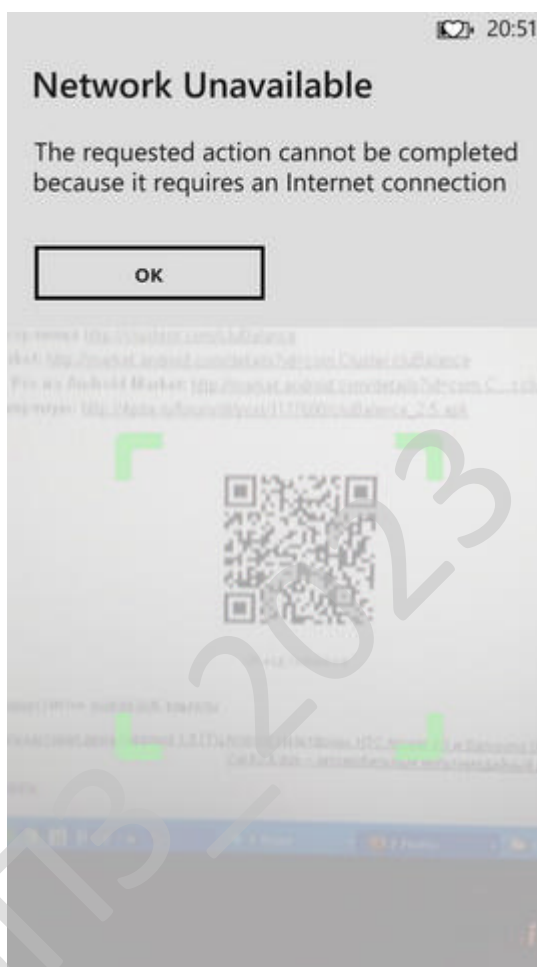


Рисунок 2.2 – Інтерфейс користувача Scan

Список прочитаних кодів перебуває в окремій вкладці. Крім збереженого посилання, також вказується й час зняття коду.

Програма безкоштовна й без реклами.

QR Code Offline

Це додаток не зустрічає вас вітанням, а відразу ж відкриває вікно для сканування коду, немов показуючи, що вона відразу готова до роботи.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Після розшифровки вам пропонується ознайомитися із захованої в кодї інформацією й відразу ж перейти по посиланню коду.



Рисунок 2.3 – Інтерфейс користувача QR Code Offline

У додатка також є історія кодів, крім того, з його допомогою ви самі можете створити QR-код. Особливо радує кількість різних варіантів інформації, яку можна з його допомогою зашифрувати.

JustScan

Ще один додаток для читання кодів. Зовсім простенький, тут є тільки екран для сканування й швидкого переходу по посиланню. Крім того, програма погано працює з маленькими кодами, а точніше, зовсім не сканує їх. Тому рекомендувати його для постійного використання я не можу.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

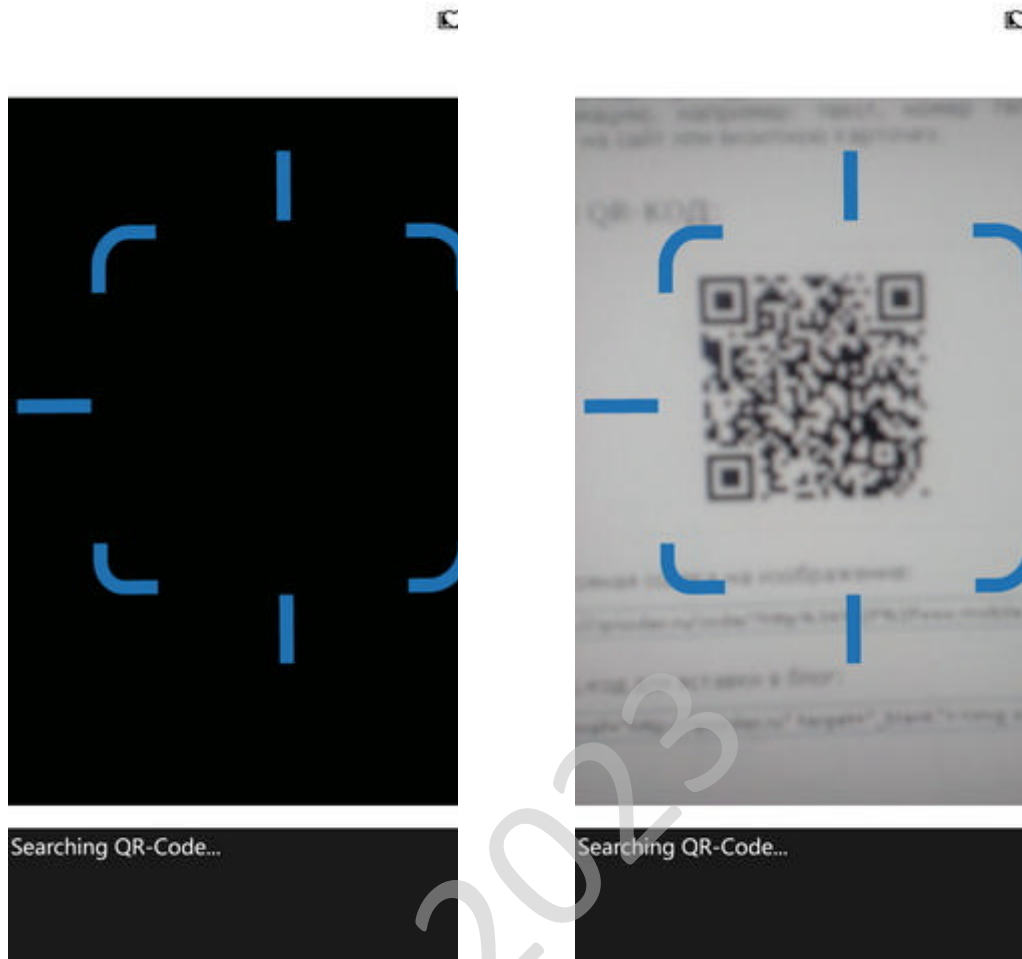


Рисунок 2.4 – Інтерфейс користувача JustScan

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.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

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.

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

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

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

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

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

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.

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 Snake.
- 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.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Змінені стилі VCL для High DPI

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

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

Нові High DPI стилі й стилізація окремих VCL компонент

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

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

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.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

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.

КБПЗ - 2023

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

QR-коди містять дані, це чесно сказати. Тип даних можна вибрати, але, звичайно, це визначає максимальний обсяг інформації, яку можна зберегти :

- номери (до 7089);
- алфавітно-цифровий (цифри, великі літери, купа символів: ~65% дорожче цифр);
- байти (лише 8 біт символів у кодуванні Latin-1, ~140% дорожче);
- кандзі (~290% дорожче).

Не дивно, що кандзі є одним із основних наборів символів, оскільки QR-коди були розроблені Denso Wave, японською компанією з автоматизації.

Насправді в останніх версіях є й інші режими кодування, але, як згадувалося раніше, поки що ми зосередимося на 8-бітних байтах. І, зрештою, QR-код — це набір бітів, тож якщо ви хочете закодувати свою інформацію як забажаєте, ви можете.

Крім того, QR-коди також можуть *перемикатися* в інший режим кодування в середині своїх даних, але ми поки не будемо розглядати цей випадок.

Розміри

QR-коди завжди квадратні, але їх розміри різняться. Розмір визначається незвичайним терміном «версія», так що версія 1 має розмір 21×21 пікселів, тоді як версія 40 (найбільша) має розмір 177×177 пікселів. QR-код на 1 версію більший на 4 пікселі ширший і вищий, тому розмір становить $(17 + \text{версія} * 4)$ пікселів.

Крім того, ми повинні називати їх не пікселями, а скоріше «модулями» (знову незвично, але, можливо, щось було втрачено в перекладі з японської).

Оскільки більші QR-коди складніше декодувати (і обчислювально

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

верхньому правому та нижньому лівому кутах, розділених лінією порожніх модулів;

– **шаблони вирівнювання** : квадрати розміром 5×5 , розташовані на кутах і перетинах сітки $n \times n$ (якщо вони не зайняті шаблонами шукача); n коливається між 2 і 6, тому їх $n^2 - 3$, за винятком версії 1, яка не має шаблону вирівнювання;

– **часові візерунки** : горизонтальна і вертикальна лінії чергування темних і світлих модулів, що з'єднують шаблони шукача (ви помітили це, лише якщо глибоко вивчили деякі QR-коди);

– **темний модуль** : просто модуль, який завжди темний, розміщений у 9-му стовпці та $(4 * \text{версія}) + 10$ -му рядку (б'юся об заклад, ви ніколи цього не помічали!).

Крім того, у великих QR-кодах (від версії 7 і вище) кілька областей зарезервовано для даних формату.

Ємність

Залежно від версії, режиму кодування та рівня виправлення помилок визначається *ємність QR-коду*. Доступний простір, який не зайнятий фіксованими шаблонами чи зарезервованими областями, поділено на групи з 8 модулів, які називаються «кодовими словами»: уявіть їх як класичні 8-бітні байти.

Тому загальна кількість доступних кодових слів є фіксованою для кожної версії: 26 для версії 1, 44 для версії 2 і так далі, до 3706 для версії 40.

Для кожної версії також визначаються кодові слова, зарезервовані для виправлення помилок.

Без зайвих слів, давайте почнемо створювати невеликий QR-код із кодуванням ISO-8859-1 байт!

ISO-8859-1

Так, QR-коди використовують ISO-8859-1 (також відомий як Latin-1) для кодування рядків байтів. Сьогодні UTF-8 більш поширений, але деякий час тому

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Також ця інша таблиця говорить нам, що версія 2 може містити 28 кодових слів даних для середньої корекції: ці 2 запасних кодових слова будуть використовуватися для інформації про дані.

Якщо ми хотіли отримати вищий рівень виправлення помилок, ми повинні були вибрати більшу версію.

Біти даних

Перші 4 біти нашої послідовності даних – це 0100, наш режим кодування.

Тоді ми скажемо, якої довжини буде наш рядок. Для цього знову потрібна таблиця, оскільки кількість бітів, зарезервованих для цього значення, є змінною.

Таблиця 3.3 – Біти даних

Режим кодування	Версія 1-9	Версія 10-26	Версія 27-40
Числовий	10	12	14
буквено-цифровий	9	11	13
Байт	8	16	16
кандзі	8	10	12

Перетворення цього на зручну функцію:

```
const LENGTH_BITS = [
  [10, 12, 14],
  [9, 11, 13],
  [8, 16, 16],
  [8, 10, 12]
];

function getLengthBits(mode, version) {
  // ECI mode folds into byte mode
  // Basically it's `Math.floor(Math.log2(mode))` but much
  faster
  // See https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Math/clz32
  const modeIndex = 31 - Math.clz32(mode);
  const bitsIndex = version > 26 ? 2 : version > 9 ? 1 : 0;
  return LENGTH_BITS[modeIndex][bitsIndex];
}
```

Оскільки `getLengthBits(4, 2) === 8` нам знадобиться 8 бітів, а 23 (довжина нашого рядка) у двійковій системі дорівнює 10111, наші перші біти:

```
01000001 0111....
```

Далі *реальні* дані. Все, що нам потрібно зробити, це отримати код символів рядка в ISO-8859-1:

```
h t t p s : / / w w w . q r c o d
e . c o m /
104 116 116 112 115 58 47 47 119 119 119 46 113 114 99 111 100
101 46 99 111 109 47
```

Тепер перетворіть усе в двійковий і об'єднайте в попередню послідовність:

```
01000001 01110110 10000111 01000111 01000111
00000111 00110011 10100010 11110010 11110111
01110111 01110111 01110010 11100111 00010111
00100110 00110110 11110110 01000110 01010010
11100110 00110110 11110110 11010010 1111....
```

Тепер ми повинні поставити блок завершення, який рівно 4 нулі, тому останні кодові слова будуть 11110000. Ми все ще маємо заповнити 3 із 28 доступних кодових слів.

Залишилося місце

Ми заповнили всі 8 бітів останнього кодового слова, інакше нам довелося б заповнити решту бітів нулями (це завжди так у байтовому режимі).

За допомогою решти кодових слів ми можемо зробити дві речі:

– замість 4-бітового блоку завершення ми можемо поставити інший блок режиму кодування (можливо, інший) і почати іншу послідовність – але лише з трьома кодовими словами ми не можемо зробити багато;

– заповніть решту кодових слів послідовностями 11101100 00010001 (що перетворюється на 236 і 17 у десятковій системі), доки не буде досягнуто обмеження.

Чому 236 і 17? Я поняття не маю, але я припускаю, що вони (Denso Wave?) зробили багато спроб і перевірили, що ці дві послідовності створюють

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27


```

    data[1] = (content.length >> rightShift) & 255;
  }
  data[dataIndexStart] = (content.length & andMask) <<
leftShift;

  for (let index = 0; index < content.length; index++) {
    const byte = content.charCodeAt(index);
    data[index + dataIndexStart] |= byte >> rightShift;
    data[index + dataIndexStart + 1] = (byte & andMask) <<
leftShift;
  }
  const remaining = dataCodewords - content.length -
dataIndexStart - 1;
  for (let index = 0; index < remaining; index++) {
    const byte = index & 1 ? 17 : 236;
    data[index + content.length + 2] = byte;
  }
  return data;
}

```

Наприкінці цього першого кроку ми повинні отримати такий результат:

```

getBytesData('https://www.qrcode.com/', 8, 28)
// Uint8Array(28) [65, 166, 135, 71, 71, 7, 51, 162, 242, 247,
119, 119, 114, 231, 23, 38, 54, 246, 70, 82, 230, 54, 246, 210, 240,
236, 17, 236]

```

Пам'ятайте: вищезгадана функція працюватиме *лише* для байтового режиму!

Інші режими кодування

Наразі ми не будемо вдаватися в подробиці, але для числового режиму нам потрібно розділити число на групи з 3 цифр і закодувати кожен групу 10 бітами ($2^{10} = 1024$, ^{тому} втрачений простір мінімальний).

Натомість буквено-цифровий режим містить 45 символів, тому рядок має бути розбитий на групи по 2 символи. Кожен символ має значення (спочатку цифри, потім великі латинські літери, потім пробіл і символи \$, %, *, +, \, -, ., /, :),

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

тому кожна пару символів можна перевести в число в діапазоні від 0 до 2024 ($= 45^2 - 1$). Тому нам потрібно 11 біт на кожні два буквено-цифрових символи ($2^{11} = 2048$).

Для режиму кандзі... о боже. По-перше, нам потрібно отримати код Shift JIS піктограми, і найкращий спосіб зробити це — використовувати бібліотеку на кшталт iconv-lite або, якщо ви хочете зробити це самостійно, використати її таблицю символів. Крім того, можна використовувати не всі символи, а лише ті, що знаходяться в діапазонах від 0x8140 до 0x9FFC і від 0xE040 до 0xEBBF. Зрештою, символ кандзі займе 13 біт.

3.2 Розробка структурної схеми

Скажу чітко: насправді показ QR-коду – це лише проблема візуалізації. Ми можемо зробити це за допомогою SVG, а `<canvas>`, групи квадратів `s` або навіть цих двох емодзі: `□■`. Це не дуже важливо й не складно для тих, хто має мінімальний досвід у відтворенні матеріалів у мережі.

Важливо отримати матрицю бітів, яка дозволить нам створити таку фігуру.

Почнемо з фактичного зберігання даних. Знову ж таки, для зручності ми можемо використовувати лише масив масивів, тобто матрицю, щоб записати, чи є модуль світлим (`0`) чи темним (`1`). Але для рядків ми `Uint8Array` знову можемо використовувати `s`, оскільки вони швидші за звичайні масиви, а також для `.set()` методу, який стане в нагоді. Почнемо з простого:

```
function getSize(version) {
  return version * 4 + 17;
}

function getNewMatrix(version) {
  const length = getSize(version);
  return Array.from({ length }, () => new Uint8Array(length));
}
```

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30


```

fillArea(qrCode, 0, 7, 1, 7, 0);
fillArea(qrCode, size - 8, 0, 8, 1, 0);
fillArea(qrCode, 0, size - 8, 1, 7, 0);
fillArea(qrCode, 7, size - 8, 8, 1, 0);
fillArea(qrCode, size - 7, 7, 1, 7, 0);
// Alignment pattern
fillArea(qrCode, size - 9, size - 9, 5, 5);
fillArea(qrCode, size - 8, size - 8, 3, 3, 0);
qrCode[size - 7][size - 7] = 1;
// Timing patterns
for (let pos = 8; pos < VERSION * 4 + 8; pos += 2) {
    qrCode[6][pos] = 1;
    qrCode[6][pos + 1] = 0;
    qrCode[pos][6] = 1;
    qrCode[pos + 1][6] = 0;
}
qrCode[6][size - 7] = 1;
qrCode[size - 7][6] = 1;
// Dark module
qrCode[size - 8][8] = 1;

// Placing message and error data
let index = 0;
for (const codeword of codewords) {
    // Counting down from the leftmost bit
    for (let shift = 7; shift >= 0; shift--;) {
        const bit = (codeword >> shift) & 1;
        const [row, column] = moduleSequence[index];
        index++;
        qrCode[row][column] = bit;
    }
}
return qrCode;
}

```

Ми отримаємо прото-QR-код версії 2. Під "прото" я маю на увазі, що його

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

не було перетворено останньою дією: маскуваням. Він полягає в виконанні XOR всіх модулів за допомогою одного з 8 попередньо визначених шаблонів. А *навіщо* нам це робити, можете запитати ви?

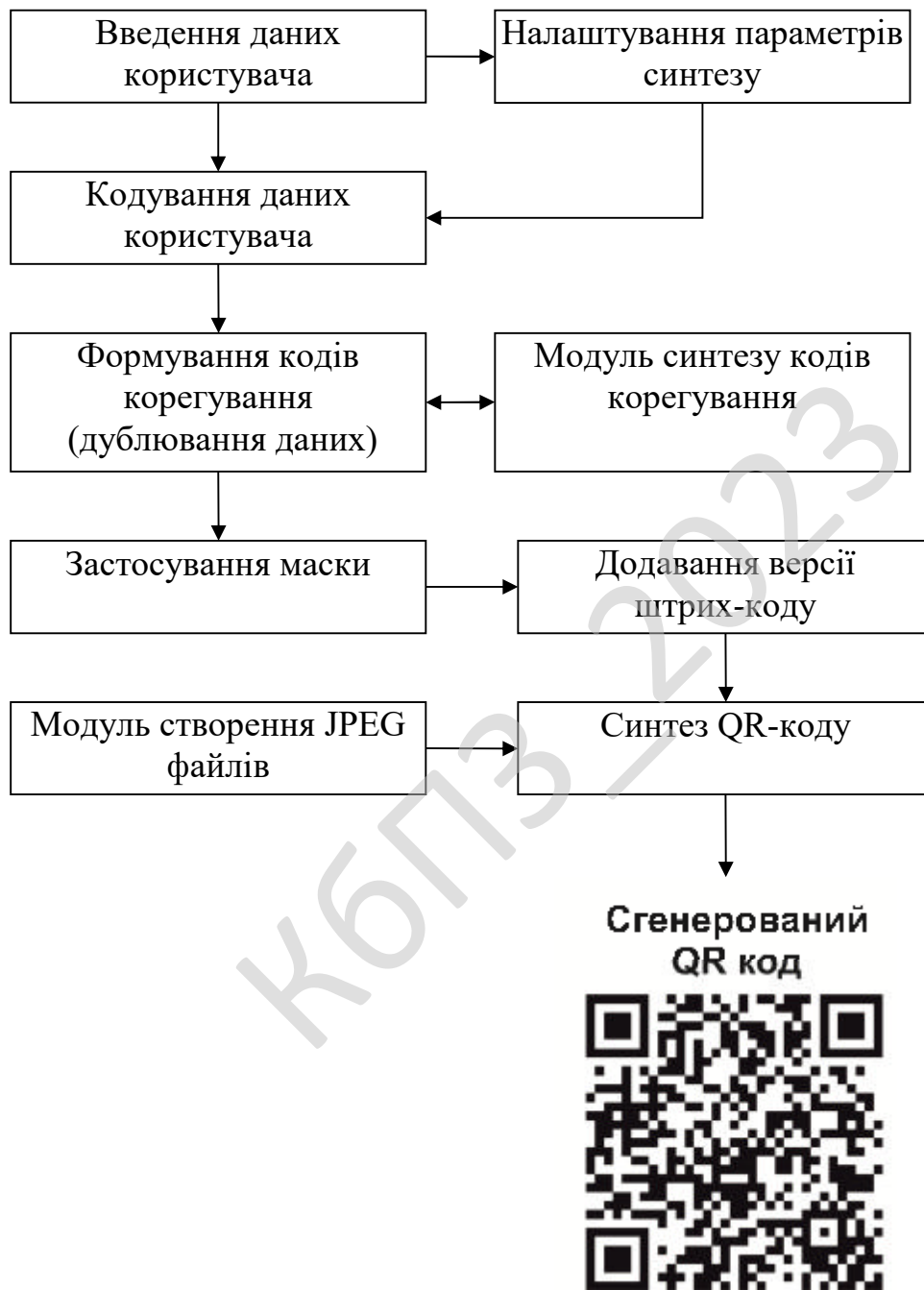


Рисунок 3.1 – Структурна схема системи

Що ж, цього разу це *має* сенс. Якщо ви подивитеся на наш прото-QR-код, то там є великі області, рівномірно заповнені темними або світлими візерунками, і сканери зазвичай не люблять їх, оскільки вони можуть не відповідати сітці або неправильно підраховувати рядки чи стовпці. Тому нам доведеться застосувати маску, щоб мінімізувати цю проблему.

На рисунку 3.1 зображено розроблену структурну схему забезпечення системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

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.

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.

3.3 Розробка функціональної схеми

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

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

створені області детектування та центрування. Після цього зчитують системну інформацію. Яка представляє собою графічно представлений бітовий масив. Та усикаючи все зайве за допомогою маски отримують кінцеві дані – текст «Кіровоградський національний технічний університет».

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

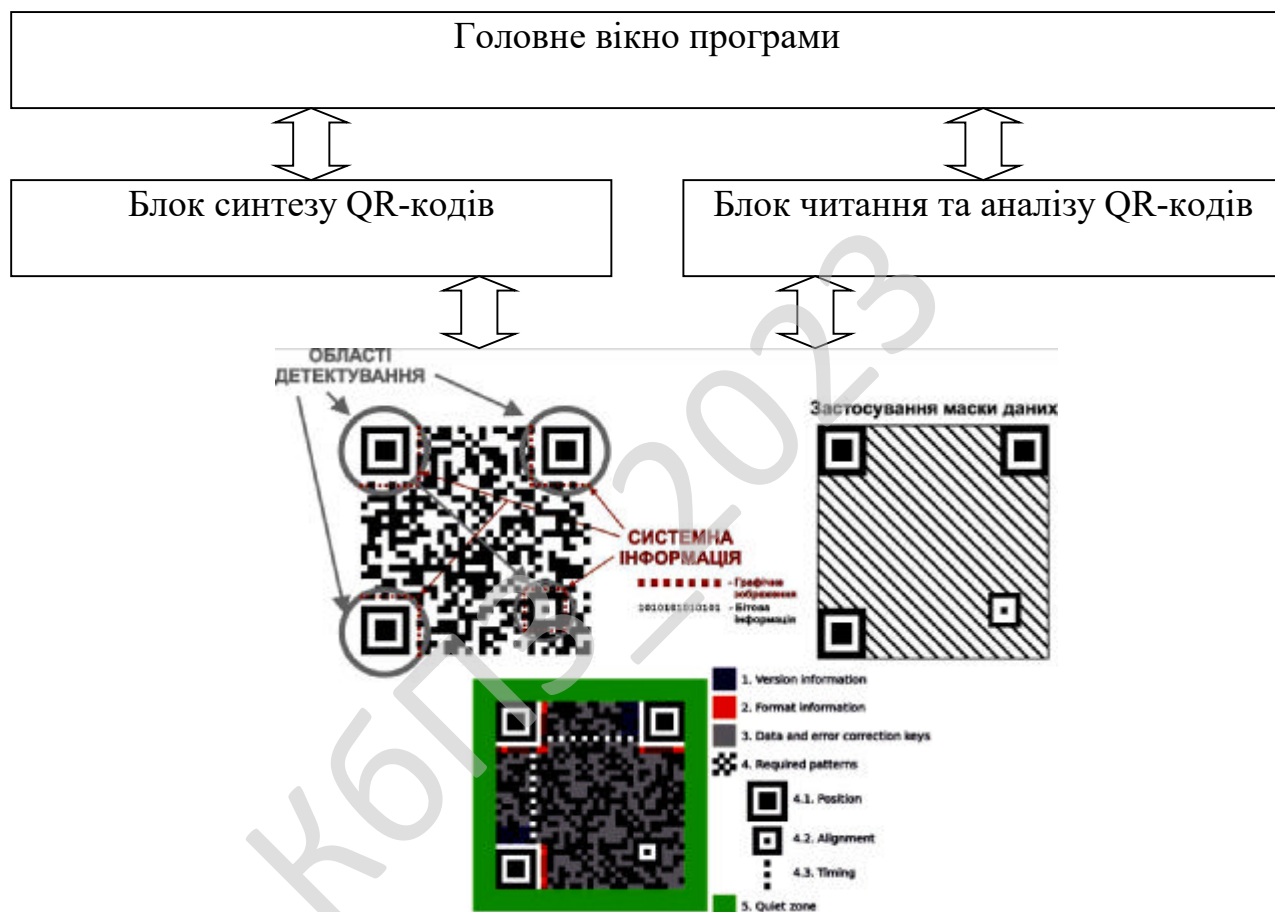


Рисунок 3.3 – Функціональна схема системи

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.4. Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

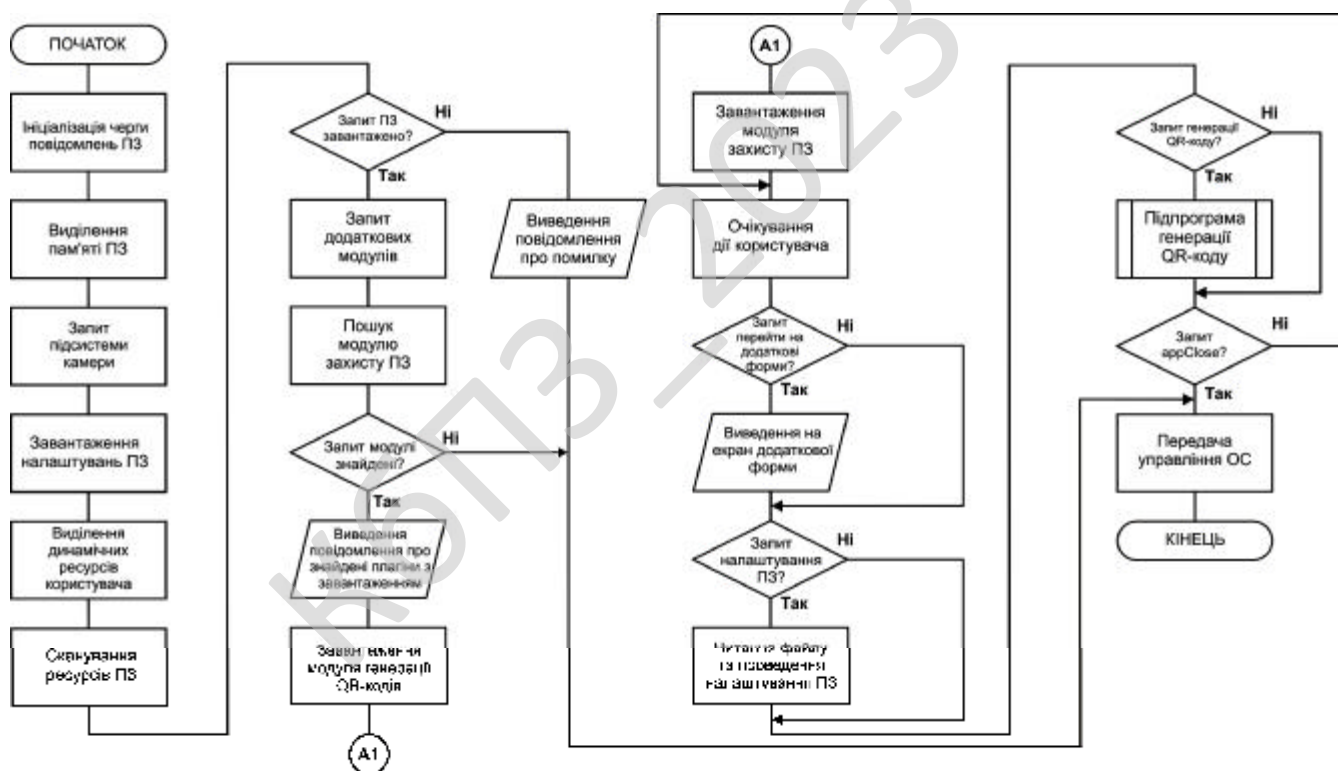


Рисунок 4.1 – Блок схема основної програми

З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик

підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

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

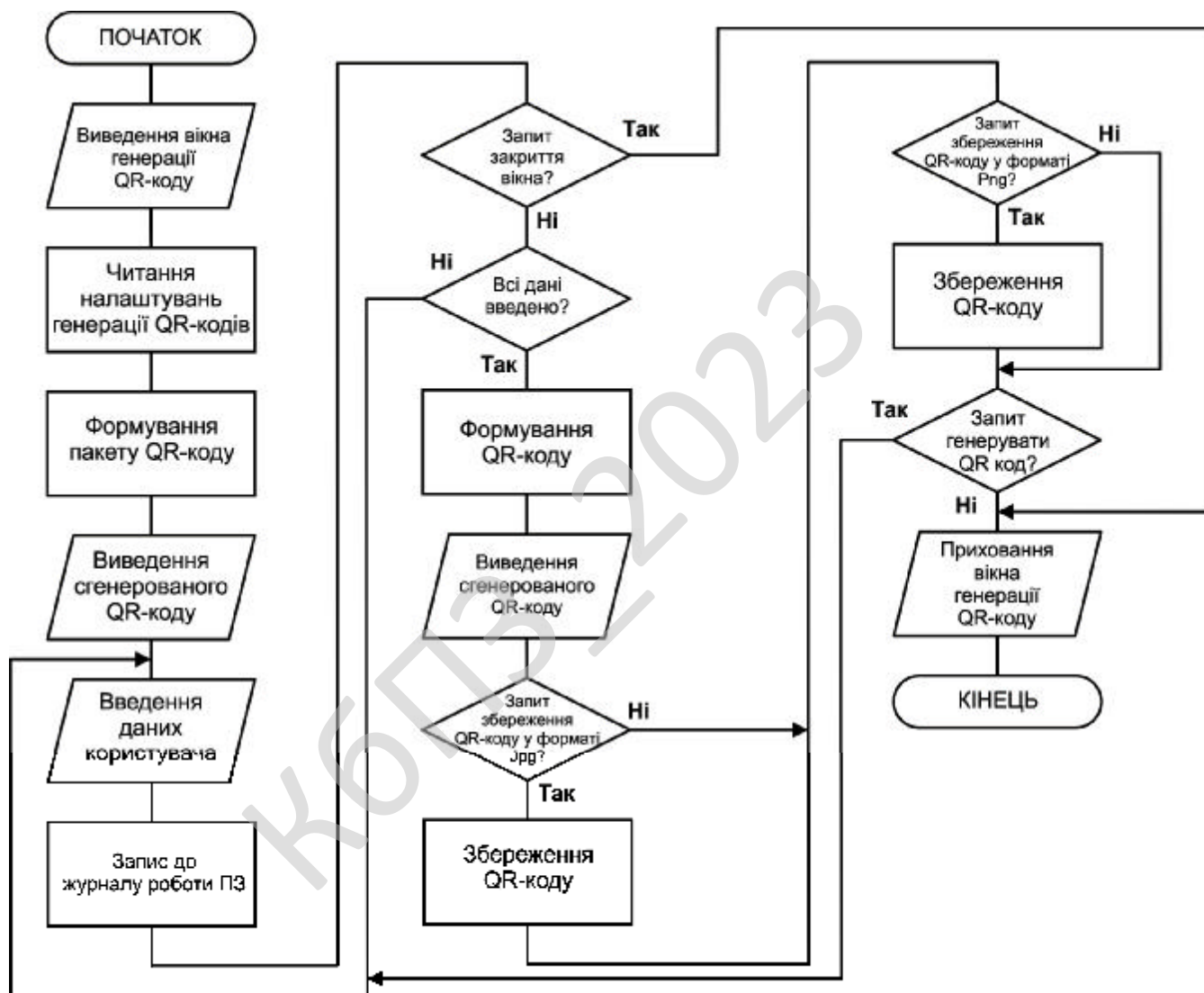


Рисунок 4.2 – Блок схема підпрограми

При розробці використовувались концепції діаграм діяльності. Тобто в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом

графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Це фундаментальна одиниця визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії.

Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Опис алгоритмів функціонування системи

Розглянемо використання компоненту TIdHTTP для спрощення завантаження даних.

```
// назва функції
function TQr2Stream(Correction_Level: TQrImage_ErrCorrLevel;
                   Width, Height: Integer;
                   const DataString: string): TStream;
// константи - шлях http
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
  // початок функції
  Result := TMemoryStream.Create; // виділення ділянки пам'яті
  // http запит
  IdHTTP1.Get(Format(UrlGoogleQrCode, [Width, Height,
                                     QrImgCorrStr[Correction_Level],
                                     HTTPEncode(DataString)]), Result);
  Result.Seek(0, 0);
end;
```

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

```

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

```

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;
Type
// стандартний тип класу від TForm
    TForm1 = class(TForm)
        BtnGetQR: TButton;
        Image1: TImage;
        Edit1: TEdit;
// опис процедури
    procedure BtnGetQRClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
// об'ява класів
    Form1: TForm1;

```

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

```

Implementation
// реалізація
// завантаження ресурсів
{$R *.dfm}
    Uses
// сторонні бібліотеки які використовуються
    GIFImage,
    HTTPApp,
    WinInet;
Type
// тип помилки
    TQrImage_ErrCorrLevel=(L,M,Q,H);
Const
// константа - база з QR кодами
UrlGoogleQrCode='http://chart.apis.google.com/chart?chs=%dx%d&cht=qr&chld=%s&chl=%s&chof=gif';
QrImgCorrStr:array [TQrImage_ErrCorrLevel] of string=('L','M','Q','H');
// отримання даних
procedure WinInet_HttpGet(const Url: string;Stream:TStream);
const
    BuffSize = 1024*1024;
var
    hInter : HINTERNET;
    UrlHandle: HINTERNET;
    BytesRead: DWORD;
    Buffer : Pointer;
begin
    hInter := InternetOpen('', INTERNET_OPEN_TYPE_PRECONFIG, nil, nil, 0);
    if Assigned(hInter) then
    begin
        Stream.Seek(0,0);
        GetMem(Buffer,BuffSize);
        try
            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;
            end;
        end;
    end;
end;

```

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

```

        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 TQ.BtnGetQRClick (Sender: TObject);
var
    ImageStream : TMemoryStream;
    GifImage : TGIFImage;
    ddd : real;
begin
    Image1.Picture:=nil;
    ImageStream:=TMemoryStream.Create;
    GifImage :=TGIFImage.Create;
    try
        try
            ddd:=strtofload (Edit1.text);
            GetQrCode (300, 300, L, floattostr (ddd), ImageStream);
            if ImageStream.Size>0 then
                begin ImageStream.Position:=0;
                    GifImage.LoadFromStream (ImageStream);
                    Image1.Picture.Assign (GifImage); end;
            except
                on E: exception do
                    ShowMessage (E.Message);
                end;
            finally
                ImageStream.Free;
            end;
        end;
    end;
end;

```

					БКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

GIFImage.Free;

end;

end;

end.

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;

– 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).

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 A / B / C, режим двійкового рахунку, режими GS1 та ін.).

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

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).

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.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм Madryga. Алгоритм Madryga складається із двох вкладених циклів. Зовнішній цикл повторюється вісім разів (для гарантії надійності число циклів можна збільшити) і полягає в застосуванні внутрішнього циклу до відкритого тексту. Внутрішній цикл перетворює відкритий текст у шифртекст і виконується однократно над кожним 8-бітовим блоком (байтом) відкритого тексту. Таким чином, весь відкритий текст послідовно вісім разів обробляється алгоритмом.

Ітерація внутрішнього циклу оперує з 3-байтовим вікном даних, названим робочим кадром (рисунок 4.3). Це вікно зрушується на 1 байт за ітерацію. (При роботі з останніми 2 байтами дані покладаються циклічно замкнутими). Перші два байти робочого кадру циклічно зрушуються на змінне число позицій, а для останнього байта виконується операція XOR з декількома бітами ключа. У міру переміщення робочого кадру всі байти послідовно циклічно зрушуються й піддаються операції XOR із частинами ключа. Послідовні циклічні зрушення перемішують результати попередніх операцій XOR і циклічного зрушення, причому на циклічне зрушення впливають результати XOR. Завдяки цьому процес у цілому оборотний.

Оскільки кожний байт даних впливає на два байти ліворуч і на один байт праворуч від себе, після восьми проходів кожний байт шифртексту залежить від 16 байтів ліворуч і 8 байтів праворуч.

При шифруванні кожна ітерація внутрішнього циклу встановлює робочий кадр на передостанній байт відкритого тексту й циклічно переміщає його до третього з кінця байту відкритого тексту. Спочатку весь ключ піддається операції XOR з випадковою константою й потім циклічно зрушується вправо на 3 біти (ключ і дані рухаються в різних напрямках, щоб мінімізувати надлишкові операції з бітами ключа). Молодші три біти молодшого байта робочого кадру

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

зберігаються, вони визначають циклічне зрушення інших двох байтів. Далі конкатенація двох старших байтом циклічно зрушується вліво на змінне число біт (від 0 до 7). Потім над молодшим байтом робочого кадру виконується операція XOR з молодшим байтом ключа. Нарешті робочий кадр зміщується вправо на один байт і весь процес повторюється.

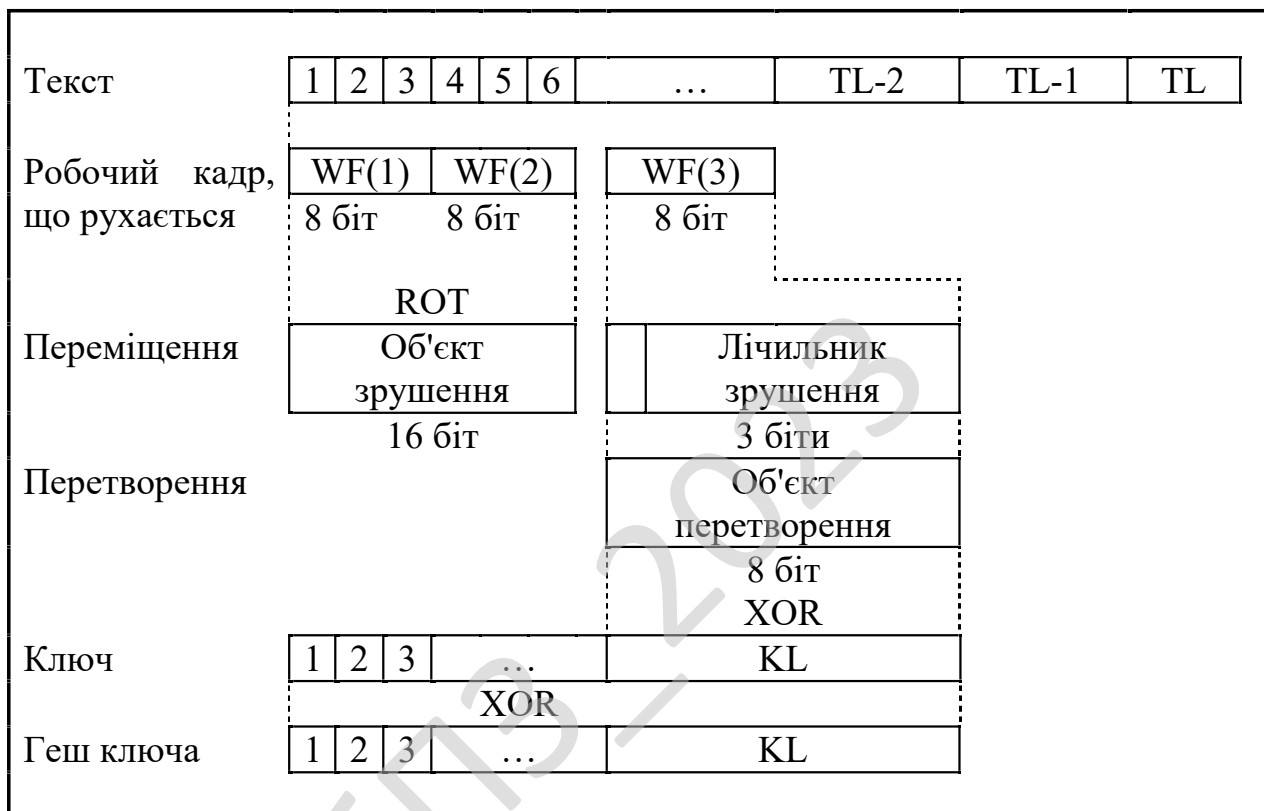


Рисунок 4.3 – Одна ітерація алгоритму Madryga

Випадкова константа призначена для перетворення ключа в псевдовипадкову послідовність. Довжина константи повинна бути рівній довжині ключа. При обміні даними абоненти повинні користуватися однією тією же константою. Для 64-бітового ключа Madryga рекомендує константу 0x0fle2d3c4b5a6678.

При розшифруванні процес повторюється у зворотному порядку. У кожній ітерації внутрішнього циклу робочий кадр встановлюється на байт, третій ліворуч від останнього байта шифртексту, і циклічно зрушується у зворотному напрямку до байта, розташованого на 2 байти уліво відносно останнього байта шифртексту. 2 байти шифртексту в процесі циклічно зрушуються вправо, а ключ – уліво. Після циклічних зрушень виконується операція XOR.

КБПЗ – 2023

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс простий та інтуїтивно зрозумілий. Головне вікно складається з поля сканування. При необхідності генерування QR-коду необхідно продовж 2 секунд тиснути на екран мобільного пристрою, тоді у вспливаючому вікні зазначити URL посилання та натиснути кнопку ок, що у подальшому проведе генерування QR-коду та виведення його на екран.



Рисунок 5.1 – Головне вікно генерації QR коду

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

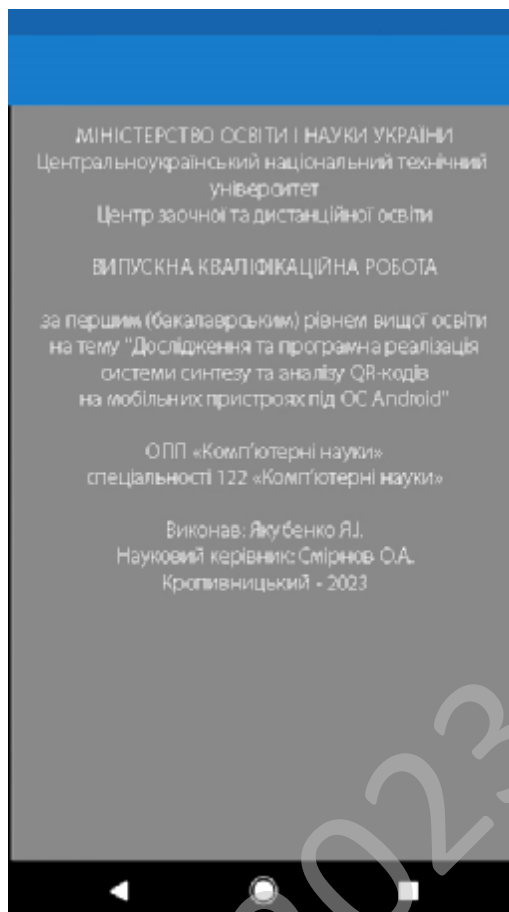


Рисунок 5.2 – Авторське право

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (флешку або ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості. Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareestruватися), заплативши авторіві певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

КБПЗ_2023

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Метою розробки є дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Об'єктом дослідження є процес синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Предметом дослідження є методи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

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

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

– Удосконалено метод синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

– Розроблено вітчизняний продукт синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

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

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android. 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	99
3. Запланований термін розробки, днів	Fp _q	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

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

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	100000
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	55
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-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

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

B – показник ступеня, що визначається співвідношенням:

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

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

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

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

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

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

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

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

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

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

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

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

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

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 35 = 59 \text{ люд/день.}$$

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

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	59	Ф 7.1-7.4
Впровадження	13	Д13
Всього	100	–

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.

$$Ч = \frac{100 \cdot 1}{60 - 5} = 1,8 \text{ ставки.}$$

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

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

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

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м.п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

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:

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

$$\Phi_{др}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки.}$$

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, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

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

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	12375	37125
Продакт-менеджер	0,25	13000	9750
Інженер-програміст	1,8	13000	70200
Інженер - електронщик	1,2	13000	46800
Інженер-системотехнік	0,25	12000	9000
Адміністратор мережі	0,5	12000	18000
Системний програміст	0,25	12000	9000
Дизайнер WEB	0,25	12000	9000
Інженер-верстальник	0,25	12000	9000
Бухгалтер-економіст	0,5	12000	18000
Всього за період розробки	$R_{cn} = 6,25$	-	$\Phi_{роб} = 235875$

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.

$$z_{cd} = \frac{235875}{6,25 \cdot 60} = 629 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

$$B_{y\delta} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

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

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

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 29000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\delta} = 8 \cdot 8 \cdot 29000 = 1858000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 185800 грн.

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Brain за 07.11.23 – джерело <http://brain.com.ua>.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" LG 22MP58VQ-P 5 мс IPS 1920x1080 250/1000M:1 178/178 D-Sub+HDMI+DVI	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

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 випробовування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	199177

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

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	1858000	-	-
2. Передавальні пристрої	185800	-	-
Всього по групі	2043800	5	102190
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
4. Нематеріальні активи	100000	10	10000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	143000	20	28600
7. Господарський інвентар	28000	25	7000
Всього по групі	180031	-	37857,75
Разом	$K_p = 2523008$		$A_p = 249636,25$

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону «Кіровоград-Авто», джерело <http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description>, складає 143000 грн.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

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.

$$Z_o = 629 \cdot 100 / 99 = 635 \text{ грн.}$$

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, %.

$$Z_d = 635 \cdot 10 \cdot 0,01 = 64 \text{ грн.}$$

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

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

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(635+64) = 154 \text{ грн.}$$

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

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

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 635 \cdot 15 \cdot 0,01 = 95 \text{ грн.}$$

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

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

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

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

Згідно виданих норм приймаємо пів пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 206$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

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

$$Z_{M1} = 206 \cdot 0,5 \cdot 3 = 309 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості, що дорівнює кількості 50 боксових версій програм та одного DVD диска для збереження резервної копії програми:

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

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 23,46 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 42 грн./шт.

$$Z_{M2} = 23,46 \cdot 50 + 42 = 1215 \text{ грн.}$$

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

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

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

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

$$Z_M = (309 + 1215 + 1702) / 99 = 33 \text{ грн.}$$

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, %.

$$O_n = 635 \cdot 15 \cdot 0,01 = 95 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 99$ прим.):

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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

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

$$A_m = 249636 \cdot 3 / (99 \cdot 12) = 631 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	635
2. Додаткова зарплата виконавців	Z_o	64
3. Відрахування на соціальні потреби	C_{oc}	154
4. Загальногосподарські витрати	G_{ocn}	95
5. Витрати на матеріали	Z_M	33
6. Освоєння нових операційних систем, мов програмування	O_n	95
7. Амортизація основних фондів	A_m	631
8. Повна собівартість програмного забезпечення	C_n	1707
9. Плановий прибуток	P_p	940
10. Ціна підприємства $C_n = C_n + P_p$	C_n	2647
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	529,4
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	3176,4

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

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

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

$$C_n = 635 + 64 + 154 + 95 + 33 + 95 + 631 = 1707 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 55 \cdot 1707 = 940 \text{ грн.}$$

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 7.10.

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	3176
Всього капітальних витрат	–	3176

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.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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	12078	6710
2. Витрати на електроенергію	$Z_{ел}$	154	86
3. Витрати на амортизацію	$Z_{ам}$	0	794
Всього витрат за рік	I	12232	7590

Витрати на обслуговування системи:

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

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

Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 90 годин на рік до 50 годин на рік, тому витрати на технічне обслуговування зменшилися з:

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

до:

$$Z_{p \text{ нов}} = 50 \cdot 100 \cdot 1,1 \cdot 1,22 = 6710 \text{ грн}.$$

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 ($C_{ел}$):

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

$$Z_{ел \text{ баз}} = 0,45 \cdot 90 \cdot 3,8 = 154 \text{ грн}.$$

$$Z_{ел \text{ нов}} = 0,45 \cdot 50 \cdot 3,8 = 86 \text{ грн}.$$

$$T_e = \frac{479208}{(2647-1707) \cdot 99 \cdot 12 / 3} = 1,3 \text{ років.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	99
2. Повна собівартість розробленої програми	Грн.	1707
3. Ціна розробленої програми	Грн.	2647
4. Плановий прибуток від реалізації розробленої програми	Грн.	940
5. Рентабельність програмної продукції	%	55
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2523008
7. Загальний прибуток від реалізації програмної продукції	Грн.	93060
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	30651
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,3
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3176
11. Величина економічного ефекту у користувача програмної продукції	Грн.	3848
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,7

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

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

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

$$E_{cn} = (12232 - 7590) - 0,25 \cdot 3176 = 3848 \text{ грн.}$$

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

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

$$T_{cn} = \frac{3176}{12232 - 7590} = 0,7 \text{ року.}$$

7.9 Висновки

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

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – система збереження життя і здоров'я працівників у процесі трудової діяльності, що включає правові, соціально-економічні, організаційні, технічні, санітарно-гігієнічні, лікувально-профілактичні, реабілітаційні та інші заходи. Загальні положення державної політики, щодо галузі охорони праці зазначені у Законі України “Про охорону праці”. Цей Закон визначає основні положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності, на належні, безпечні і здорові умови праці, регулює за участю відповідних органів державної влади відносини між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні [20]. Законодавство про працю містить норми і вимоги з техніки безпеки і виробничої санітарії, норми, що регулюють робочий час і час відпочинку, звільнення та переведення на іншу роботу, норми праці щодо жінок, молоді, гігієнічні норми і правила тощо. Загальний нагляд за додержанням норм охорони праці покладено на прокуратуру, спеціальний – на професійні спілки. Контроль за безпекою праці здійснюють також, державні й відомчі спеціалізовані інспекції. Науково-технічний прогрес вніс серйозні зміни в умови виробничої діяльності робітників розумової діяльності. Їх праця стала більш інтенсивною, напруженою і вимагає значних витрат розумової, емоційної і фізичної енергії. Це призвело до необхідності у знаходженні комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку. Охорона здоров'я робітників, забезпечення безпеки умов праці, ліквідація та профілактика професійних захворювань і виробничого травматизму складає одну з головних турбот людського суспільства.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

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

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

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат;

праці під час експлуатації електронно-обчислювальних машин»). Таним чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Ia. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

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

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

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Ia			Фактичні		
	Температура, °С	Вологість, %	Швидкість повітря, м/с	Температура, °С	Вологість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,1
Тепла	23-25	50-70	0,1	24-25	50-65	0,11

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

В літню пору застосовується кондиціонер.

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

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ.

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

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

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

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

8.4 Техніка безпеки та протипожежна профілактика

Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» будинок можна віднести до II групи по ступені вогнестійкості й до категорії Д по ступені пожежонебезпеки.

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка, поміщена в ізолюваний кабель. Висота проводки становить 2,2м від рівня підлоги, її кріплення здійснюється за допомогою металевих власників. Біля кожного стола організований розподільний щит, розташований на текстолітовій пластинці, закріпленої на стіні на рівні 1м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення. Всі обчислювальні машини з'єднані із клемми заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 V, а одна, забезпечує подачу напруги в 36 V. Про це є відповідні написи на кожному розподільному щиті.

Робота обслуговуючого персоналу полягає в інсталяції необхідного програмного забезпечення й наступному його використанні в діалоговому режимі роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат. З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність устаткування, його електробезпечність; достатня пожежобезпечність приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки. До небезпечних і шкідливих факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електроні справності встаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрана дисплея й робочого місця.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6], приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення, сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів по способі захисту людини від поразки електричним струмом, оскільки їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однієї з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту. Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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, 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, $S = 42$ м²;

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

A – ширина приміщення, $A = 6$ м;

B – довжина приміщення, $B = 7$ 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: $F=90391$ Лм.

Для розрахунку дудемо використовувати світлодіодні стельові панелі Призма-72 6400К, світловий потік яких $F_n = 7200$ Лм.

Число ламп визначається по формулі:

$$N=F/F_n$$

де: F – світловий потік,

F_n – 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 вцілому.

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.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 19.09.22).

10. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу : <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення: 16.10.2022).

КБПЗ – 2023

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

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 синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.
- Досліджена система синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.
- На основі отриманих результатів досліджень створена програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

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.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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

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

Програма реалізована на мові високого рівня Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Android.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Madryga.

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 3848 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,7 роки.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Якубенко Я.І. Дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. JJ Geewax. API Design Patterns. Manning Publications Co. 2021. 481 p.
3. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
4. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
5. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
6. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
7. 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.
8. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
9. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.
10. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
11. 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.
12. 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.

13. 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.

14. 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>.

15. 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

16. 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.

17. 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.

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

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobayev, 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.

28. 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.

29. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

30. 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.

31. 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.

32. 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.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

33. 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.

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

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

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

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

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

39. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

40. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for DOM XSS vulnerabilities.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

Scientific & practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp. [Электронный Журнал]. Georgia. Tbilisi: SCSA – 2018.

41. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.

42. Смірнов О.А., Коваленко О.В., Коваленко А.С., Смірнов С.А. Розробка методу передтестової компіляції й розподілу доступу. Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215

43. Smirnov Oleksii, Kovalenko Oleksandr, Kovalenko Anna, Smirnov Serhii. Method of testing the DOM XSS vulnerability. International Conference «Information technologies, systems and networks ITSН-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. 2017. P7.

44. Смірнов О.А., Смірнов С.А., Коваленко О.В., Коваленко А.С. Технологія тестування DOM XSS уразливості. Науково-практичний журнал кібер безпеки (SPCSJ) № 1. [Електронний журнал]. Грузія. Тбілісі: SCSA - 2017.

45. Смірнов О.А., Лисенко І.А. Інформаційна технологія проектування тестових наборів з урахуванням вимог до програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 4 (44). - Полтава: ПолтНТУ. - 2017. - С. 112-115.

46. Смірнов О.А., Смірнов С.А., Рябой Д.К., Рябая О.В. Модель вузла комутації з відносними пріоритетами, резервуванням ресурсів і обліком реальної надійності обслуговуючих приладів .Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті».

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.

47. 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.

48. 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.

49. 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.

50. 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.

51. 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.

52. 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.

53. 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.

					ВКРМ-122.23.0066.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

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-122.23.0066.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Якубенко Я.І.				<i>Дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android</i>	Літ.	Аркуш	Аркушів
Перевірів	Смірнов О.А.					М	1	6
Н. Контр.	Коваленко А.С.					ЦНТУ КН-22МЗ		
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

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. № 37-13 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 синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

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-122.23.0066.00.00.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 синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android;
- 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-122.23.0066.00.00.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 мобільних пристроях під ОС Android 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 Android.

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.

					ВКРМ-122.23.0066.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

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

					ВКРМ-122.23.0066.00.00.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 – 94 аркуші.

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 15.12.2023 р.

					ВКРМ-122.23.0066.00.00.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

_____ Смірнов О.А.

*Дослідження та програмна реалізація
системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС
Android*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 40

Літера: РП

Кропивницький – 2023 року

ФАЙЛ ГОЛОВНОГО ВІКНА ПЗ – MAIN.PAS

```

unit main;
{
Якубенко Ярослав Ігорович
Дослідження та програмна реалізація системи синтезу та аналізу QR-кодів
на мобільних пристроях під ОС Android
Центральноукраїнський національний технічний університет
Кропивницький 2023 року
}
Interface // інтерфейс

Uses // бібліотеки
  SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  QR_Android, 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 QR_Android1Change(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure BtnFontClick(Sender: TObject);
    procedure ComboBox3Change(Sender: TObject);
  private
    { Декларация-Private }
    QR_Android1 : TAsQR_Android;
    procedure print_demo(bc:TAsQR_Android);
  public
    { Декларация-Public }
  end;

var
  Form1: TForm1;

```

Implementation

```

Uses // бібліотеки
  WinTypes, WinProcs, Printers, QR_Android2;

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  {
  Створити QR_Android об'єкта.
  це об'єкт звільняється, коли форма буде зруйнована
  }
  QR_Android1 := TQR_Android.Create(self);
  QR_Android1.Top := 50;
  QR_Android1.Left := 30;
  QR_Android1.Typ := bcCodePostNet;
  QR_Android1.Modul := 2;
  QR_Android1.Ratio := 2.0;
  QR_Android1.Height := 50;
  QR_Android1.OnChange := Self.QR_Android1Change;
  ComboBox2.ItemIndex := integer(QR_Android1.ShowText);
end;

procedure TForm1.ComboBox1Click(Sender: TObject);
begin
  QR_Android1.Typ := TQR_AndroidType(ComboBox1.ItemIndex);
end;

procedure TForm1.Edit2Change(Sender: TObject);
begin
  try
    QR_Android1.Angle := StrToFloat(Edit2.Text);
  except
    QR_Android1.Angle := 0.0;
  end;
end;

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  QR_Android1.Top := Y;
  QR_Android1.Left := X;
end;

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

procedure TForm1.ComboBox2Click(Sender: TObject);
begin
  QR_Android1.ShowText := TQR_AndroidOption(ComboBox2.ItemIndex);
end;

procedure TForm1.CBChecksumClick(Sender: TObject);
begin
  QR_Android1.Checksum := CBChecksum.Checked;
end;

procedure TForm1.print_demo(bc:TAsQR_Android);
var
  tmpQR_Android : TAsQR_Android;

```

```

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

procedure TForm1.Button2Click(Sender: TObject);
begin
  print_demo(QR_Android1);
end;

procedure TForm1.EditWidthEnter(Sender: TObject);
begin
  EditWidth.Text := IntToStr(QR_Android1.Width);
end;

procedure TForm1.EditWidthExit(Sender: TObject);
begin
  QR_Android1.Width := StrToInt(EditWidth.Text);
  EditWidth.Text := IntToStr(QR_Android1.Width);
end;

procedure TForm1.QR_Android1Change(Sender: TObject);
begin
  Image1.Picture := nil;
  QR_Android1.DrawQR_Android(Image1.Canvas);
end;

procedure TForm1.Edit1Change(Sender: TObject);
begin
  QR_Android1.Text := Edit1.Text;
end;

procedure TForm1.BtnFontClick(Sender: TObject);
begin
  with FontDialog1 do
    begin
      Font := QR_Android1.ShowTextFont;
      if Execute then
        QR_Android1.ShowTextFont := Font;
    end;
end;
end;

```

```
procedure TForm1.ComboBox3Change(Sender: TObject);  
begin  
    QR_Android1.ShowTextPosition := TShowTextPosition(ComboBox3.ItemIndex);  
end;  
end.
```

К6П3_2023

ОСНОВНИЙ ПРОЕКТ ПЗ

```
program QR_Android_DIPLOM;
{
Якубенко Ярослав Ігорович
Дослідження та програмна реалізація системи синтезу та аналізу QR-кодів
на мобільних пристроях під ОС Android
Центральноукраїнський національний технічний університет
Кропивницький 2023 року
}

Uses // бібліотеки
  Forms,
  main in 'main.pas' {Form1},
  QR_Android1 in 'QR_Android1.pas', // перша бібліотека, налаштування QR коду
  QR_Android2 in 'QR_Android2.pas'; // друга бібліотека, генерування QR коду

{$R *.RES} // ресурси

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

КБПЗ_2023

ФАЙЛ БІБЛІОТЕКИ - QRCODE1.PAS

```

unit QR_Android1;
{
Якубенко Ярослав Ігорович
Дослідження та програмна реалізація системи синтезу та аналізу QR-кодів
на мобільних пристроях під ОС Android
Центральноукраїнський національний технічний університет
Кропивницький 2023 року
}
interface

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

type
  TQRAsQR_Android = class(TQRImage)
  private
    FQR_Android :TASQR_Android;
    FIgnoreOnChange :Boolean;
    procedure FixupSize;
    procedure QR_AndroidToCanvas;
    procedure SetQR_Android(const Value: TASQR_Android);
    function GetQR_Android: TASQR_Android;
    function GetAngle: double;
    function GetChecksum: boolean;
    function GetChecksumMethod: TChecksumMethod;
    function GetColor: TColor;
    function GetColorBar: TColor;
    function GetModul: integer;
    function GetRatio: double;
    function GetShowText: TQR_AndroidOption;
    function GetText: string;
    function GetTyp: TQR_AndroidType;
    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: TQR_AndroidOption);
    procedure SetText(const Value: string);
    procedure SetTyp(const Value: TQR_AndroidType);
    function GetQR_AndroidHeight: Integer;
    function GetQR_AndroidWidth: Integer;
    procedure SetQR_AndroidHeight(const Value: Integer);
    procedure SetQR_AndroidWidth(const Value: Integer);
  protected
    procedure Loaded; override;
    procedure Resize; override;
    procedure OnQR_AndroidChange(Sender :TObject);
  public
    constructor Create(AOwner :TComponent); override;
    property QR_Android :TASQR_Android read GetQR_Android write SetQR_Android;
  published
    property Angle :double read GetAngle write SetAngle;
    property QR_AndroidHeight :Integer read GetQR_AndroidHeight write
SetQR_AndroidHeight;
    property QR_AndroidWidth :Integer read GetQR_AndroidWidth write
SetQR_AndroidWidth;
    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
clBlack;
        property Modul :integer read GetModul write SetModul;
        property Ratio :double read GetRatio write SetRatio;
        property ShowText :TQR_AndroidOption read GetShowText write
SetShowText default bcoNone;
        property Text :string read GetText write SetText;
        property Typ :TQR_AndroidType read GetTyp write SetTyp default
bcCode_2_5_interleaved;
        property Picture stored False;
    end;

implementation

Uses // бібліотеки
    Math;

end;

{ реалізація коду TQRAsQR_Android }
procedure TQRAsQR_Android.QR_AndroidToCanvas;
begin
    //очищення канви
    Picture.Assign(nil);
    //рисування QR_Android
    FixupSize;
    QR_Android.DrawQR_Android(Self.Canvas);
end;
constructor TQRAsQR_Android.Create(AOwner :TComponent);
begin
    inherited;
    FQR_Android := TQR_Android.Create(nil);
    FQR_Android := TQR_Android.Create(Self);
    FQR_Android.OnChange := OnQR_AndroidChange;
    FQR_Android.Height := 50; // по замовчанню
    FQR_Android := TQR_Android.Create(Self);
    FQR_Android.OnChange := OnQR_AndroidChange;
    FQR_Android.Height := 50; // по замовчанню
end;
{
destructor TQRAsQR_Android.Destroy;
begin
    inherited;
end;
}
procedure TQRAsQR_Android.FixupSize;
begin
    if (not (csLoading in ComponentState)) then
        begin
            FIgnoreOnChange := True;
            try
                // Налаштування ширини QR_Android
                Height := QR_Android.CanvasHeight;
                Width := QR_Android.CanvasWidth;
            finally
                FIgnoreOnChange := False;
            end;
        end;
end;

function TQRAsQR_Android.GetAngle: double;
begin
    Result := FQR_Android.Angle;
end;

```

```
end;

function TQRAsQR_Android.GetQR_Android: TASQR_Android;
begin
    Result := FQR_Android;
end;

function TQRAsQR_Android.GetQR_AndroidHeight: Integer;
begin
    Result := QR_Android.Height;
end;

function TQRAsQR_Android.GetQR_AndroidWidth: Integer;
begin
    Result := QR_Android.Width;
end;

function TQRAsQR_Android.GetChecksum: boolean;
begin
    Result := QR_Android.CheckSum;
end;

function TQRAsQR_Android.GetChecksumMethod: TChecksumMethod;
begin
    Result := QR_Android.CheckSumMethod;
end;

function TQRAsQR_Android.GetColor: TColor;
begin
    Result := QR_Android.Color;
end;

function TQRAsQR_Android.GetColorBar: TColor;
begin
    Result := QR_Android.ColorBar;
end;

function TQRAsQR_Android.GetModul: integer;
begin
    Result := QR_Android.Modul;
end;

function TQRAsQR_Android.GetRatio: double;
begin
    Result := QR_Android.Ratio;
end;

function TQRAsQR_Android.GetShowText: TQR_AndroidOption;
begin
    Result := QR_Android.ShowText;
end;

function TQRAsQR_Android.GetText: string;
begin
    Result := QR_Android.Text;
end;

function TQRAsQR_Android.GetTyp: TQR_AndroidType;
begin
    Result := QR_Android.Typ;
end;

procedure TQRAsQR_Android.Loaded;
```

```

begin
  inherited;
  QR_AndroidToCanvas;
end;

procedure TQRAsQR_Android.OnQR_AndroidChange(Sender: TObject);
begin
  if FIgnoreOnChange then EXIT;
  if (not (csLoading in ComponentState)) then
    QR_AndroidToCanvas;
end;

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

procedure TQRAsQR_Android.SetAngle(const Value: double);
begin
  QR_Android.Angle := Value;
end;

procedure TQRAsQR_Android.SetQR_Android(const Value: TQR_Android);
begin
  FQR_Android.Assign(Value);
end;

procedure TQRAsQR_Android.SetQR_AndroidHeight(const Value: Integer);
begin
  QR_Android.Height := Value;
end;

procedure TQRAsQR_Android.SetQR_AndroidWidth(const Value: Integer);
begin
  QR_Android.Width := Value;
end;

procedure TQRAsQR_Android.SetChecksum(const Value: boolean);
begin
  QR_Android.CheckSum := Value;
end;

procedure TQRAsQR_Android.SetChecksumMethod(const Value: TChecksumMethod);
begin
  QR_Android.CheckSumMethod := Value;
end;

procedure TQRAsQR_Android.SetColor(const Value: TColor);
begin
  QR_Android.Color := Value;
end;

procedure TQRAsQR_Android.SetColorBar(const Value: TColor);
begin
  QR_Android.ColorBar := Value;
end;

procedure TQRAsQR_Android.SetModul(const Value: integer);
begin
  QR_Android.Modul := Value;
end;

```

```
procedure TQRAsQR_Android.SetRatio(const Value: double);
begin
  QR_Android.Ratio := Value;
end;

procedure TQRAsQR_Android.SetShowText(const Value: TQR_AndroidOption);
begin
  QR_Android.ShowText := Value;
end;

procedure TQRAsQR_Android.SetText(const Value: string);
begin
  QR_Android.Text := Value;
end;

procedure TQRAsQR_Android.SetTyp(const Value: TQR_AndroidType);
begin
  QR_Android.Typ := Value;
end;
end.
```

K6П3_2023

ФАЙЛ БІБЛІОТЕКИ - QRCODE2.PAS

```

unit QRCODE2;
{
Якубенко Ярослав Ігорович
Дослідження та програмна реалізація системи синтезу та аналізу QR-кодів
на мобільних пристроях під ОС Android
Центральноукраїнський національний технічний університет
Кропивницький 2023 року
}
interface

Uses // бібліотеки
  Classes, Graphics;

type
  TQR_AndroidType =
  (
    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
  );
  TBarLineStyle = (white, black, black_half);
{
тільки для внутрішнього використання
}
{
black_half означає чорна лінія 2/5
висоти (використовується для PostNet)
}
  TQR_AndroidOption = (bcoNone, bcoCode, bcoType, bcoBoth);
{ Вид тексту, щоб показати }
  TShowTextPosition =
  (
    stpTopLeft,
    stpTopRight,
    stpTopCenter,
    stpBottomLeft,
    stpBottomRight,
    stpBottomCenter
  );
  TChecksumMethod =
  (
    csmNone,

```

```

csmModulo10
);
TAsQR_Android = class(TComponent)
private
  { Декларация-Private }
  FHeight : integer;
  FText : string;
  FTop : integer;
  FLeft : integer;
  FModul : integer;
  FRatio : double;
  FTyp : TQR_AndroidType;
  FChecksum:boolean;
  FShowText:TQR_AndroidOption;
  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: TQR_AndroidType);
  procedure SetAngle(const Value: Double);
  procedure SetText(const Value: string);
  procedure SetShowText(const Value: TQR_AndroidOption);
  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-декларация }
    function MakeData : string;
    procedure DoChange; virtual;
public
    { Декларация-Public }
    constructor Create(Owner:TComponent); override;
    destructor Destroy; override;
    procedure Assign(Source: TPersistent);override;
    procedure DrawQR_Android(Canvas:TCanvas);
    procedure DrawText(Canvas:TCanvas);
    property CanvasHeight :Integer read GetCanvasHeight;
    property CanvasWidth :Integer read GetCanvasWidth;
published
    { Published- Декларация }
{
висота QR_Android (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 QR_Android
}
    property Modul : integer read FModul write SetModul;
    property Ratio : Double read FRatio write SetRatio;
    property Typ : TQR_AndroidType 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:TQR_AndroidOption 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 QR_Android.d32}
{$else}
    {$R QR_Android.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 QR_Android }
    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 при 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 TQR_Android.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 TQR_Android.Destroy;
begin
  FShowTextFont.Free;
  inherited;
end;

procedure TQR_Android.Assign(Source: TPersistent);
var
  BSource : TAsQR_Android;
begin
  if Source is TAsQR_Android then
    begin
      BSource := TAsQR_Android(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 TQR_Android.GetTypText:string;
begin
  result := BCdata[FTyp].Name;
end;

{ встановлення Ширини }
procedure TQR_Android.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 TQR_Android.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 TQR_Android.MakeData : string;
var
  i : integer;
begin
  {
  Розрахунок ліній
  }
  MakeModules;
  { числовий тип QR_Android}
  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('QR_Android повинен бути числовий');
      end;
      {Отримати структуру QR_Android}
      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 QR_AndroidType', [self.ClassName]);
end;
Showmessage(Format('Data <%s>', [Result]));

end;

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

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

function TQR_Android.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 TQR_Android.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 TQR_Android.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 QR_Android 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 TQR_Android.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 TQR_Android.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 TQR_Android.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 TQR_Android.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;
  {Стопкод}
  result := result + tabelle_39[FindIdx('*')].data;
end;

```

```

function TQR_Android.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 TQR_Android.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 QR_AndroidType 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 TQR_Android.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 TQR_Android.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 TQR_Android.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 TQR_Android.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 TQR_Android.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 TQR_Android.SetLen(pI:byte):string;
begin
  Result := StringOfChar('0', pI-Length(FText)) + FText;
end;

function TQR_Android.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 TQR_Android.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 TQR_Android.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 TQR_Android.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 TQR_Android.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;
end;

```

```

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 TQR_Android.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 TQR_Android.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;
end;

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

```

```

SavePen := TPen.Create;
SaveBrush := TBrush.Create;
{отримання QR_Android }
data := MakeData;
try
  Savefont.Assign(Canvas.Font);
  SavePen.Assign(Canvas.Pen);
  SaveBrush.Assign(Canvas.Brush);
  DoLines(data, Canvas); {від рисовка QR_Android}
  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 TQR_Android.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 QR_Android}
                TextOut(PosX, PosY, FText); { контекст QR_Android }
              if FShowText in [bcoTyp, bcoBoth] then
                TextOut(FLeft, FTop+Round(Font.Height*2.5), GetTypText);
            {тип та ім'я QR_Android}
            finally
              Font.Assign(SaveFont);

```

```

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

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

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

procedure TQR_Android.SetTyp(const Value: TQR_AndroidType);
begin
    if Value <> FTyp then
    begin
        FTyp := Value;
        DoChange;
    end;
end;

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

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

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

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

```

```

        DoChange;
    end;
end;

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

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

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

function TQR_Android.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 TQR_Android.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 TQR_Android.SetShowTextFont(const Value: TFont);
begin
    FShowTextFont.Assign(Value);
    DoChange;
end;

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

```