

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

**Пояснювальна записка**  
до бакалаврської роботи  
на ступінь вищої освіти бакалавр  
на тему: **«РОЗРОБКА МОБІЛЬНОЇ ГРИ ДЛЯ ВИВЧЕННЯ ОСНОВ  
ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА БАЗІ ANDROID  
DEVELOPMENT TOOLS ДЛЯ ECLIPSE»**

Виконала: студентка 4 курсу, групи ПД-42  
спеціальності:

121 Інженерія програмного забезпечення  
(шифр і назва спеціальності)

Логункова Є.І.

(прізвище та ініціали)

Керівник Негоденко О.В.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Київ – 2021

# ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

## НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність – 121 Інженерія програмного забезпечення

**ЗАТВЕРДЖУЮ**

Завідувач кафедри  
інженерії програмного забезпечення

Негоденко О.В.

“ \_\_\_\_ ” \_\_\_\_\_ 2021 року

### ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Логункова Євгенія Ігорівна

(прізвище, ім'я, по  
батькові)

1. Тема роботи: «Розробка мобільної гри для вивчення основ тестування програмного забезпечення на базі Android Development Tools для Eclipse»

Керівник роботи Негоденко Олена Василівна, завідувач кафедри ІІЗ

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

затверджені наказом вищого навчального закладу від “12” березня 2021 року №65.

2. Строк подання студентом роботи “01” червня 2021 року

3. Вихідні дані до роботи:

Запропоновано комплексне рішення операційної системи;

Наведено модель мобільного програмного комплексу для вивчення основ тестування програмного забезпечення;

Методи аналізу середовища мобільного програмного комплексу для вивчення основ тестування програмного забезпечення

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

4.1 Теоретичні аспекти за темою дослідження

---

4.2 Проектування мобільної гри для вивчення основ тестування програмного забезпечення

---

4.3 Розробка мобільної гри для вивчення основ тестування програмного забезпечення

---

4.4 Тестування мобільної гри для вивчення основ тестування програмного забезпечення

---

5. Перелік графічного матеріалу

1. Титульний слайд

---

2. Актуальність дослідження

---

3. Мета, об'єкт дослідження, предмет дослідження

---

4. Аналіз існуючих аналогів

---

5. Постановка завдань дослідження

---

6. Наукова новизна отриманих результатів

---

7. Проектування мобільної гри

---

8. Діаграма використання

---

9. Діаграма класів режиму тестування

---

10. Архітектура системи «Для вивчення основ тестування програмного забезпечення»

---

11. Діаграма класів режиму тестування

---

12. Результати роботи

---

13. Висновки

---

6. Дата видачі завдання “19” квітня 2021 року

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи до	Примітка
1	Підбір науково-технічної літератури	19.04.2021	
2	Дослідження актуальності теми	25.09.2020	
3	Розгляд аналогів	15.10.2020	
4	Теоретичні аспекти за темою дослідження	05.12.2020	
5	Проектування мобільної гри	20.01.2021	
6	Розробка мобільної гри	01.03.2021	
7	Тестування мобільної гри	07.04.2021	
8	Розробка демонстраційних матеріалів	21.04.2021	
9	Здача роботи в деканат	01.06.2021	

Студент \_\_\_\_\_ Логункова Є.І.  
( підпис ) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_ Негоденко О.В.  
( підпис ) (прізвище та ініціали)





## РЕФЕРАТ

Текстова частина бакалаврської роботи 76 с., 40 рис., 1 табл., 30 джерел.

Ключові слова: ANDROID STUDIO, UMTS, ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, APK

Об'єкт дослідження – процес навчання основам тестування програмного забезпечення.

Предмет дослідження – методи та засоби розробки мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.

Мета роботи – покращити процес вивчення основ тестування програмного забезпечення за допомогою мобільної гри.

Методи дослідження – методи теорії інформації, методи структурного аналізу і проектування, методи розробки програмного забезпечення, методи тестування, валідації та верифікації програмного забезпечення.

Наукова новизна даної роботи полягає в наступному:

- На основі результатів виконаних досліджень розроблено систему навчання та перевірки знань для тестувальників програмного забезпечення.
- Встановлено ефективні методи аналізу середовища мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.
- Показано модель мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.

Упровадження розробленого застосунку допоможе користувачам покращити свої знання з програмування.

В роботі виконано аналіз існуючих застосунків для вивчення основ тестування програмного забезпечення. В результаті аналізу було визначено основні потреби користувачів.

Проаналізовано можливості середовища розробки. Розроблено логіку практичних завдань та загальну концепцію представлення інформації для користувачів застосунку.

Галузь використання – завдяки вільному доступу, застосунок може використовувати будь-яка людина, котра бажає вивчати процес тестування. Результати роботи можуть бути використані при розробці, встановленні та налаштуванні програмного комплексу для вивчення основ тестування програмного забезпечення.



## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	10
ВСТУП .....	11
РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ ЗА ТЕМОЮ ДОСЛІДЖЕННЯ .....	13
1.1 Поняття та сутність мобільної гри для вивчення основ тестування програмного забезпечення .....	13
1.2 Дослідження аналогів.....	17
1.3 Постановка завдань дослідження.....	23
РОЗДІЛ 2 ПРОЕКТУВАННЯ МОБІЛЬНОЇ ГРИ ДЛЯ ВИВЧЕННЯ ОСНОВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	24
2.1 Вибір інструментарію програмування .....	24
2.2 Інфологічна структура мобільної гри для вивчення основ тестування програмного забезпечення .....	35
2.3 Алгоритм роботи мобільної гри для вивчення основ тестування програмного забезпечення .....	40
РОЗДІЛ 3 РОЗРОБКА МОБІЛЬНОЇ ГРИ ДЛЯ ВИВЧЕННЯ ОСНОВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	44
3.1 Формування мобільної гри для вивчення основ тестування програмного забезпечення .....	44
3.2 Тестування мобільної гри для вивчення основ тестування програмного забезпечення.....	57
ВИСНОВКИ.....	66
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	67
ДОДАТКИ.....	70
Додаток А.....	70
Додаток Б .....	74
Додаток В.....	77

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

БД	база даних
ІС	інформаційні системи
ОС	операційна система
ПЗ	програмне забезпечення
USIM	модуль ідентифікації абонента мережі UMTS
PDH	Plesiochronous Digital Hierarchy – плезіохоронна цифрова ієрархія
SDH	Synchronous Digital Hierarchy - синхронна цифрова ієрархія
EDGE	Enhanced Data Rates for GSM Evolution
HSDPA	High-Speed Downlink Packet Access
IMT- 2000	International Mobile Telecommunications 2000

## ВСТУП

*Актуальність дослідження.* На сьогодні, інформаційні технології, різного виду щільно увійшли в життя сучасної людини. Мобільні пристрої, легкі для перенесення та обігу, грають величезну роль в житті людини. Зростання ринку мобільних пристроїв вже перестало бути лінійним. Кількість великих гравців ринку до сьогоднішнього моменту збільшилася в кілька разів порівняно з початком 2000-х. На початку-середині 2000-х, коли мобільні пристрої стали доступні широким масам, для мобільних пристроїв, стали розроблятися різні програми, націлені на вирішення локальних задач, в більшості своїй не пов'язаних з доступом в мережу Інтернет. У середині 2000-х почалося зростання ринку мобільного Інтернету, що привело до створення додатків, заздалегідь орієнтованих на доступ в Інтернет.

Більшість власників сучасних мобільних телефонів використовують їх у якості своїх помічників у повсякденному житті, працюють з програмами для створення розкладів, написання заміток, установок нагадувань і будильників та є спектр користувачів, цільове застосування мобільних технологій яких, направлено на ігрові цілі. Це передусім молодь та діти, хоча застосування ігрових мобільних технологій контингентом більш старшого віку не є виключенням.

Виходячи з того, що майже кожна сучасна людина володіє мобільними технологіями актуальність розробки не виявляється сумнівною.

*Мета та завдання роботи.* Мета проекту – покращити процес вивчення основ тестування програмного забезпечення за допомогою мобільної гри.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- здійснити порівняльну характеристику існуючих мобільних програмних комплексів для вивчення основ тестування програмного забезпечення;

- розробити мобільну гру для вивчення основ тестування програмного забезпечення;
- вивести архітектуру розробки, структури даних, алгоритми ключових модулів та ситуаційного використання.

**Об'єкт та предмет дослідження.** Об'єктом роботи виступає процес навчання основам тестування програмного забезпечення.

Предметом є методи та засоби розробки мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.

**Методи дослідження.** Для вирішення поставлених завдань використовувалися методи теорії інформації, методи структурного аналізу і проектування, методи розробки програмного забезпечення, методи тестування, валідації та верифікації програмного забезпечення.

***Наукова новизна отриманих результатів.***

- На основі результатів виконаних досліджень розроблено систему навчання та перевірки знань для тестувальників програмного забезпечення.
- Встановлено ефективні методи аналізу середовища мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.
- Показано модель мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.

**Практичне значення отриманих результатів.** Результати роботи можуть бути використані при розробці, встановленні та налаштуванні програмного комплексу для вивчення основ тестування програмного забезпечення.

## РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ЗА ТЕМОЮ ДОСЛІДЖЕННЯ

### 1.1 Поняття та сутність мобільної гри для вивчення основ тестування програмного забезпечення

Мобільна гра для вивчення основ тестування програмного забезпечення як феномен культури охоплює широкі верстви суспільства і має важливе соціальне значення, вона є джерелом естетичного і пізнавального досвіду сучасної людини, засобом виховання, невід'ємною частиною соціальної комунікації, з чого формується необхідність і актуальність як теоретичного, так і практичного дослідження різнобічних аспектів діяльності.

Мобільна гра для вивчення основ тестування програмного забезпечення є полем перетину різних областей культури, мистецтва, науки, освіти. За технологією вивчення основ тестування програмного забезпечення можна судити про обізнаність суспільства, адже, будучи синтетичним простором, мобільна гра для вивчення основ тестування програмного забезпечення виходить за рамки традиції, народжуючи сучасні практики, синтезуючи їх і перетворюючи традиційну споглядальність сприйняття класичного програмування до активної участі глядача в навчальному процесі. У зв'язку з модифікацією поняття «Мобільна гра для вивчення основ тестування програмного забезпечення», розширенням його функцій виникає проблема теоретичного переосмислення різних сторін навчальної діяльності в області сучасного програмування.

Аналізуючи теоретичні дослідження в галузі програмування, можна зробити висновок, що наразі вони носять розрізнений характер, не дають можливості оцінити технологію вивчення основ тестування програмного забезпечення як єдину систему. Оскільки в мобільному додатку в єдності працюють концепція, візуальний образ та навчання, виникає потреба в

системному аналізі ситуації, що дозволить вивести проектування і організацію навчання на професійний рівень.

Впровадження мережевих технологій у навчальну практику в даний час призвело до того, що з'явилися нові форми і методи організації навчального процесу. Одна з таких нових форм організації діяльності – Мобільна гра для вивчення основ тестування програмного забезпечення. Ця нова форма дає ряд переваг і практично не обмежений доступ до інформації для всіх зацікавлених користувачів.

Технологія вивчення основ тестування програмного забезпечення, як форма подачі інформації практикується в найрізноманітніших сферах діяльності, в тому числі в бібліотечній практиці, а тепер підхоплена програмістами.

З мистецької точки: Мобільна гра для вивчення основ тестування програмного забезпечення – інформаційний багатофункціональний Інтернет ресурс, який значно полегшує доступ всіх зацікавлених сторін до взаємної та плідної взаємодії.

Використовуючи Мобільну гру для вивчення основ тестування програмного забезпечення та поза діяльності можна розкрити індивідуальні здібності кожного окремого користувача, підвищити його самооцінку, націлити на позитивне сприйняття самого себе і навколишнього світу, включити в різні види роботи, збагатити особистий досвід, знання про різноманітність людської діяльності, сформувані необхідні практичні вміння і навички.

Існує величезна різноманітність мобільних додатків, як по формі, так і за змістом. Створювати мобільні додатки можна різними способами і засобами: використовуючи офісні програми, спеціальні прикладні програми або мережеві ресурси.

Мобільна гра для вивчення основ тестування програмного забезпечення – інформаційний продукт створений засобами мережевого ресурсу як банер, який може виступати в ролі навчального порталу, виставки

робіт, інтерактивної сторінки, онлайн тестування. По суті це простір певної геометричної форми і розміру наповнений ілюстративним або демонстраційним матеріалом, доповнений заголовком, супровідним текстом і має дизайнерське оформлення.

Запропонований інформаційний продукт, в цій роботі являє собою Мобільну гру для вивчення основ тестування програмного забезпечення.

Критерії оцінювання продукту:

1. Відповідність заявленій тематиці
2. Якість ілюстративного матеріалу
3. Естетична привабливість
4. Творчий підхід

Використовуючи даний інформаційний продукт в навчальній практиці необхідно враховувати наступні моменти: Мобільна гра для вивчення основ тестування програмного забезпечення повинна бути не тільки барвистою, інформативною і привабливою, але і змістовною; бажано використовувати цю форму в процесі навчання не заради самої форми, а в контексті певної теми вивчаючи і освоюючи основні поняття і прийоми роботи; якщо продукт буде соціально значущим це додасть йому безсумнівну вагу, а автору можливість самореалізації.

При створенні даного продукту в процес роботи можна залучити необмежену кількість програмістів з різних напрямків. На етапі розробки концепції Мобільної гри для вивчення основ тестування програмного забезпечення можна організувати раду, на якій самі програмісти відберуть, ті роботи, які увійдуть у базу знань. Пораду можна зробити, як очною, так і заочною, скориставшись мережевими ресурсами, запропонувавши відвідувачам оцінити роботи за допомогою критеріїв оцінки.

Створення Мобільної гри для вивчення основ тестування програмного забезпечення допоможе програмістам продемонструвати в повному обсязі знання і навички роботи з графічними і мультимедійними продуктами,

отриманими в навчальному процесі, проявити свою креативність і творчу активність.

Створення Мобільної гри для вивчення основ тестування програмного забезпечення процес не миттєвий, він вимагає підготовки матеріалів для наповнення, створення концепції мобільної гри та наявність певних знань і досвіду роботи з мережевими сервісами.

При створення інформаційного продукту можна дотримуватися наступного алгоритму:

1. Розробка концепції Мобільної гри для вивчення основ тестування програмного забезпечення, вибір сервісу

2. Підготовка ілюстративного матеріалу (сканування, обробка, створення ілюстративного матеріалу, підготовка тексту, створення робочої папки,)

3. Технічна підготовка проекту (наявність необхідних технічних засобів і доступу до мережі Інтернет)

3. Створення Мобільної гри для вивчення основ тестування програмного забезпечення (Робота з обраним сервісом: реєстрація, розміщення матеріалів в обраний шаблонах, наповнення змістом, дизайн проекту, збереження)

4. Проведення презентації або публікація проекту в Інтернеті

При роботі над Мобільною грою для вивчення основ тестування програмного забезпечення, велика увага приділяється дизайнерським рішенням: за допомогою кольору, графіки, анімації з'являється можливість створити настрій, привернути увагу, підкреслити і виділити важливі моменти. Використання анімаційних заставок допомагає Мобільну гру для вивчення основ тестування програмного забезпечення зробити живою і динамічною.

Мультимедійність Мобільної гри для вивчення основ тестування програмного забезпечення дозволить більш ефективно вирішувати змістовні завдання, а інтерактивність, допомагає залучити більше програмістів в



процес, створити зворотний зв'язок. Познайомившись із запропонованою інформацією, користувач може має можливість: висловити свою думку, додати свій матеріал, взяти участь в обговоренні в режимі «он лайн».

Представлений матеріал повинен бути якісним.

## 1.2 Дослідження аналогів

Новим напрямком програмного забезпечення тестування є застосування мобільних сервісів, до яких можна віднести такі:

iSpring Play (раніше iSpring Viewer) - мобільний додаток, що дозволяє переглядати курси і тести, створені в iSpring і розміщені в мережі інтернет, а також зберігати їх на пристрої для подальшого перегляду без підключення до інтернету.

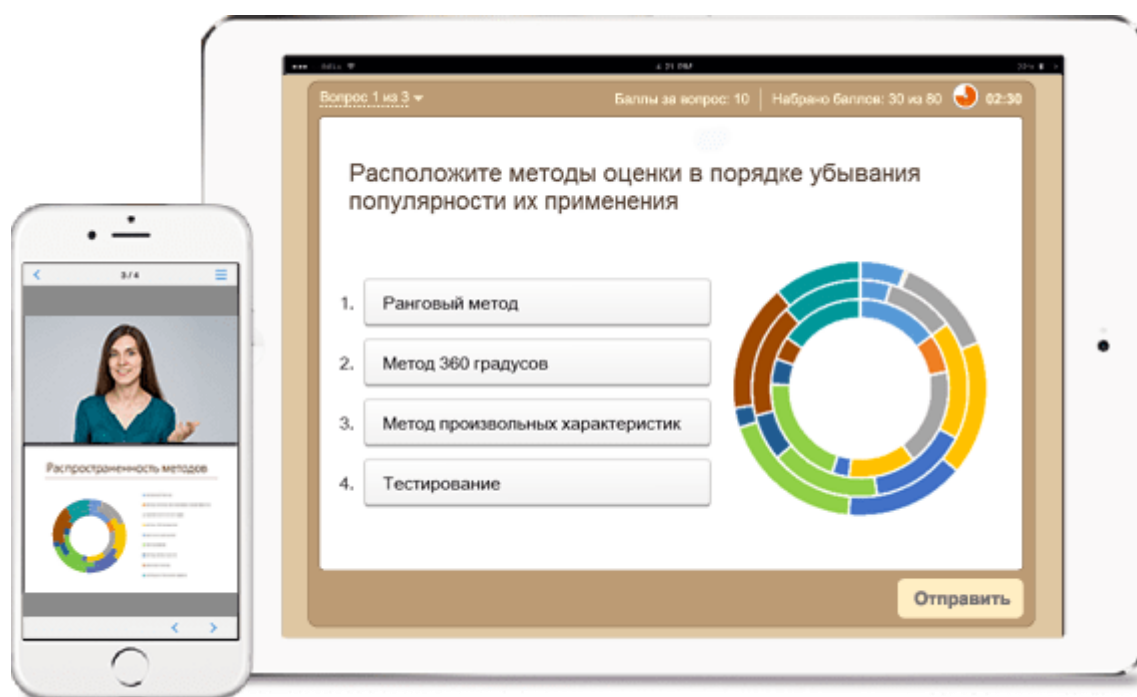


Рисунок 1.1 iSpring Play

Додаток забезпечує швидкий доступ до матеріалів на пристрої. З iSpring Play навчання дійсно стає мобільним: студенти можуть завантажити курси на планшет і переглядати їх навіть офлайн.

Цей додаток розроблено для наступних пристроїв:  
iPhone і планшети Apple iPad з операційною системою iOS версії 6.0 і вище.

Планшети і смартфони, що працюють під управлінням операційної системи Android версії 4.0 і вище.

Додаток iSpring Play доступно для безкоштовного завантаження через онлайн-магазини App Store (для iOS) і Google Play (для Android).

Можливості iSpring Play:

Збереження матеріалів на пристрої для навчання оффлайн.

Усі курси і тести, створені в програмах iSpring, в повноекранному режимі.

Зручна бібліотека для зберігання переглянутих курсів.

Додавання в Вибране, щоб важливі курси були завжди під рукою.

Кілька простих операцій дозволять швидко перетворити курс / тест / презентацію в мобільний формат, і студенти зможуть зануритися в процес дистанційного навчання з звичних планшетів.

Асистент – тестування. Мобільна заміна застарілого настільного додатку Асистент (Assist), призначеного для тренування рішення тестів в форматах \* .qst і \* .qsz, які активно використовуються при навчанні у вищих навчальних закладах СНД.

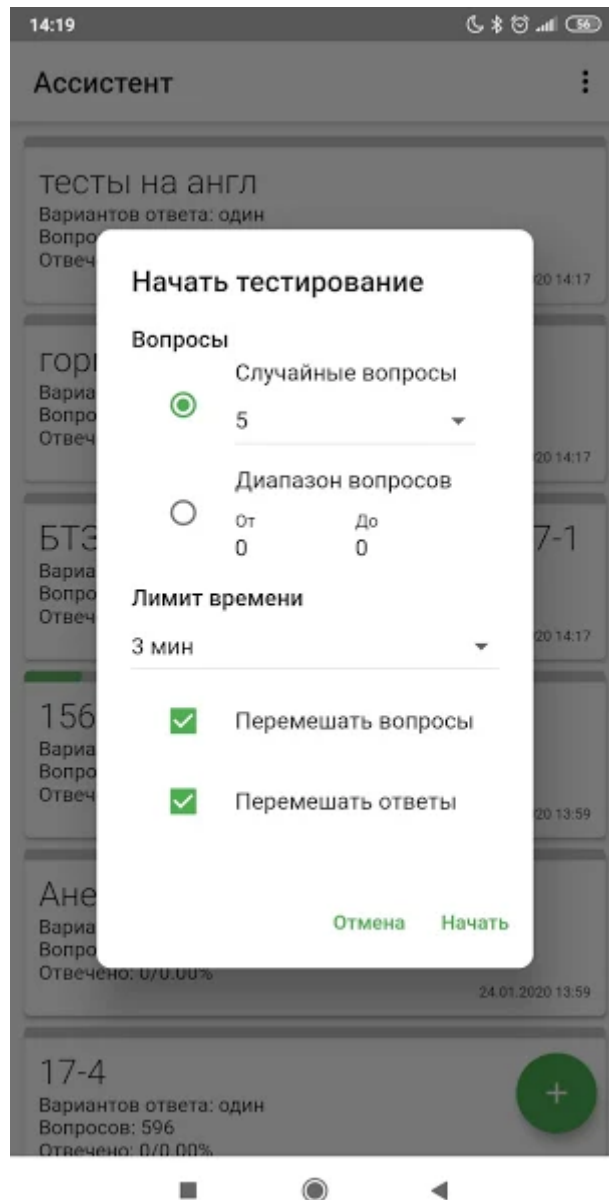


Рисунок 1.2 Ассистент - тестування

Основні можливості:

- читання обох форматів qst і qsz
- перегляд вмісту тесту з усіма варіантами відповідей
- перегляд вмісту тесту тільки з правильними відповідями
- тренувальний тест або іспит
- можливість вирішити всі питання або кілька випадкових
- перегляд кількості питань із загальної кількості на які правильно відповіли
- обмеження часу на рішення тесту

Kahoot! - це сервіс не тільки для створення різноманітних форм тестів, але і для проведення онлайн вікторин за допомогою спеціального клієнта, що встановлюється на смартфонах учнів (Android, iOS, Windows Phone). Дозволяє викладачеві діагностувати відповіді учнів, а тому, якого навчають дізнаватися правильні відповіді. Безкоштовний, вимагає реєстрації.



Рисунок 1.3 Kahoot!

Quizlet - це сервіс для створення тестів, в яких вибирається правильна відповідь із запропонованих варіантів. Сервіс безкоштовний, багатомовний, головна його особливість в тому, що він працює на смартфонах під Android та iOS і вимагає реєстрації.

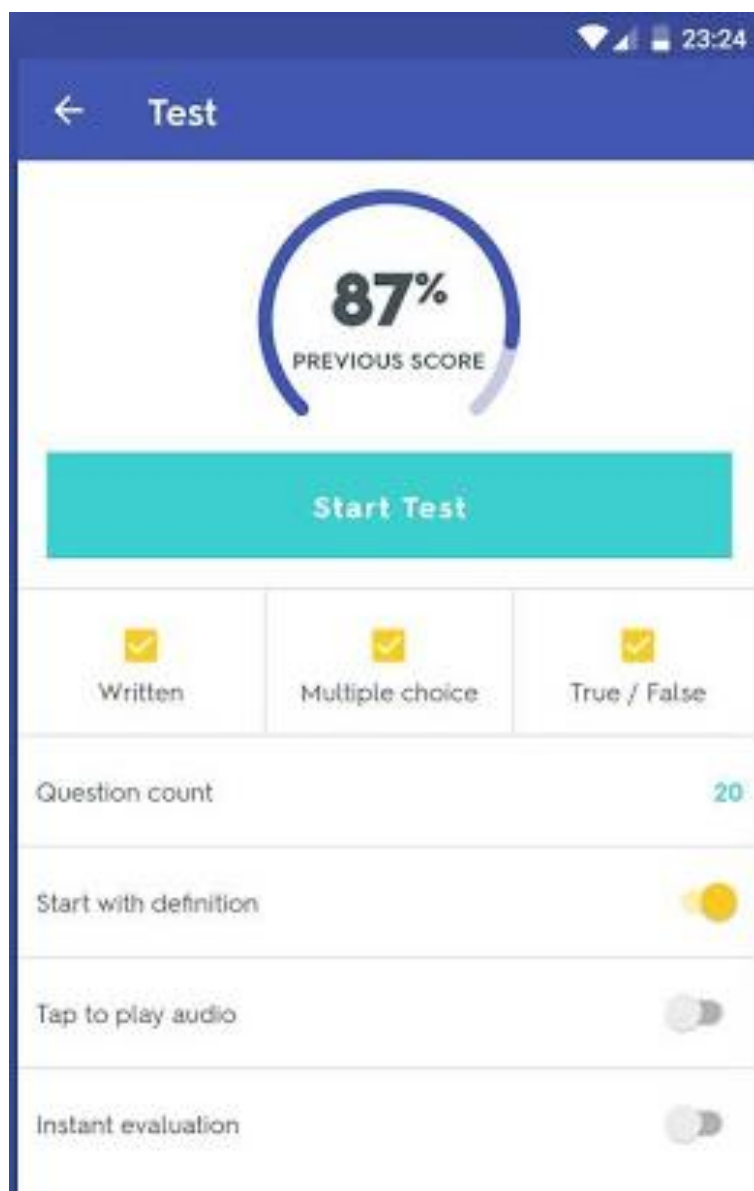


Рисунок 1.4 Quizlet

Plickers - це сервіс у вигляді мобільного додатку, який дозволяє влаштувати опитування на смартфонах учнів в аудиторії і аналізувати викладачеві статистику по всіх студентах. Додаток для Android і iOS завантажується безкоштовно.

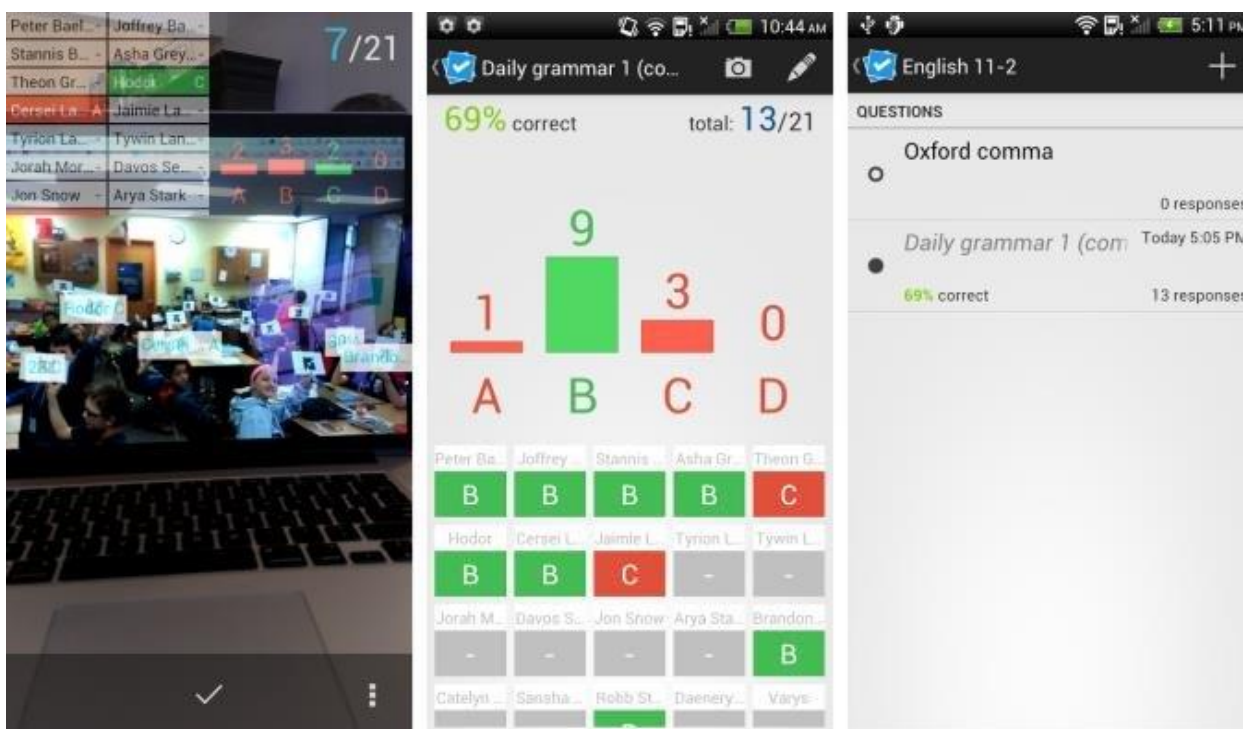


Рисунок 1.5 Plickers

## Name of the new exam

<https://www.onlineexambuilder.com/name-of-the-new-exam/exam-166811>

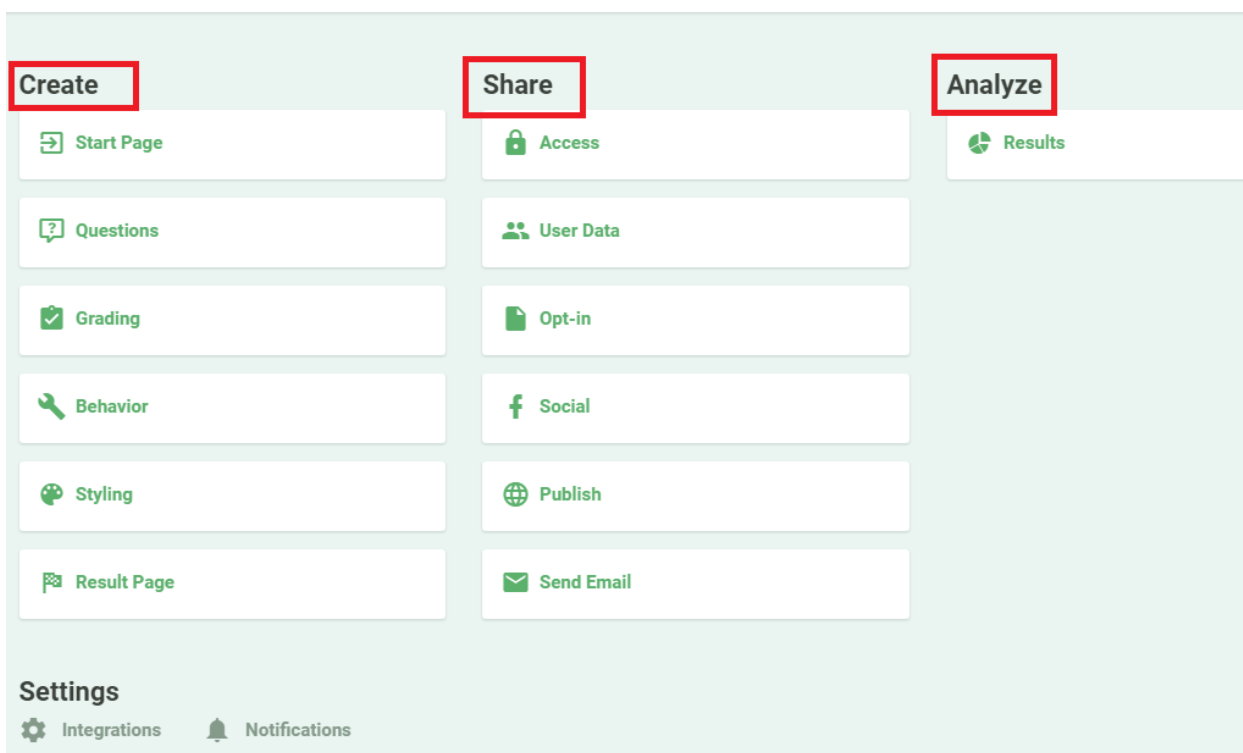


Рисунок 1.6 Easy Test Maker

Easy Test Maker - це сервіс у вигляді мобільного додатку для створення тестів, в яких потрібно вибирати вірні і помилкові твердження. Щоб провести тестування в більш традиційному форматі їх можна відформатувати в форматах .pdf або .doc. для роздрукування. Безкоштовний на період ознайомлення.

### **1.3 Постановка завдань дослідження**

Потрібно розробити клієнт-серверний додаток на базі ОС Android, де клієнт може отримати інформацію, про історію програмування, про видатних програмістів, про генезис мов програмування тощо. Пройти тестування за отриманими знаннями. Взяти участь у вікторинах.

Програмний засіб має забезпечувати наступні функціональні можливості:

1. відображення історичної інформації
2. зчитування інформації з поля введення
3. реалізація тестування
4. зчитування даних відповідей
5. відображення інформації про результат проходження тестування.

## РОЗДІЛ 2.

### ПРОЕКТУВАННЯ МОБІЛЬНОЇ ГРИ ДЛЯ ВИВЧЕННЯ ОСНОВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 2.1 Вибір інструментарію програмування

Для початку необхідно визначитися, на яких пристроях буде функціонувати система. Для цього доцільно висунути ряд вимог до апаратної платформи і знайти рішення, яке найбільшою мірою їм задовольняє.

По-перше, пристрій повинен надавати оперативний і надійний доступ до додатку.

По-друге, необхідна наявність досить потужного процесора і оперативної пам'яті для здійснення необхідних обчислень і досягнення необхідної швидкодії. Вбудована пам'ять, використовувана для зберігання операційної системи, програмного комплексу і користувальницьких даних повинна бути досить великою. Бажано наявність слоту для підключення карти пам'яті з метою збільшення обсягу пам'яті.

По-третє, пристрій повинен так само володіти хорошим дисплеєм для відображення інформації в денний і нічний час. Батарея повинна тримати заряд досить тривалий час.

Всім перерахованим вимогам достатньою мірою задовольняють цифрові пристрої, звані комунікаторами або смартфонами.

Про вибір операційної системи мова піде далі.

Операційна система Symbian, яка беззастережно лідирувала на ринку до 2010 року в даний час майже повністю втратила свої позиції. Спершу Sony Ericsson оголосив про припинення використання Symbian, щоб зосередитися на випуску пристроїв під управлінням Android. Слідом за ним Samsung припиняє підтримку Symbian і переходить на використання Android, Windows Phone 7 і Bada, ОС власної розробки.



Навесні 2011 року сама Nokia оголосила про те, що Windows Phone 7 стане ключовою платформою для смартфонів, проте компанія не планує відмовлятися від платформи Symbian і стане підтримувати мобільну платформу Symbian «щонайменше» до 2020 року.

Незважаючи на це, перспектив у якісному розвитку даної операційної системи не передбачається. І, незважаючи на те, що розробка автоматизованої системи можлива і в рамках даної платформи, від цього варіанту варто відмовитися.

Apple iOS (до 2010 року відома як iPhone OS) мобільна операційна система, розроблена американською компанією Apple на основі Mac OS X спочатку для iPhone, а потім розширена для підтримки таких мобільних пристроїв, як Apple iPod, iPad і Apple TV і в даний час динамічно розвивається.

Розробка додатків для iOS ведеться в середовищі XCode, який підтримує мови C, C++, Objective-C, Objective-C++, Java, AppleScript, Python і Ruby з різними моделями програмування.

Apple не ліцензує iOS для установки на стороннє обладнання, а ціна комунікатора iPhone в Україні досить велика. Цей факт безумовно сильно ускладнює впровадження автоматизованої системи, розробленої для iOS з фінансової точки зору. Тому дана ОС так само не була обрана.

Набирає деяку популярність операційна система Windows Phone 7, проте вона має низку недоліків, серед яких необхідно виділити відсутність вбудованої підтримки баз даних. SQL Server Compact не включено до складу WP7. Також не включені SQLite або будь-які інші сторонні СУБД. Відсутня так само підтримка взаємодії між процесами (IPC) і немає підтримки сокетів.

Продовжуючи аналізувати ринок операційних систем для смартфонів, можна прийти до доцільності використання ОС Android.

Операційна система Android – портативна (мережева) операційна система для комунікаторів, планшетних комп'ютерів, цифрових програвачів, наручних годинників. Спочатку розроблялася компанією Android Inc., яку

потім купила Google. Згодом Google ініціювала створення альянсу Open Handset Alliance (ОНА), який зараз і займається підтримкою і подальшим розвитком платформи.

Частки ринку, які займали операційні системи в певні квартали, наведені в таблиці 2.1.

Таблиця 2.1 – Частки ринку операційних систем для смартфонів

Квартал	Android	iOS	Symbian	Microsoft	Bada	RIM	Other
2016 Q2	0%	5,20%	62,30%	11,90%	0%	12,90%	20,60%
2016 Q3	0%	4,60%	49,50%	10,40%	0%	13,10%	35,50%
2016 Q4	0%	2,80%	57,50%	12,10%	0%	15,40%	27,60%
2017 Q1	0,60%	13,10%	50,30%	11,20%	0%	18,20%	24,80%
2017 Q2	1,10%	10,60%	46,50%	12,20%	0%	19,70%	29,60%
2017 Q3	1,60%	10,50%	48,80%	10,20%	0%	19,70%	28,90%
2017 Q4	1,80%	13,00%	51,00%	9,30%	0%	20,50%	24,90%
2018 Q1	3,40%	17,00%	44,20%	7,90%	0%	19,00%	27,50%
2018 Q2	7,60%	16,20%	44,70%	7,90%	0%	20,60%	23,60%
2018 Q3	9,60%	15,30%	44,20%	6,80%	0%	19,30%	24,10%
2018 Q4	17,20%	14,20%	41,20%	5,00%	0,90%	16,10%	21,50%
2019 Q1	25,30%	16,60%	36,30%	2,80%	1,10%	17,50%	17,90%
2019 Q2	31,10%	16,10%	32,90%	3,40%	1,30%	11,60%	15,20%
2019 Q3	36,00%	16,80%	27,40%	3,60%	1,70%	10,90%	14,50%
2019 Q4	43,40%	18,20%	22,10%	1,60%	1,90%	9,70%	12,80%
2020 Q1	52,50%	15,00%	16,90%	1,50%	2,20%	8,90%	11,90%
2020 Q2	56,90%	18,90%	11,70%	1,90%	2,10%	8,70%	8,50%

Операційна система Android має цілу низку переваг [13]:

По-перше, це повна відкритість системи. Розробка додатків доступна всім бажаючим, і тому Android є однією з найбільш легко розширюваних платформ.

По-друге, система повністю універсальна. Android OS можна встановлювати і на смартфони, і на кишенькові комп'ютери, і навіть на телефони (у полегшеному вигляді).

Це одна з найбільш зручних і практичних мобільних операційних систем, що використовуються в сучасному світі і її налаштування дуже просте.

Популярність операційної системи Android стала наслідком того, що провідні компанії мобільних телефонів, наприклад, HTC, Motorola, Samsung Electronics, і багато інших виробників стали використовувати Android на своїх смартфонах.

Перевагою Android від Google також є так само і те, що нова версія Android запускається майже кожні 3 місяці. Оновлення телефону новими версіями допомагає користувачам насолоджуватися більш якісними послугами.

Перед тим, як приступити до розробки програми для Android, необхідно ознайомитися з архітектурою системи та основними особливостями цієї платформи.

Система Android – це програмний стек для мобільних пристроїв, який включає операційну систему, програмне забезпечення проміжного шару (middleware), а також основні користувацькі додатки.

Архітектуру Android прийнято ділити на чотири рівні [1]:

- Рівень ядра;
- Рівень бібліотеки середовища виконання;
- Рівень каркаса додатків;
- Рівень додатків.

На рисунку 2.1 показані основні компоненти операційної системи та їх взаємодія між собою.

Кожна програма в ОС Android запускається у власному примірнику віртуальної машини Dalvik. Таким чином, всі працюючі процеси ізольовані від операційної системи і один від одного. І взагалі, архітектура Android Runtime така, що робота програм здійснюється строго в рамках оточення віртуальної машини. Завдяки цьому здійснюється захист ядра операційної системи від можливої шкоди з боку інших її складових. Тому код з

помилками не зможе зіпсувати Android і пристрій на його базі, коли вони спрацюють. Така захисна функція, разом з виконанням програмного коду, є однією з ключових для надбудови Android Runtime.



Рисунок 2.1 Архітектура операційної системи Android

Рівнем вище розташовується Application Framework, іноді званий рівнем каркаса додатків. Саме через каркаси додатків розробники отримують доступ до API, що надаються компонентами системи, що лежать нижче рівнем. Крім того, завдяки архітектурі фреймворку, будь-якого додатка надаються вже реалізовані можливості інших додатків, до яких дозволено одержувати доступ. У базовий набір сервісів і систем, що лежать в основі кожної програми і є частинами фреймворку, входять [2]:

- Багатий і розширюваний набір уявлень (Views), який може бути використаний для створення візуальних компонентів додатків, наприклад, списків, текстових полів, таблиць, кнопок або навіть вбудованого web-браузера.

- Контент-провайдери (Content Providers), керуючі даними, які одні програми відкривають для інших, щоб ті могли їх використовувати для своєї роботи.

- Менеджер Ресурсів (Resource Manager), що забезпечує доступ до ресурсів без функціональності (що не несуть коду), наприклад, до строковим даними, графіку, файлів та інших.

- Менеджер Оповіщень (Notification Manager), завдяки якому всі програми можуть відображати власні повідомлення для користувача в рядку стану.

- Менеджер Дій (Activity Manager), який управляє життєвими циклами додатків, зберігає дані про історію роботи з діями, а також надає систему навігації по них.

- Менеджер Місця розташування (Location Manager), що дозволяє додаткам періодично отримувати оновлені дані про поточний у географічному положенні пристрій.

Таким чином, завдяки Application Framework, додатки в ОС Android можуть отримувати в своє розпорядження допоміжний функціонал, завдяки чому реалізується принцип багаторазового використання компонентів додатків і операційної системи. Природно, в рамках політики безпеки.

На вершині програмного стека Android лежить рівень додатків (Applications). Сюди належить набір базових додатків, який встановлений на ОС Android. Наприклад, в нього входять браузер, поштовий клієнт, програма для відправки SMS, карти, календар, менеджер контактів та багато інших. Список інтегрованих програм може змінюватися в залежності від моделі пристрою та версії Android. І крім цього базового набору до рівня додатків відносяться в принципі всі програми під платформу Android, в тому числі і система, що розробляється.

Вибір пристроїв-комунікаторів в якості апаратної основи для реалізації проекту і Android OS в якості операційної системи значною мірою зумовив і

те, який інструментарій використовуватиметься для розробки і тестування додатків.

Система розроблялася на об'єктно-орієнтованій мові Java, розробленій компанією Sun Microsystems, яка в даний момент придбана корпорацією Oracle. Вибір даної мови був зроблений з низки міркувань.

Програми на мові Java транслюються в байт-код, що виконується віртуальною машиною (Java Virtual Mashine) - програмою, яка займається обробкою байт і передавальної інструкції обладнанню як інтерпретатор.

Гідність даного способу виконання програм полягає в повній незалежності байт-коду від операційної системи та обладнання, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. На комунікаторах з операційною системою Android таку функцію виконує віртуальна машина Dalvik.

Повний контроль виконання програми віртуальною машиною виливається в організацію гнучкої системи безпеки. Будь-які операції, які перевищують встановлені повноваження програми, викликають колізії при роботі з даними) викликають негайне переривання, яке може бути відповідно оброблено.

До недоліків концепції віртуальної машини можна віднести те, що виконання байт-коду віртуальною машиною в деякій мірі знижує продуктивність програм і алгоритмів, реалізованих на мові Java. Але і вплив цих особливостей зведено до мінімуму, завдяки застосуванню технології трансляції байт-коду в машинний код безпосередньо під час роботи програми (JIT-технологія) і широкому використанню переносного орієнтованого коду в стандартних бібліотеках.

Система Java створювалася об'єктно-орієнтованою з самого початку. Об'єктно-орієнтована парадигма в чому найбільш зручна при створенні програмного забезпечення.

Потрібно відзначити, що існує можливість розробляти програми і на C / C + + (за допомогою Native Development Kit), і на Basic (за допомогою

Simple) і з використанням інших мов. Також можна створювати власні програми за допомогою конструкторів додатків, таких як App Inventor.

Для розробки програм на мові Java необхідно спеціальне програмне забезпечення.

Java Development Kit (JDK) - безкоштовно розповсюджуваний корпорацією Oracle Corporation комплект розробника додатків мовою Java, що включає в себе стандартні бібліотеки класів Java, приклади, документацію, різні утиліти і виконавчу систему Java (JRE).

Варто відзначити такий важливий факт, що до складу JDK не входить інтегроване середовище розробки, яка встановлюється окремо.

Всі сучасні інтегровані середовища розробки додатків на Java, такі, як NetBeans IDE, Sun Java Studio Creator, IntelliJ IDEA, Borland JBuilder, Eclipse, спираються на сервіси, що надаються JDK. Більшість з них для компіляції Java-програм використовують компілятор з комплекту JDK. Тому ці середовища розробки або включають в комплект поставки одну з версій JDK, або вимагають для своєї роботи попередньої інсталяції JDK на машині розробника.

Незважаючи на те, що для розробки можна використовувати цілий ряд сучасних інтегрованих засобів розробки (IDE), список яких наведено в попередньому параграфі, вибір припав на Eclipse. Це відбулося з наступних причин:

Eclipse - найбільш повно документоване, вільне і доступне інтегроване середовище розробки для Java. Eclipse також дуже простий у вивченні. Це робить Eclipse дуже привабливим IDE для розробки додатків під Android;

Компанія Google випустила плагін для Eclipse - Android Development Tools, який дозволяє створювати Android-проекти, компілювати їх, і, що дуже важливо, використовувати емулятор мобільного пристрою для запуску та налагодження додатків.

Плагін Android Development Tools для Eclipse автоматично створює необхідну структуру Android проекту і встановлює необхідні параметри налаштування компілятора.

Android SDK включає в себе різноманітні інструменти, бібліотеки та документацію, які допомагають розробляти мобільні додатки для платформи Android. У їх число входять:

Емулятор Android – віртуальний мобільний пристрій, який запускається на звичайному комп'ютері. Емулятор використовується для проектування, налагодження і тестування програм в реальному середовищі виконання Android.

Dalvik Debug Monitor Service (DDMS) - інтегрований з Dalvik, стандартною віртуальною машиною платформи Android, цей інструмент дозволяє керувати процесами на емуляторі або пристрої, а також допомагає у налагодженні додатків.

Операційна система Android є інноваційною, яка з кожним роком набирає все більшу і більшу популярність. Популярна операційна система для телефонів і інших гаджетів створена на базі Linux, яка була і залишилася конкурентом Windows від компанії Microsoft. ОС Android – це відкрита продукція, для якої є в наявності велика кількість різноманітних програм в безмежному безкоштовному доступі. До того ж ОС безперервно розвивається і поліпшується.

Володарі і творці ОС Android це компанія Open Headset Alliance, де налічується близько 80 різних фірм, навіть Google. 1-е дітище з даною системою на борту було виставлено на загальний огляд на початку 2008 року. Після чого послідували пропозиції від виробників мобільної техніки. Зараз Андройд можливо зустріти не стільки на телефонах, але і на планшетних пристроях, а також і в фоторамках.

Основна частина програм для даної системи написана на мові Java. Завдяки опису операційної системи Android можна з'ясувати, як вона працює, які застосовуються двигуни і бібліотеки.



Починаючи з версії операційної системи під назвою Android 1.6 помітно перетворилася працездатність завантаження і покупки програм Android Market. Як і раніше, всі додатки поділяються на 2 масштабні групи Програми та Розваги, а далі на підкатегорії. У переліку підкатегорій показуються частоту завантаження контенту будь-якого з розділів. Найактуальніше нововведення - в опис програми або ж вставлені скріншоти, які дають можливість розцінити інтерфейс нового контенту. Це особливо необхідно при завантаженні комерційних програм, хоча крім того знадобиться і для приблизної оцінки дизайну нової заставки або гри, щоб не витратити трафік і час даремно. Загалом, для вітчизняних користувачів, як і раніше доступні лише безкоштовні завантаження. Крім того спрацьовує система оцінки. Будь-який користувач, що завантажив програму, зможе розцінити її і кинути текстове пояснення для інших бажаючих завантажити.

Заключною версією на даний момент вважається Android 4.0 - Ice Cream Sandwich. У ній - поліпшені шрифти і анімація, комфортний багатозадачний інтерфейс, вдосконалена ревізія орфографії і потокове визначення голосу.

Безсумнівно, чудовий телефон з даною сильною операційною системою – це маленький комп'ютер, готовий виконати найскладніші завдання повсякденної необхідності.

Спочатку мова JAVA називався Оак ( «Дуб») розроблявся Джеймсом Гослінгом для програмування побутових електронних пристроїв. Згодом він був перейменований в Java і став використовуватися для написання клієнтських додатків і серверного програмного забезпечення. Названий на честь марки кави Java, яка, в свою чергу, отримала найменування однойменного острова (Ява), тому на офіційній емблемі мови зображена чашка з гарячою кавою. Існує й інша версія походження назви мови, пов'язана з алюзією на каву-машину як приклад побутового пристрою, для програмування якого спочатку мова створювалася.

Java є об'єктно-орієнтованою мовою програмування, розроблена компанією Sun Microsystems (в подальшому придбаною компанією Oracle). Програми Java можуть працювати на будь-якій віртуальній Java-машині незалежно від комп'ютерної архітектури. Дата офіційного випуску - 23 травня 1995 року.

Мова Java активно використовується для створення мобільних додатків під операційну систему Android. Розробку додатків можна вести в середовищі Android Studio, NetBeans, в середовищі Eclipse, використовуючи при цьому плагін Android Development Tools (ADT) або в IntelliJ IDEA. Версія JDK при цьому повинна бути 5.0 або вище.

8 грудня 2014 року Android Studio визнана компанією Google офіційним середовищем розробки під ОС Android.

У якості середовища розробки Android програми був обраний програмний продукт Android Studio. Він є інтегрованим середовищем розробки (IDE) для роботи з платформою Android, анонсована 16 травня 2013 року на конференції Google I / O.

IDE перебувала у вільному доступі починаючи з версії 0.1, опублікованій в травні 2013, а потім перейшла в стадію бета-тестування, починаючи з версії 0.8, яка була випущена в червні 2014 року. Перша стабільна версія 1.0 була випущена в грудні 2014 року, тоді ж припинилася підтримка плагіна Android Development Tools (ADT) для Eclipse.

Android Studio, заснована на програмному забезпеченні IntelliJ IDEA від компанії JetBrains, офіційний засіб розробки Android додатків [3]. Дане середовище розробки доступно для Windows, OS X і Linux. Основні особливості – реалізована можливість верстки в реальному часі, є безліч варіантів розмірів і дозволів екранів. Присутній розділ довідки. Вбудовані інструменти поліпшення якості додатків. Легкість в написанні коду. Готові шаблони основних макетів і компонентів Android. Є інструменти для відстеження ефективності рекламних оголошень.

## 2.2 Інфологічна структура мобільної гри для вивчення основ тестування програмного забезпечення

Функціональні вимоги до проєктованого мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.

Функціональні вимоги визначають функціональність ПЗ, яку потрібно побудувати, щоб користувачі змогли виконати свої завдання. В ході аналізу були виявлені наступні вимоги:

1) додаток «Для вивчення основ тестування програмного забезпечення» має нести інформацію про історію програмування, мови програмування, розробників та тестування;

2) додаток «Для вивчення основ тестування програмного забезпечення» має надавати можливість брати участь у опитуванні;

3) додаток «Для вивчення основ тестування програмного забезпечення» має надавати можливість проходити тести;

4) додаток «Для вивчення основ тестування програмного забезпечення» має перевіряти відповіді на завдання;

Не функціональні вимоги до мобільного програмного комплексу для вивчення основ тестування програмного забезпечення

Не функціональні вимоги описують властивості і обмеження, що накладаються на систему. Для реалізації програми були виявлені наступні вимоги:

1) додаток «Для вивчення основ тестування програмного забезпечення» має бути написано на мові Java з використанням Android SDK;

2) додаток «Для вивчення основ тестування програмного забезпечення» має працювати на платформі Android версії 4.0 і вище;

3) сервер повинен бути написаний на мові Java з використанням фреймворка Spring Framework.

Діаграма варіантів використання

В ході аналізу було виділено основний актор системи - Користувач (рис. 2.2). Він може використовувати всі функції мобільного додатка, які представлені нижче:

- 1) Виконати вхід: користувач може здійснити вхід в додаток;
- 2) Переглянути корисну інформацію про видатних програмістів світу;
- 3) Пройти тестування, щоб виявити рівень засвоєного матеріалу;
- 4) Брати участь в опитуваннях, шляхом надання відповіді та доповнення інформації, що представлена у додатку.

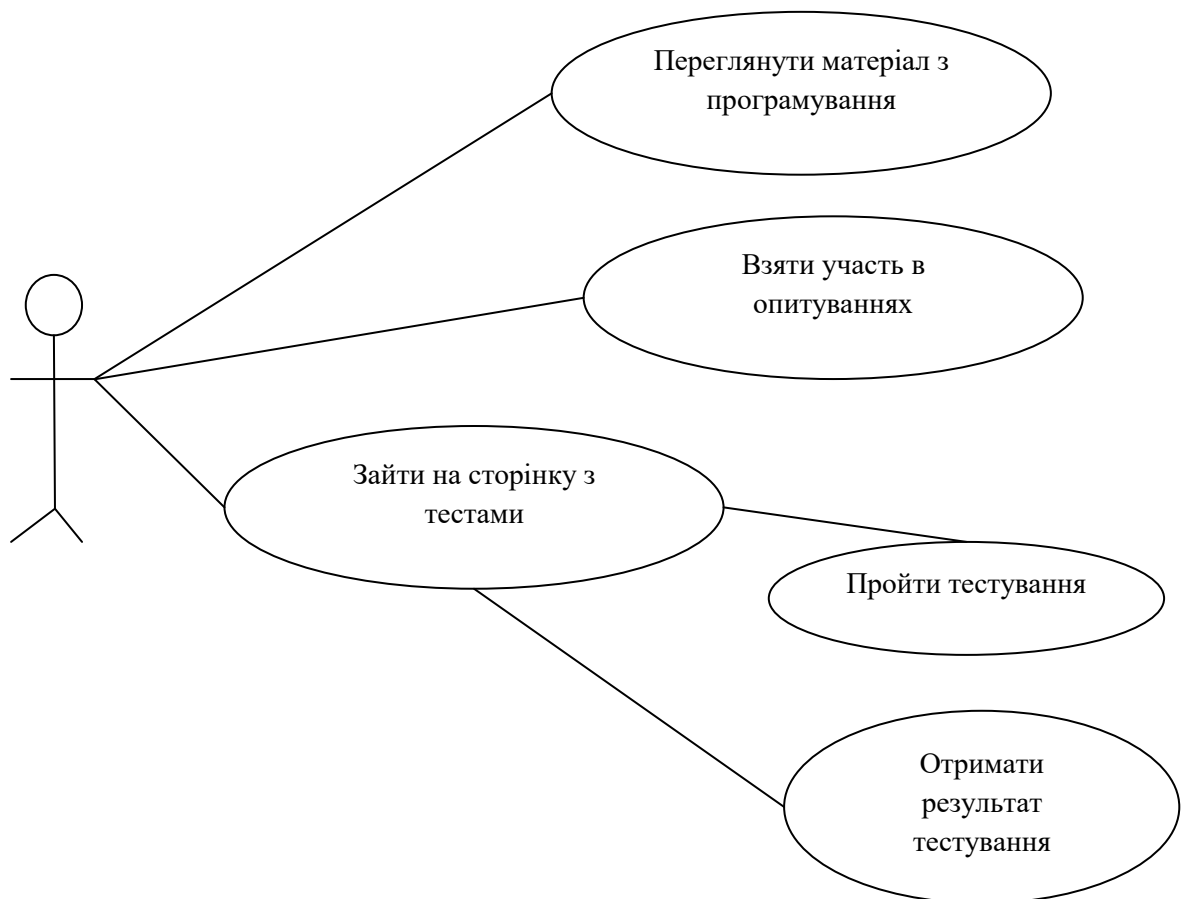


Рисунок 2.2 Діаграма використання

5) Виконати завдання: користувач може виконати з будь-якої сторінки, може завантажити та доповнити історію;

На рис. 2.3 представлена архітектура системи. Її основними компонентами є сервер і клієнтська програма.

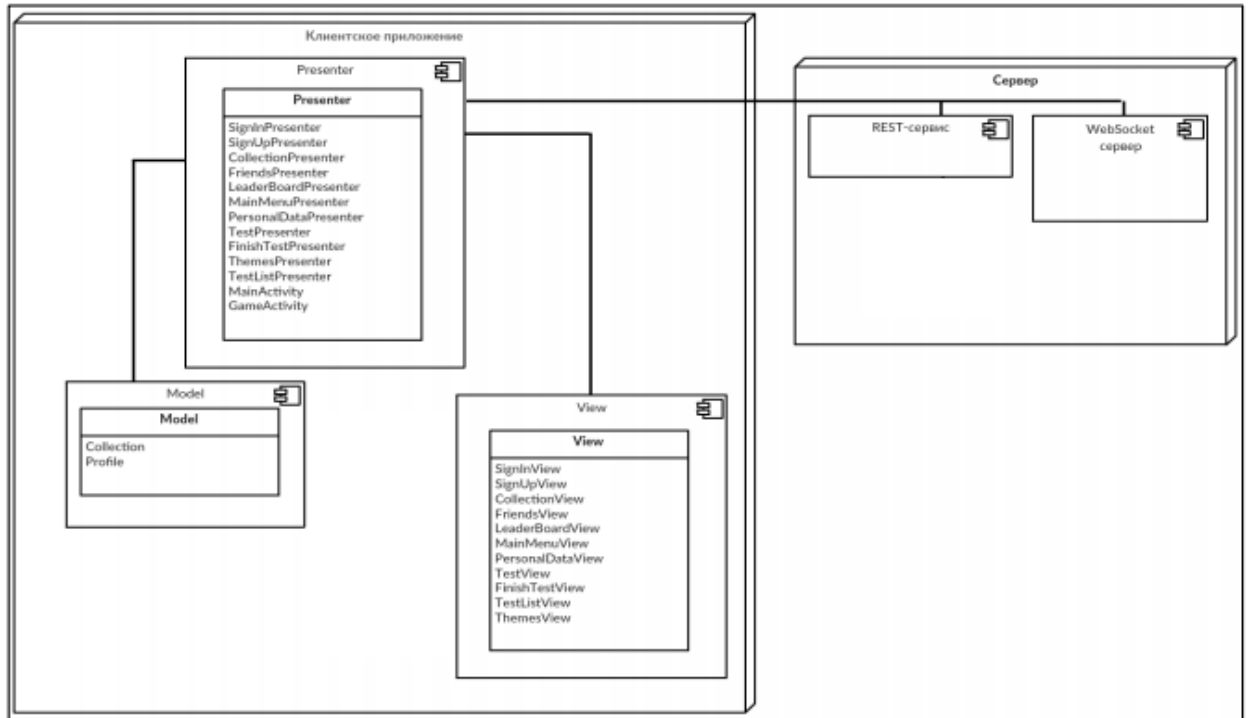


Рисунок 2.3 Архітектура системи «Для вивчення основ тестування програмного забезпечення»

Архітектура програми буде використовувати патерн проектування Model-View-Presenter [6] (рис. 2.3). Основна ідея даного патерну полягає в тому, щоб розділити модель даних програми, призначений для користувача інтерфейс і взаємодія з користувачем на три окремих компонента таким чином, щоб модифікація одного з компонентів надавала мінімальний вплив на інші. Даний патерн передбачає наявність 3 основних типів класів:

1) Model - класи, що відповідають за зберігання даних програми, а також методи роботи з цими даними;

2) View - класи, що відповідають за інтерфейс програми. Вони відображає дані, які передаються класами шару Presenter, а також реагують на дії користувача (наприклад, натискання кнопки), передаючи управління класу шару Presenter;

3) Presenter - класи, що забезпечують зв'язок між Model і View.

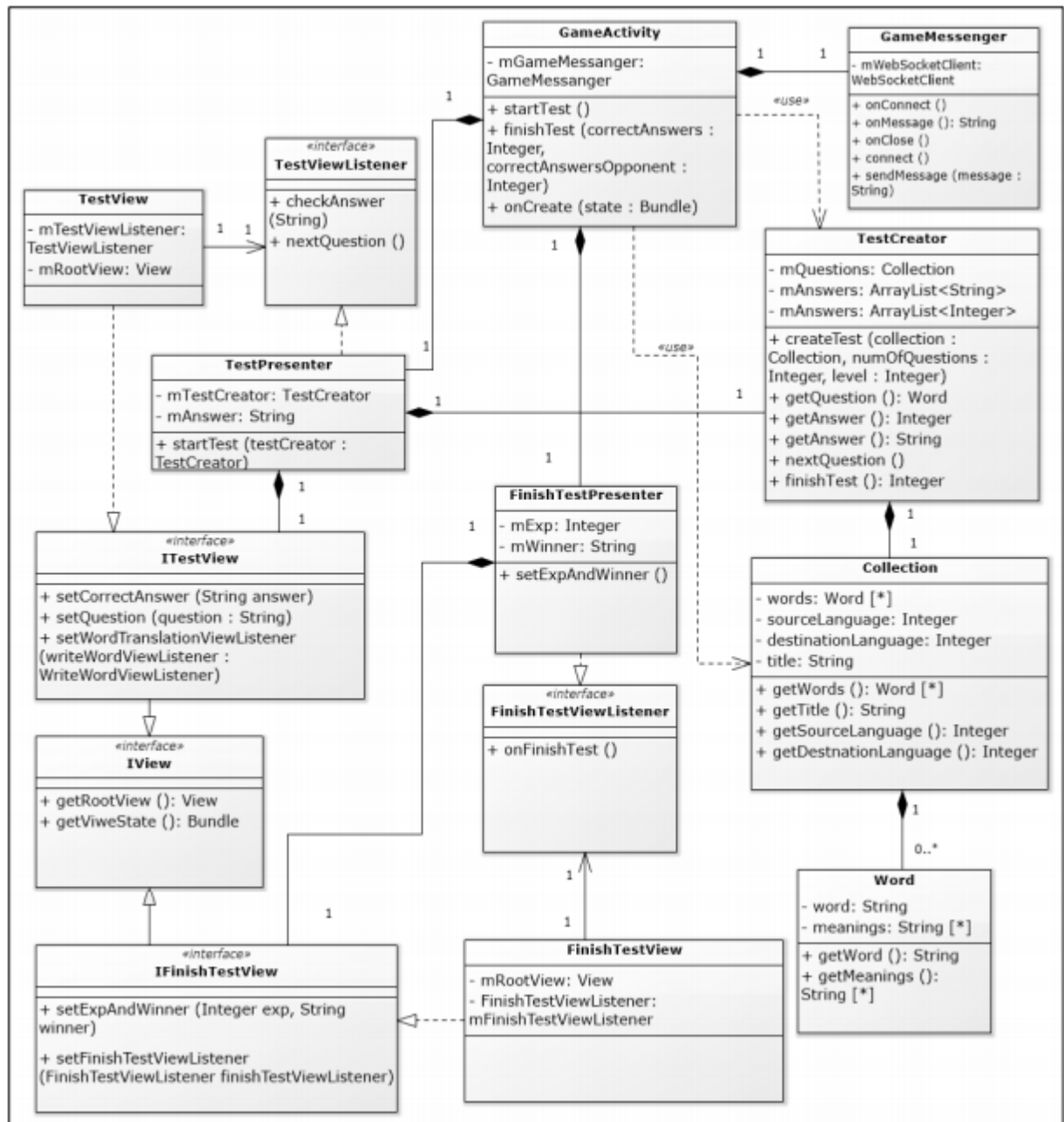


Рисунок 2.4 Діаграма класів режиму тестування

Реагують на зміни в Model, а також обробляють дії користувача, які їм передали класи шару View.

Окрема частина програми - режим тестування. Для нього створюється окрема активність GameActivity. У неї передаються дані, необхідні для створення тесту, ім'я та унікальний ідентифікатор того хто проходить тест, в ній же встановлюється з'єднання з Socket-сервером. Для відображення даних використовуються наступні класи: TestView і FinishTestView. Для обробки

користувальницьких дій використовуються класи: TestPresenter і FinishTestPresenter. Для створення тестів використовується клас TestCreator.

На рис. 2.4 зображена діаграма класів даного режиму.

REST-сервіс обробляє http-запити клієнта. Такі запити приходять на один з наступних класів (в залежності від адреси запиту): CollectionsController, RegistrationController, UserController і LeadersController. Вони отримують дані з http-запитів і передають їх класів CollectionsService і PersonService для подальшої обробки. CollectionsService і PersonService - сервіс класи для реалізації бізнес логіки. Вони використовують клас DAO <T> для доступу. Класи entity є даними, які зберігаються в проекті. На рис. 2.5 зображена діаграма класів REST-сервісу.

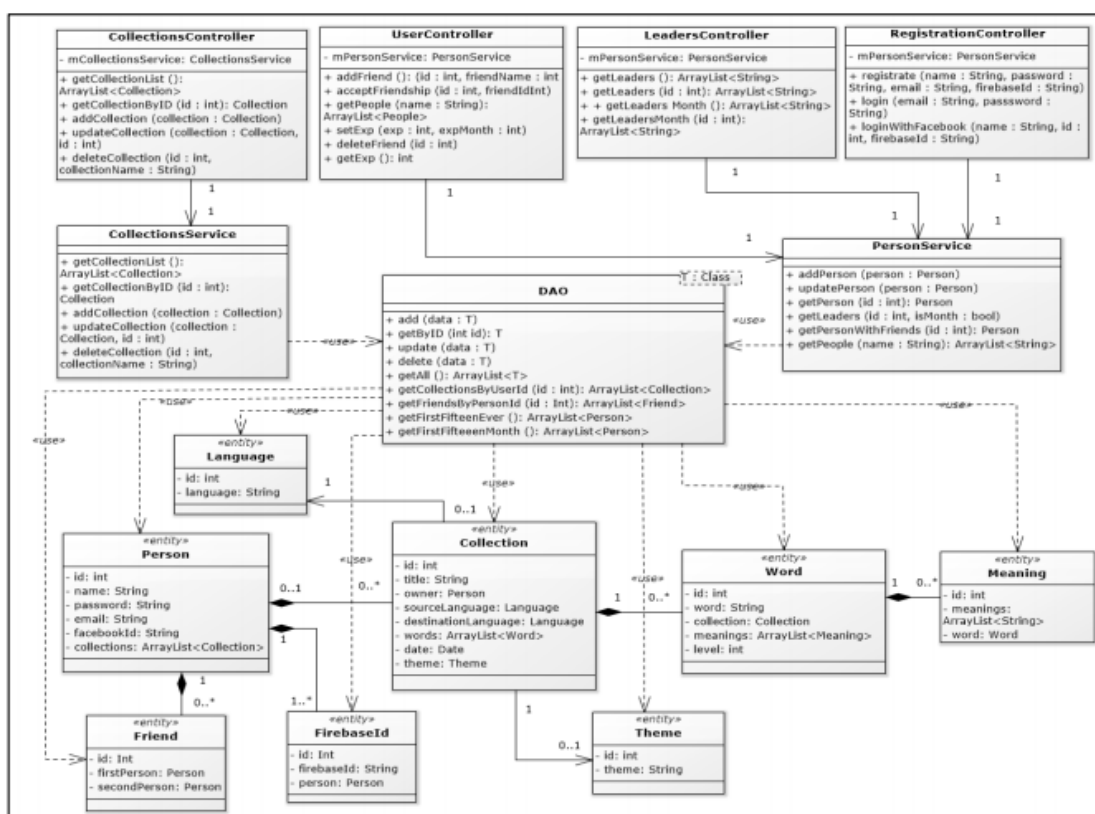


Рисунок 2.5 Діаграма класів REST-сервісу

## 2.3 Алгоритм роботи мобільної гри для вивчення основ тестування програмного забезпечення

Для реалізації мобільного програмного комплексу для вивчення основ тестування програмного забезпечення обрано каскадну модель життєвого циклу (рис. 2.6), яка реалізує, принцип одинарного виконання кожного з основних процесів і етапів в їх визначених рамках. Перехід на наступний етап реалізується після того, як буде проведена робота на поточному етапі, і відкатів на здійсненій стадії не передбачено. Кожен етап закінчується придбанням певного результату, який використовується в якості базової інформації для наступного етапу [16].



Рисунок 2.6 Етапи розробки мобільного програмного комплексу для вивчення основ тестування програмного забезпечення

Розробка мобільних додатків складається з наступних функціональних виробничих етапів:

1. Розробка технічного завдання.

Приступити до реалізації проекту неможливо без ідеї, яка повинна пройти попереднє тестування. Тільки так можна буде зрозуміти, наскільки



пропозиція відповідає споживчому попиту. Перш ніж приступити до розробки необхідно визначити обсяг цільової аудиторії і на підставі аналітичних робіт починати розробляти технічний план, який допоможе реалізувати проект якісним і функціональним.

## 2. Дизайн

Під кожний індивідуальний додаток необхідно придумати і розробити оригінальний дизайн. Зовнішнє і внутрішнє оформлення допомагає формувати довіру, лояльне ставлення і тим самим робить продукт більш успішним і конкурентоспроможним. Якісний дизайн надає унікальності додатку, тому актуальніше за все всі графічні і візуальні образи створювати «з нуля».

## 3. Створення копії або прототипу.

Така перевірна робота дозволить наочно переконатися в тому, що додаток оснащено зручним і зрозумілим інтерфейсом, що він має всі якісні функціональні показники і працездатність ключових параметрів програми. Для програмування краще задіяти перевірені і сучасні технологічні рішення

## 4. Тестування готового продукту.

Експертна, перевірна оцінка дозволить виключити всі недоліки і при необхідності провести деякі коригувальні процеси. Надалі залишається тільки знайти якісні і авторитетні магазини додатків. Публікація додатків на тематичних ресурсах дозволить ефективно розкручувати свій продукт, залучаючи максимальну кількість зацікавлених користувачів.

Загальна структура клієнт-серверної взаємодії з боку сервера представлена на рис. 2.7.

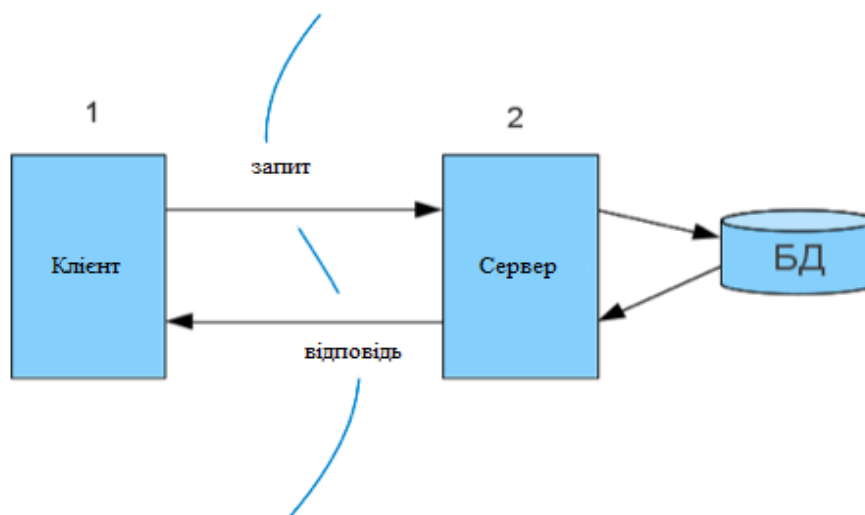


Рисунок 2.7 Загальна структура клієнт-серверної взаємодії з боку сервера

Однак, у рамках роботи розглянемо цей же погляд з боку клієнта, і в зв'язку з цим, немає ніякої різниці між двуланковою і три ланковою архітектурою.

Може бути безліч клієнтів, які використовують один акаунт для спілкування з сервером.

Кожен клієнт, як правило, має своє власне локальне сховище. Таким чином, у ряді випадків, локальне сховище може бути синхронізоване з хмарою, і, відповідно, з кожним із клієнтів. Оскільки це окремий випадок і не впливає на архітектуру програми, опускаємо його [18].

Слід зазначити, що оскільки, деякі розробники прагнуть позбутися «серверної частини» деякі додатки побудовані навколо синхронізації їх сховищ в «хмарі». Тобто, фактично, мають так само, дволанкову систему, але з перенесенням архітектури її розгортання на рівень операційної системи. У деяких випадках така структура виправдана, але така система не так легко масштабується, і її можливості дуже обмежені.

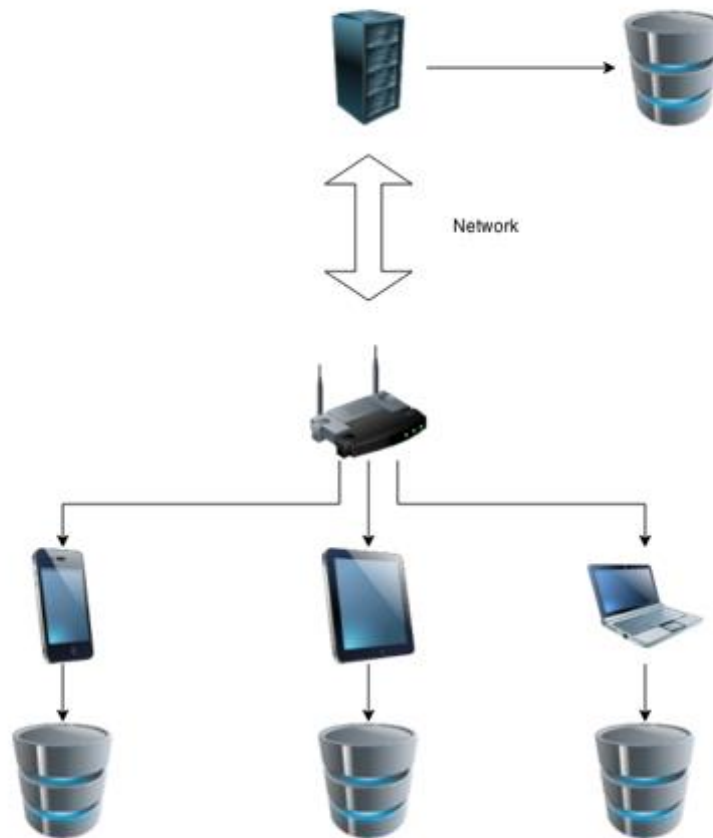


Рисунок 2.8 Загальна структура програми

На самому примітивному рівні мобільний програмний комплекс для вивчення основ тестування програмного забезпечення складається з наступних архітектурних шарів:

Ядро додатку, яке включає в себе компоненти системи, не доступні для взаємодії з користувачем;

Графічний інтерфейс користувача;

Компоненти повторного використання: бібліотеки, візуальні компоненти й інше;

Файли оточення: AppDelegate, .plist і т. д.;

Ресурси програми: графічні файли, звуки, необхідні бінарні файли.

## РОЗДІЛ 3.

### РОЗРОБКА МОБІЛЬНОЇ ГРИ ДЛЯ ВИВЧЕННЯ ОСНОВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Формування мобільної гри для вивчення основ тестування програмного забезпечення

Програми для Android складаються з компонентів, які система може запускати і керувати так, як їй необхідно. Для цього система повинна бути в змозі запустити процес для програми в якому знаходяться необхідні компоненти, і ініціалізувати потрібні їй об'єкти. Одним з компонентів Android - додатку є діяльність ( activity ).

Activity являє собою візуальний інтерфейс (окремий екран) для однієї дії, яку користувач може зробити.

Додаток може складатися з одного activity або з декількох. Це залежить від типу додатка і його дизайну. Одне activity може викликати інше.

Кожне activity задає вікно для відображення, яке, зазвичай, займає весь екран, але може бути менше і плавати поверх інших вікон. Activity може використовувати додаткові вікна, наприклад, спливаючий діалог, який вимагає проміжної відповіді користувача, або вікно, яке відображає користувачам важливу інформацію при виборі елемента, вартого особливої уваги.

Візуальний інтерфейс будується на основі ієрархії візуальних компонентів – об'єктів, похідних від базового класу View. Android має ряд готових до використання компонентів, включаючи кнопки, текстові поля, смуги прокрутки, меню, прапорці і багато іншого.

Перш ніж створювати додатки на мові Java, необхідно встановити інтегровану середу, JDK, JSDK, бібліотеки і зробити необхідні настройки.

Для роботи необхідно виконати наступні кроки:

- Завантажити і про інсталювати j2sdk1.5.0 (або іншу версію за адресою:

<http://www.oracle.com/technetwork/java/archive-139210.html>).



Рисунок 3.1 Налаштування середовища розробки

- Клацнути по ехе-файлу.
- Перезавантажити комп'ютер.
- Прописати наступні змінні середовища. (Пуск> мій комп'ютер-> права кнопка миші-> властивості, вкладка: «Додатково - змінні середовища (оточення)»):

JAVA\_HOME (рис. 3.2):

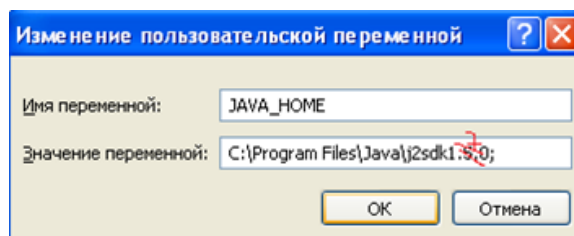


Рисунок 3.2 Змінна середовища JAVA\_HOME

CLASSPATH (рис. 3.3):

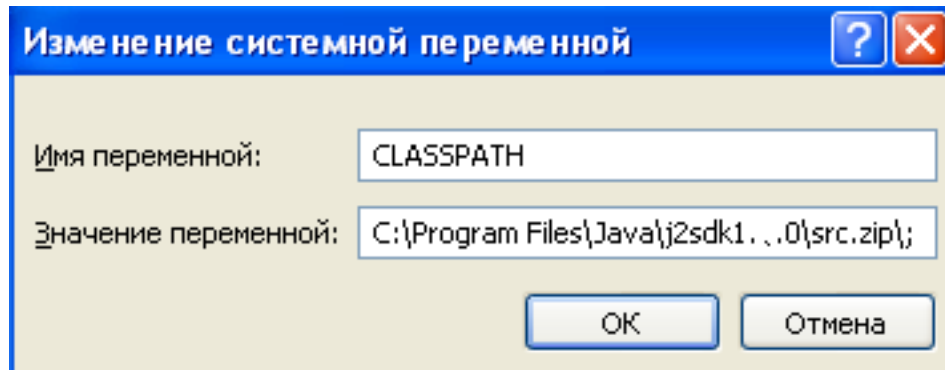


Рисунок 3.3 Змінна середовища CLASSPATH

відповідно до того файлового шляху, куди були розпаковані архіви.

- Створити папку в тому місці, де будуть зберігатися проекти (наприклад, в корені, на диску "D", під назвою 'MyWebProjects') (рис. 3.4):

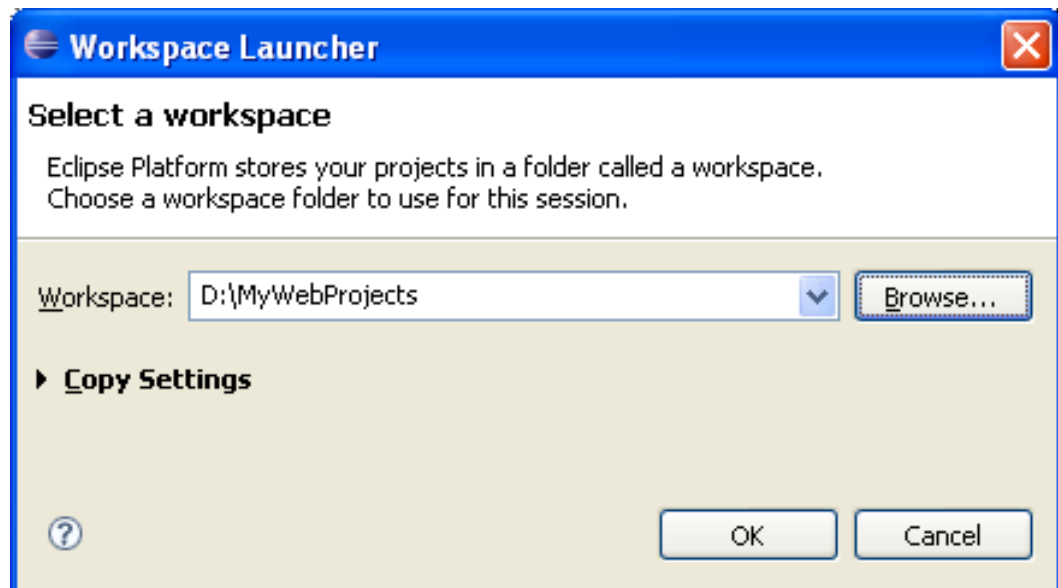


Рисунок 3.4 Вибір Workspace

- Запустити Eclipse, клікнувши по файлу з ярликом Екліпса - синім кружком. Надалі є можливість, наприклад, закріпити його в меню «пуск». Як Workspace у вікні за запитом потрібно вибрати заготовлену папку.

- Через деякий час з'явиться Welcome-сторінка Eclipse з ярликами (рис. 3.5):



Рисунок 3.5 Welcome-сторінка Eclipse

- Потрібно клацнути лівою кнопкою миші на ярлику 'Workbench', і відкриється перспектива 'Resource', яку Eclipse відображає за замовчуванням (рис. 3.6):

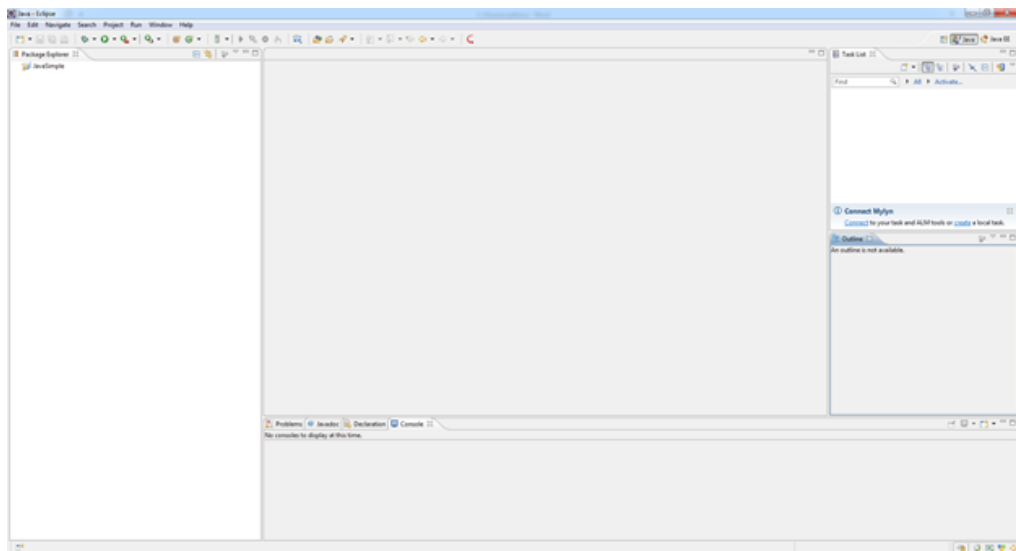


Рисунок 3.6 Перспектива 'Resource'

- Для того щоб створювати Java-додатки, потрібно створити проект. Для цього натисніть на вкладку File-> New-> Java Project і дайте ім'я проекту, наприклад, 'JavaSimple' або 'MySimpleProject' (рис. 3.7):

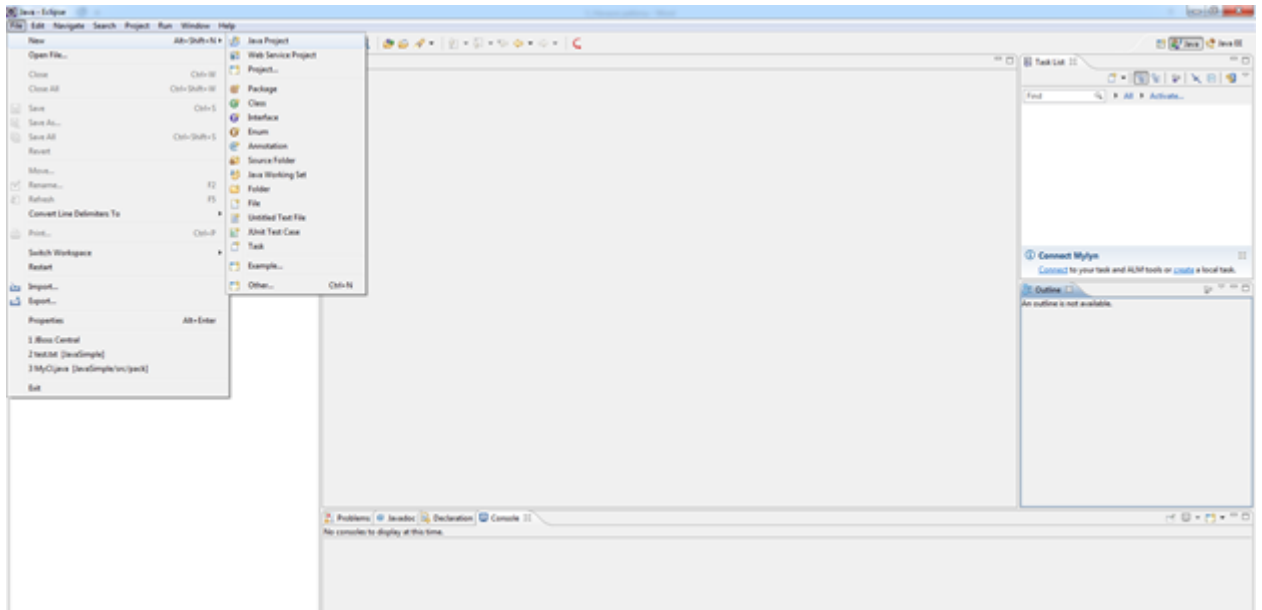


Рисунок 3.7 Створення проекту

Тепер знайдіть папку `src` (source) і створіть свій пакет для додатків. Дайте йому назву, наприклад, 'pack'.

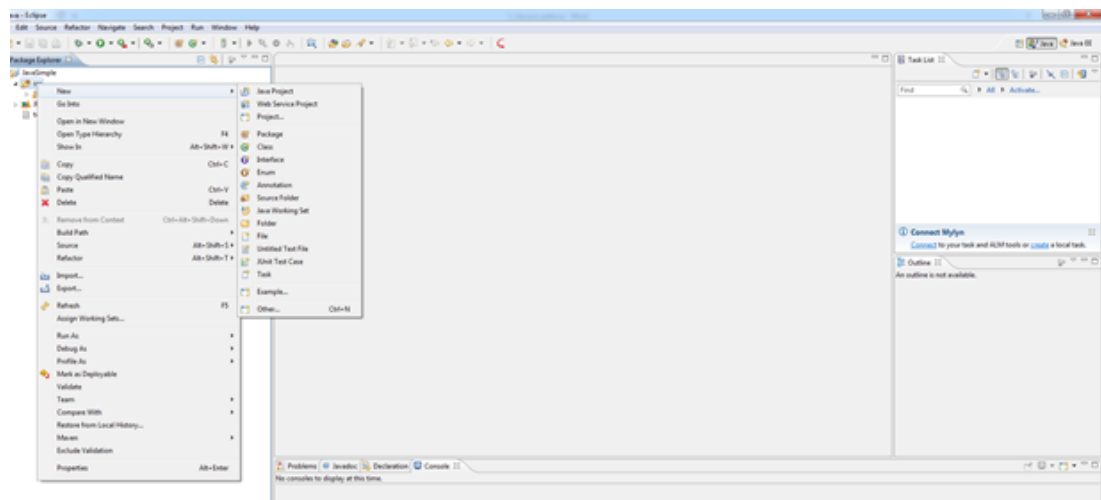


Рисунок 3.8 Створення проекту

Створіть тепер перший клас (права клавіша миші по пакету - вибираємо `New / Class`).



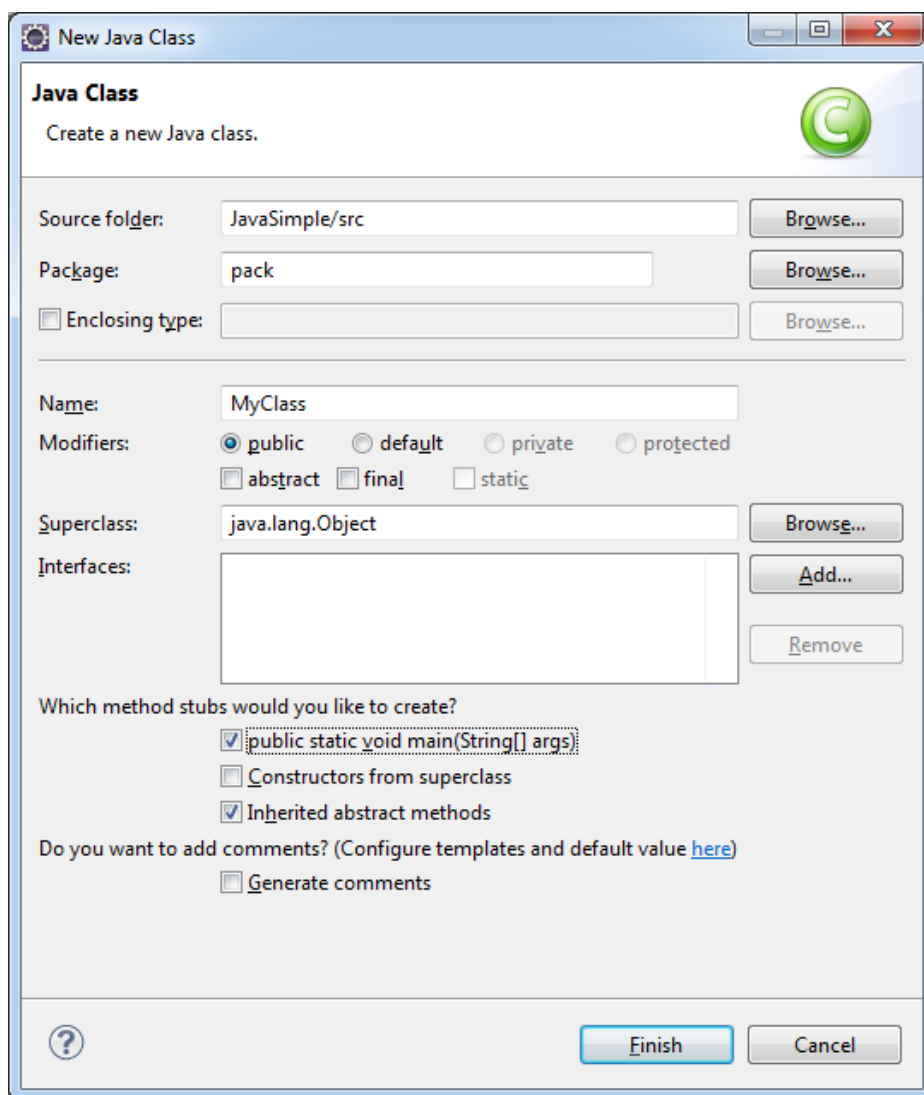


Рисунок 3.9 Створення класу

Натиснимо Finish і отримаємо заготовку під клас.

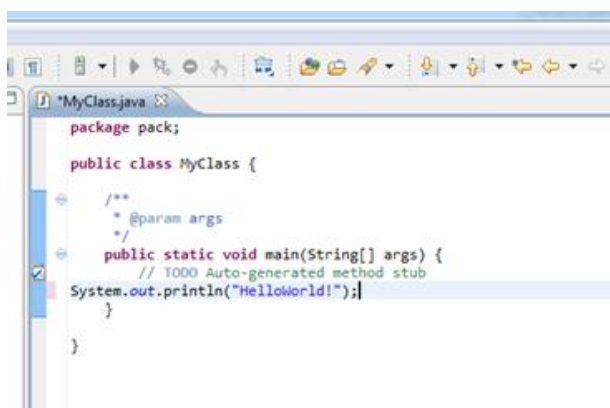


Рисунок 3.10 Формування додатку

Натисніть на зелений трикутник в горизонтальній верхній панелі:

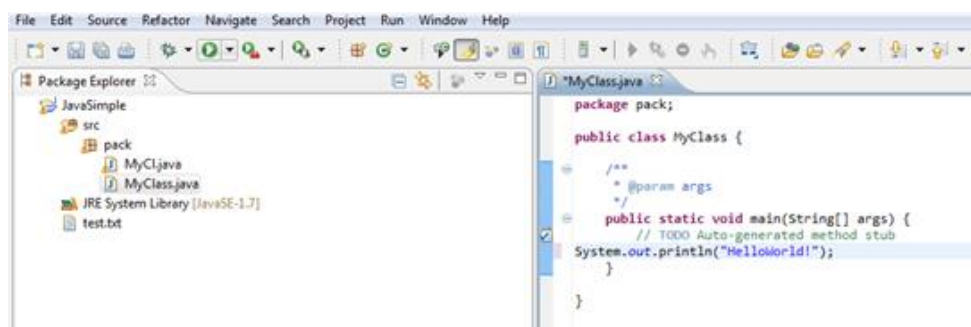


Рисунок 3.11 Формування додатку

Виберіть Java Application:

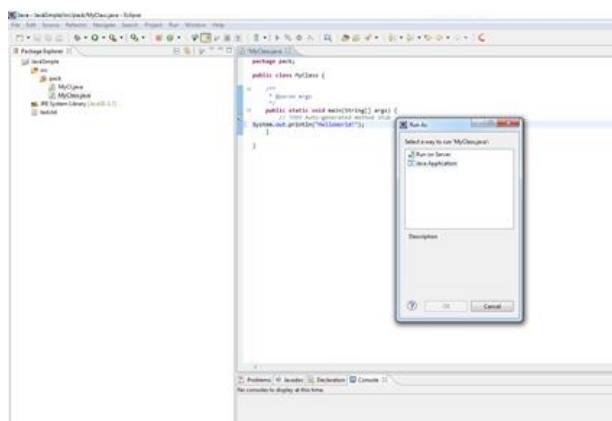


Рисунок 3.12 Формування додатку

І внизу в консолі відіб'ється ваш текст:

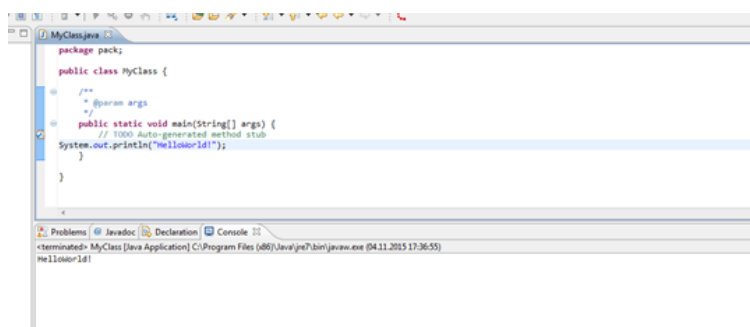
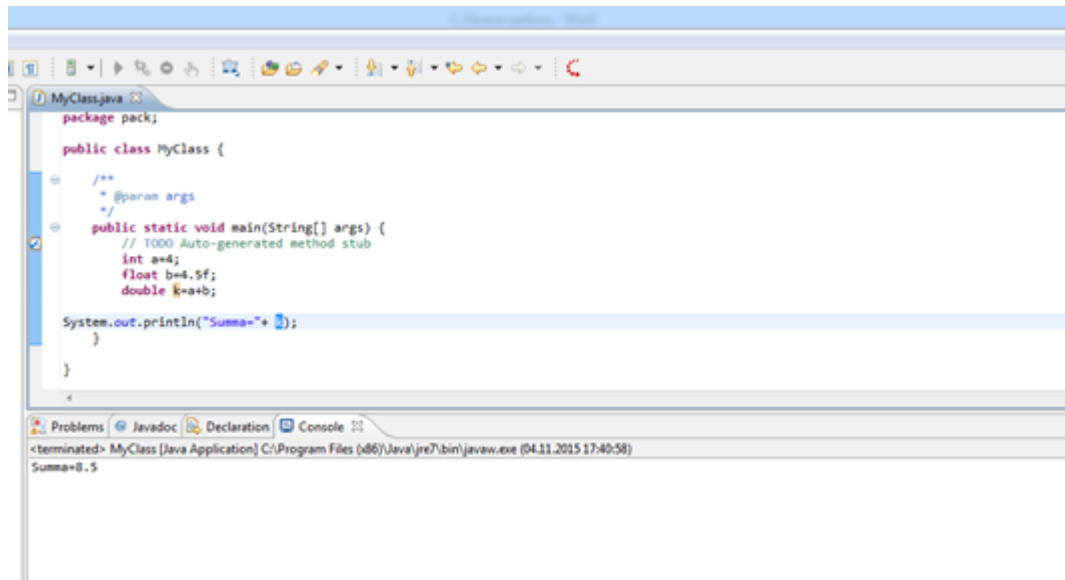


Рисунок 3.13 Формування додатку

Протестуйте його (наприклад, виведіть напис або підрахуйте арифметичний вираз).



```

package pack;

public class MyClass {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int a=4;
        float b=4.5f;
        double k=a+b;

        System.out.println("Summa="+ k);
    }
}

```

Problems Javadoc Declaration Console

<terminated> MyClass [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (04.11.2015 17:40:58)

Summa=8.5

Рисунок 3.14 Формування додатку

Здійснюємо проектування всіх основних компонентів мобільного додатку для вивчення основ тестування програмного забезпечення.

Структура проекту являє собою дерево каталогів. При компіляції в каталозі проекту створюється папка bin/. Відкомпільований код Java-класів разом з файлами даних і ресурсів поміщається в архівний файл з розширенням .Ark. Цей файл використовується для розповсюдження програми та встановлення його на мобільних пристроях.

#### Каталог ресурсів

У даному каталозі зберігаються використовувані в додатку статичні файли ресурсів: зображення, рядки, файли розмітки і т.д.

У підкаталозі res/drawable/ розміщені всі графічні файли, використовувані в додатку. Наприклад: файл icon.png – значок програми, який буде відображатися в меню запуску встановлених на телефоні додатків.

У підкаталозі `res /values/` зберігаються загальні константи для усієї програми. Наприклад:

```
<string name="menu_info">Info</string>
<string name="search_name">Name</string>
```

Підкаталог `res / layout /` містить файли розмітки у форматі XML, які визначають зовнішній вигляд вікна діяльності і розташування на ньому елементів управління. Перевага оголошення користувача інтерфейсу в XML - файлі полягає в тому, що це дає можливість відокремити структуру інтерфейсу додатку від програмного коду, який керує поведінкою програми.

Додатково, оголошення розмітки в XML – файлі полегшує можливість по візуалізації структури користувальницького інтерфейсу, що спрощує налагодження програми.

Розглянемо структуру файлу розмітки на прикладі файлу `objects.xml`.

Файл розмітки повинен містити тільки один елемент, який повинен бути об'єктом `View` або `ViewGroup`.

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
```

Після визначення кореневого елемента, додаються додаткові об'єкти, тим самим формується ієрархія елементів, яку визначає створювана розмітка.

```
<ListView
    android:layout_height="fill_parent"
    android:id="@android:id/list"
```

```

android:layout_width="fill_parent"
android:drawSelectorOnTop="false"/>

```

Елемент `<ListView>` являє собою вертикальний список з прокруткою. Пов'язані зі списком дані `ListView` отримує від об'єкта `ListAdater`.

#### Структура файлу `R.java`

Коли проект компілюється в перший раз, середа створює клас `R` і поміщає його в файл `R.java`. Цей клас використовується в кодї програми для звернення до ресурсів, які знаходяться в каталозі `res/`.

Клас `R` містить набір внутрішніх класів з ідентифікаторами ресурсів, які він створює в залежності від внутрішнього вмісту каталогу `res /:` `drawable`, `layout`, `id`, `string`.

Наприклад:

```

public static final class drawable {
    public static final int androidmarker=0x7f020000;
    public static final int h_hot=0x7f020001;
    ...
public static final class id {
    public static final int ObjectRess=0x7f07000b;
    public static final int ObjectName=0x7f07000a;
    public static final int ress=0x7f070006;
    public static final int button_add=0x7f070007;
    ...

```

#### Структура файлу `AndroidManifest.xml`

Маніфест – структурний XML - файл, що задає конфігурацію програми. Він оголошує компоненти програми, перераховує будь-які бібліотеки, пов'язані з додатком, а також оголошує дозволи, які додаток надає.

Файл маніфесту інкапсулює всю структуру `Android`-додатку, його функціональні можливості і конфігурацію.

Структура маніфесту багаторівнева. Порядок розташування елементів на одному рівні довільний. Елемент `<application>` є основним елементом маніфесту і містить безліч дочірніх елементів, що визначають структуру і роботу програми. Елементи `<manifest>`, `<application>` і `<uses-sdk>` є обов'язковими.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.tost"                                android:versionCode="1"
    android:versionName="1.0.">
```

```
    <uses-sdk android:minSdkVersion="8" />
```

```
    <application                                android:icon="@drawable/icon"
    android:label="@string/app_name">
```

Елемент `<application>` містить опис компонентів додатка, доступних у пакеті. У даного елемента є дочірні елементи, які оголошують кожен з компонентів, що входять до складу програми.

```
    <provider android:name=".providers.TemplesProvider"
        android:authorities="com.TemplesProvider">
    </provider>
```

...

```
    <uses-library android:name="com.google.android " />
```

```
        <activity android:name=".activities.MainActivity"
            android:label="Tost MainMenu"
            android:launchMode="singleTask">
            <intent-filter>
```

```

        <action
android:name="android.intent.action.MAIN" />
        <category
android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
...

```

Елемент `<uses-permission>` запрошує дозволи, які додатком повинні бути надані системою для його нормального функціонування.

```
<uses-permission android:name="android.permission.INTERNET" />
```

### Файли діяльностей

Файл діяльності являє собою клас, що містить код який реалізує логіку роботи програми в даному вікні. У кожному з таких класів обов'язково повинен визначатися метод зворотного виклику `onCreate`. Цей метод викликається при створенні діяльності. У середині методу налаштовують статичний інтерфейс діяльності – створюють уявлення, пов'язують дані зі списками і т.д. Цей метод приймає один параметр – об'єкт `Bundle`, що містить попередній стан діяльності (якщо цей стан збережено)

У файлі `MainActivity.java` задається функціональність головного вікна програми: визначаються дії, які відбуваються після натискання кнопок, а саме запуск, що активують відповідні кнопки.

```

final Button button = (Button) findViewById (R.id.button);
    button.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

```

```

Intent Activity = new Intent(MainActivity.this,
Activity.class);

startActivity(Activity);

}

});

```

- Процедура зчитування даних з текстового файлу `ReadInTxt (String in)`.
- Створення спливаючого меню в процедурі `onCreateOptionsMenu (Menu menu)`. Створення обробника подій вибору пункту меню `onOptionsItemSelected (MenuItem item)`
  - Створення спливаючого діалогу побудови курсу лікування.
  - Процедура додавання групи об'єктів у курс `addGroup (Uri uri, Drawable drawable)`
  - Процедури виклику діалогів побудови курсу `ConstructRouteDialog ()`, `CollectRouteDialog ()`, `AddTostRouteDialog ()`

У файлі `ObjectsEditorActivity.java` описується функціональність вікна POI:

- Створюється список груп об'єктів
- Створення обробника подій вибору елементів списку (груп об'єктів), а саме переходи на нові списки об'єктів даної групи.

Розглянемо структуру даних файлів на прикладі `ObjectDbHelper.java`. Клас `ObjectDbHelper`, що розширює клас `SQLiteOpenHelper` представляє таблицю `Object`. У ньому оголошена структура таблиці і в методі `onCreate ()` проводиться створення таблиці і заповнення її текстовими даними при першому запуску програми на пристрої.

```
public void onCreate{
```



```

db.execSQL("CREATE TABLE " + TABLE_NAME + " (_id
INTEGER PRIMARY KEY AUTOINCREMENT, "+ NAME + " TEXT, " +
ADDRESS + " TEXT, " + LAT + " TEXT, " + LON + " TEXT);");
ContentValues values = new ContentValues();
}

```

### Провайдери

Розглянемо структуру провайдера на прикладі файлу класу провайдера `ObjectViewProvider.java`.

Клас `ObjectViewProvider` даного файлу розширює базовий клас `ContentProvider` представляє логіку доступу до вмісту даних `Object`.

## 3.2 Тестування мобільної гри для вивчення основ тестування програмного забезпечення

Проведемо тестування мобільного програмного комплексу для вивчення основ тестування програмного забезпечення. Завантажимо додаток на мобільний телефон на полі з'явиться знак програми. Натискаємо на знак програми та очікуємо завантаження. На екрані з'явиться головне вікно програми (рис. 3.15).

Нижня панель активних кнопок 

має чотири пункти:

- програмісти;
- парадигми;
- мови;
- тестування.

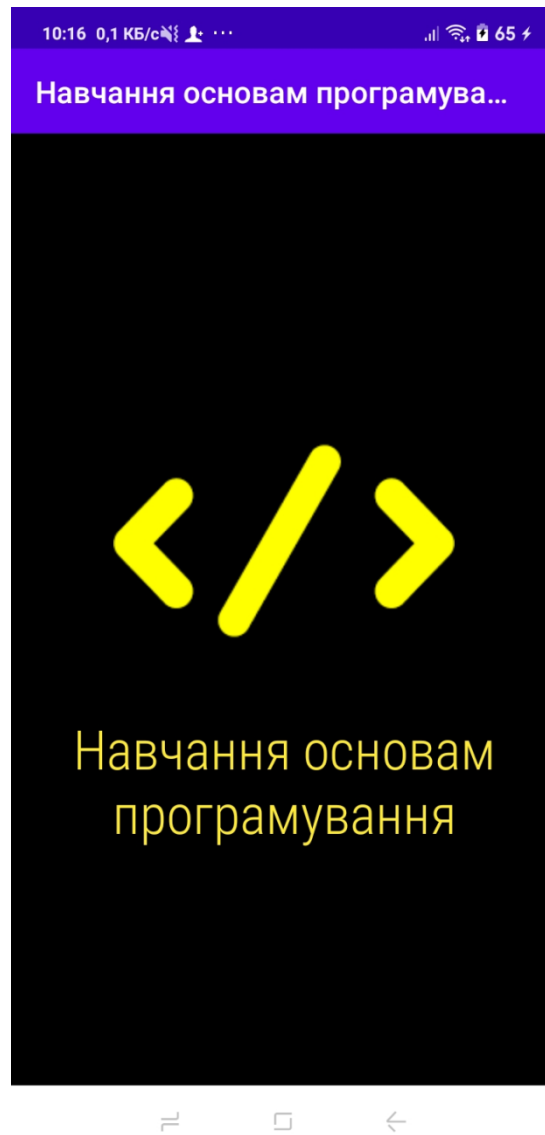


Рисунок 3.15 Головне вікно програми

Вкладення програмісти розповідають про видатних програмістів наведено на рис. 3.16.-3.19.

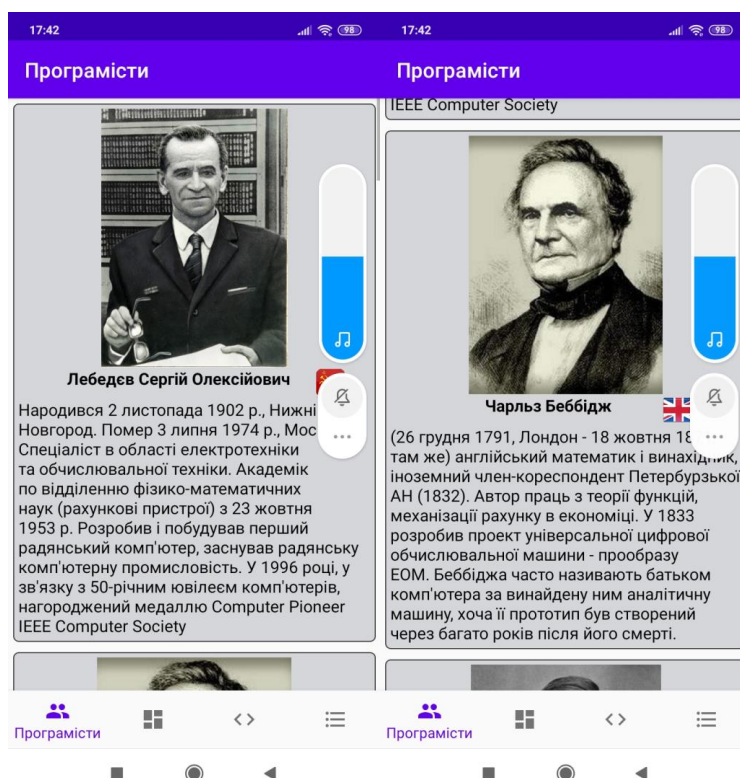


Рисунок 3.16 Вкладення видатні програмісти

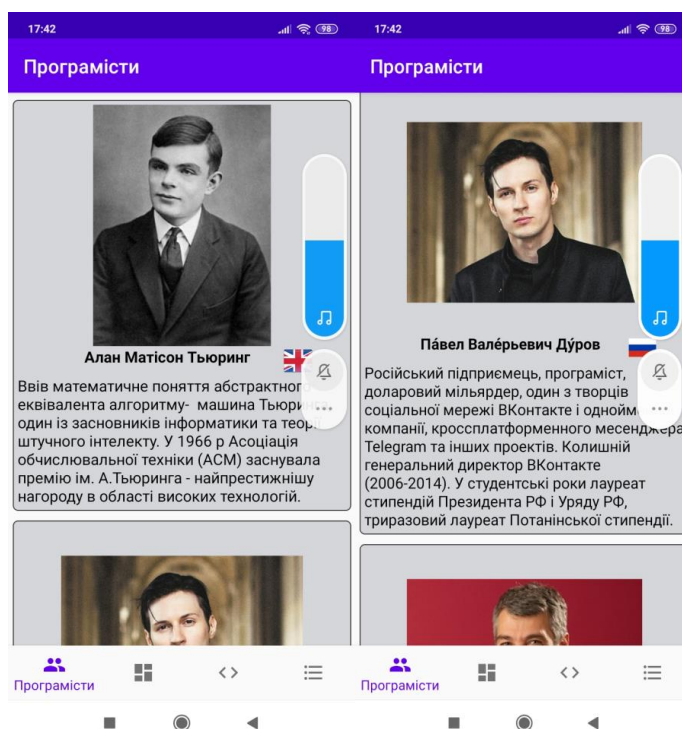


Рисунок 3.17 Вкладення видатні програмісти

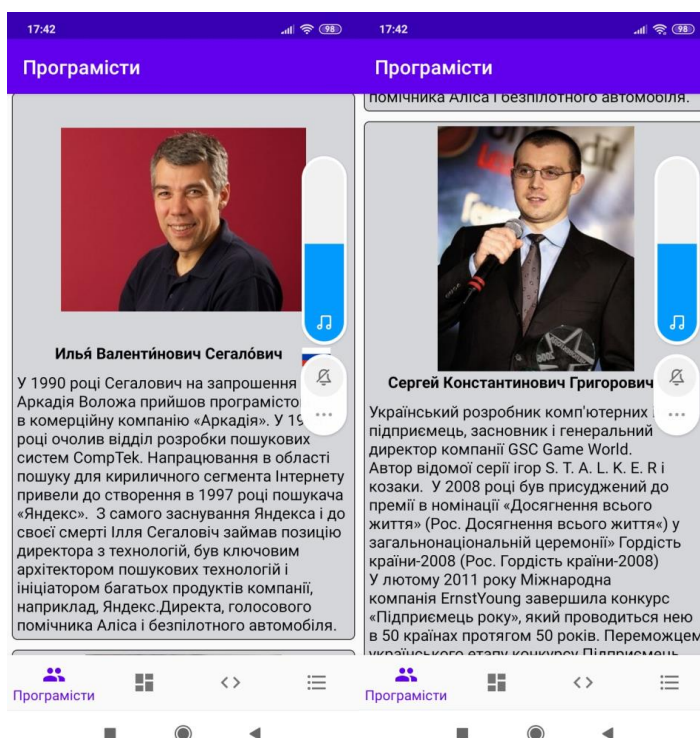


Рисунок 3.18 Вкладення видатні програмісти

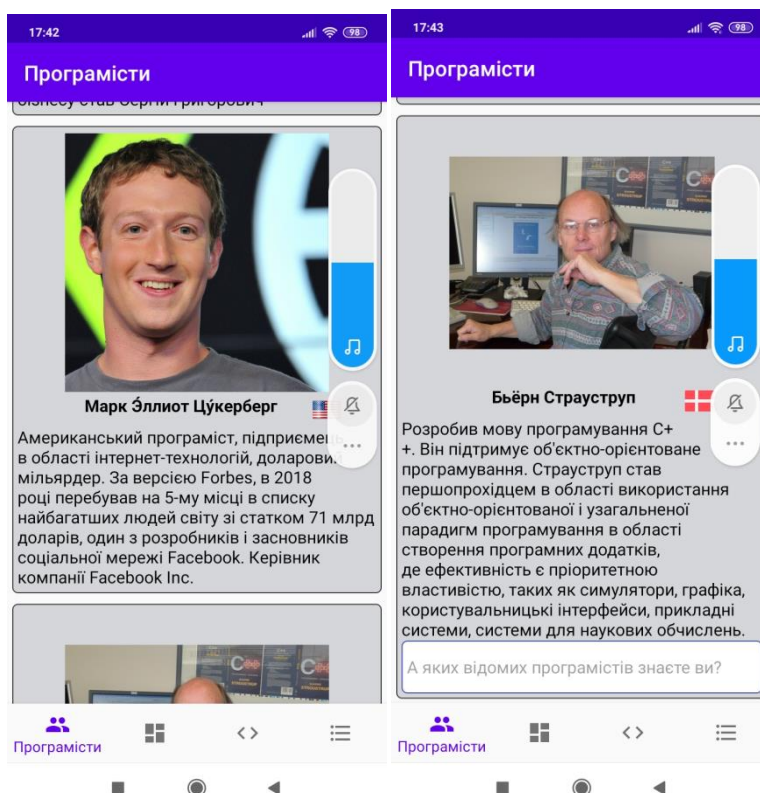


Рисунок 3.19 Вкладення видатні програмісти

Наступним вкладенням є парадигми (Рис. 3.20).

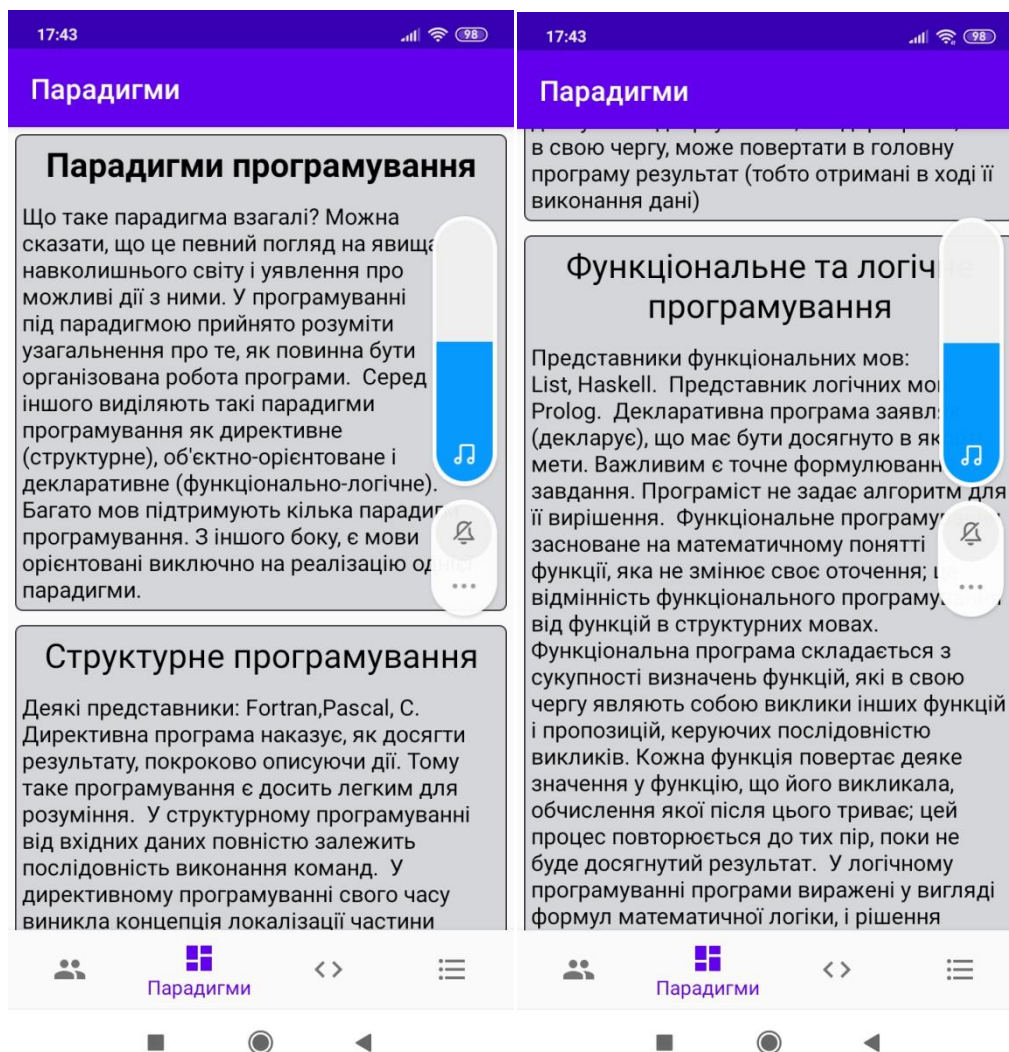


Рисунок 3.20 Вкладення парадигми

Наступним вкладенням є мови програмування. Розглянемо детально вкладення «Мови програмування» результати наведемо на рис. 3.21-3.22.

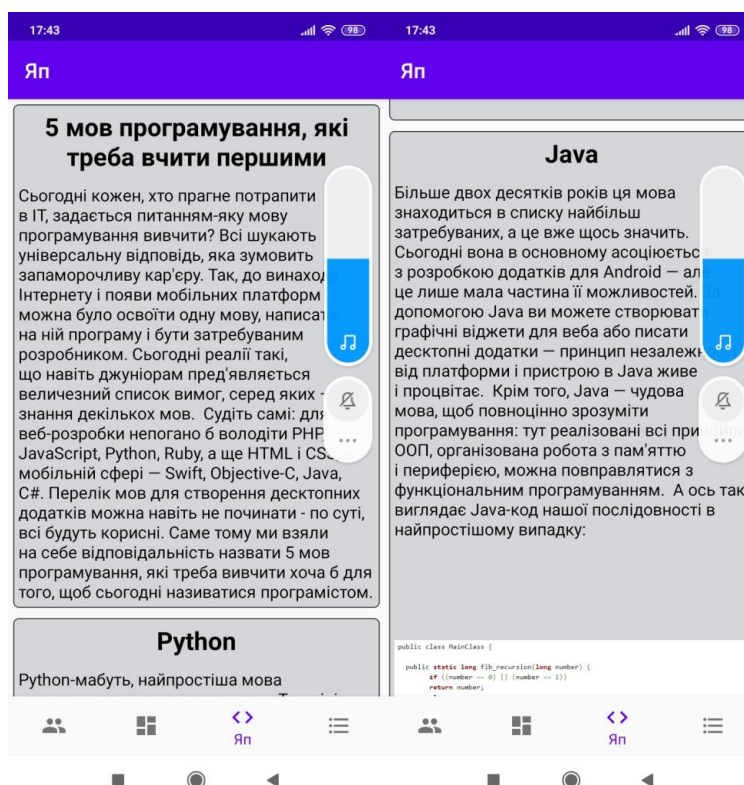


Рисунок 3.21 Вкладення мови програмування

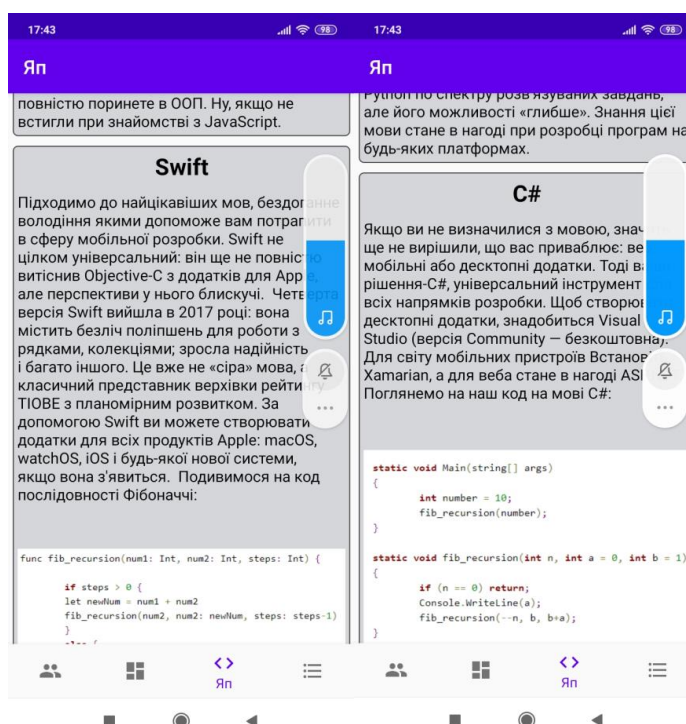


Рисунок 3.22 Вкладення мови програмування

Четверте кладення – це тестування. Пройдемо тестування та результати наведемо на рис. 3.23-3.25.

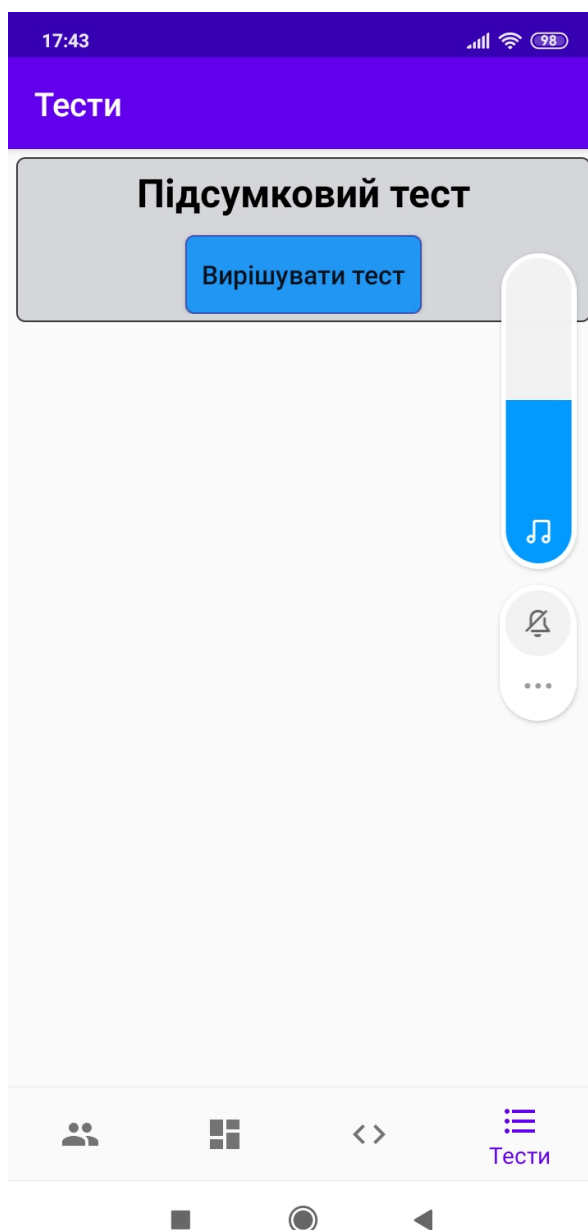


Рисунок 3.23 Тестування

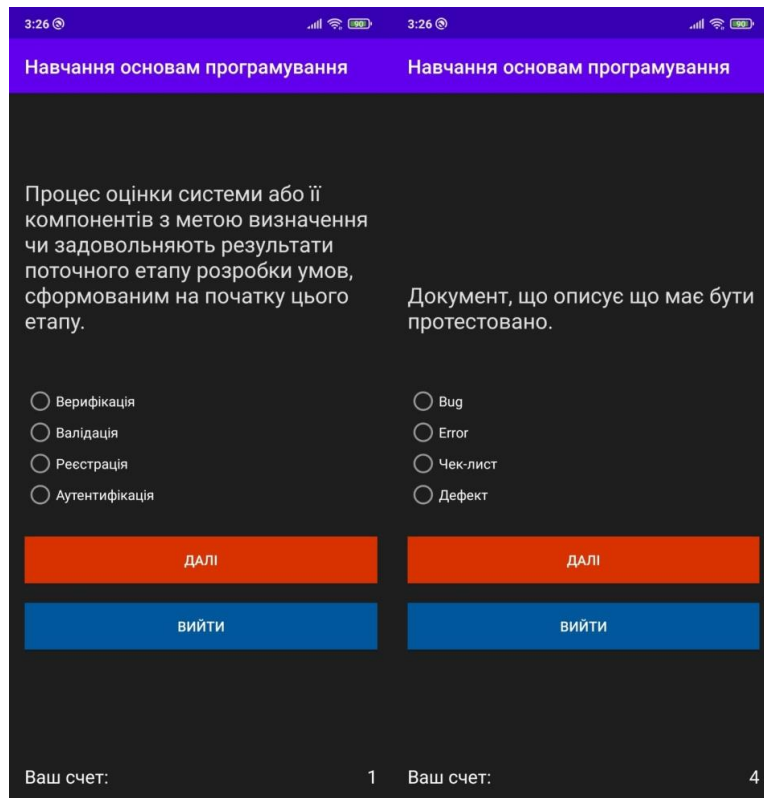


Рисунок 3.24 Тестування

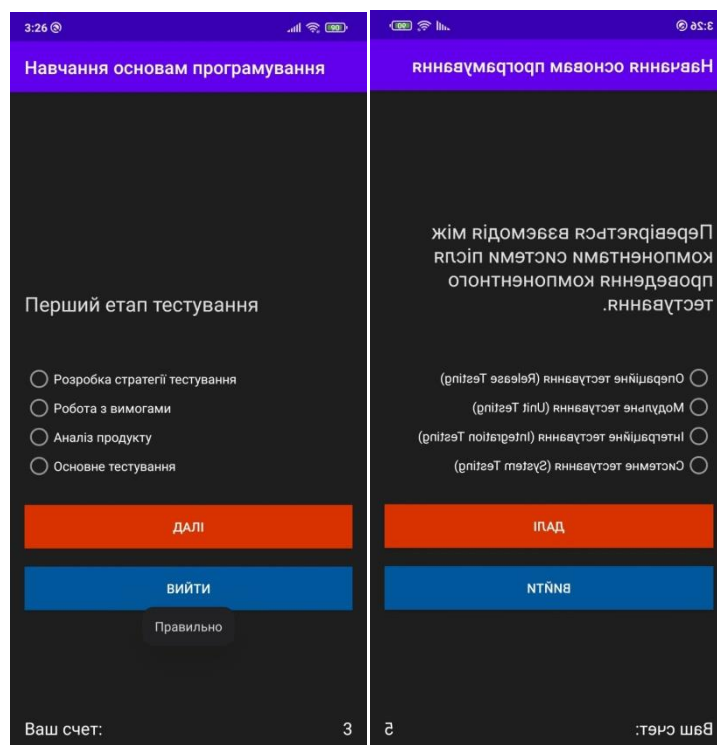


Рисунок 3.25 Тестування



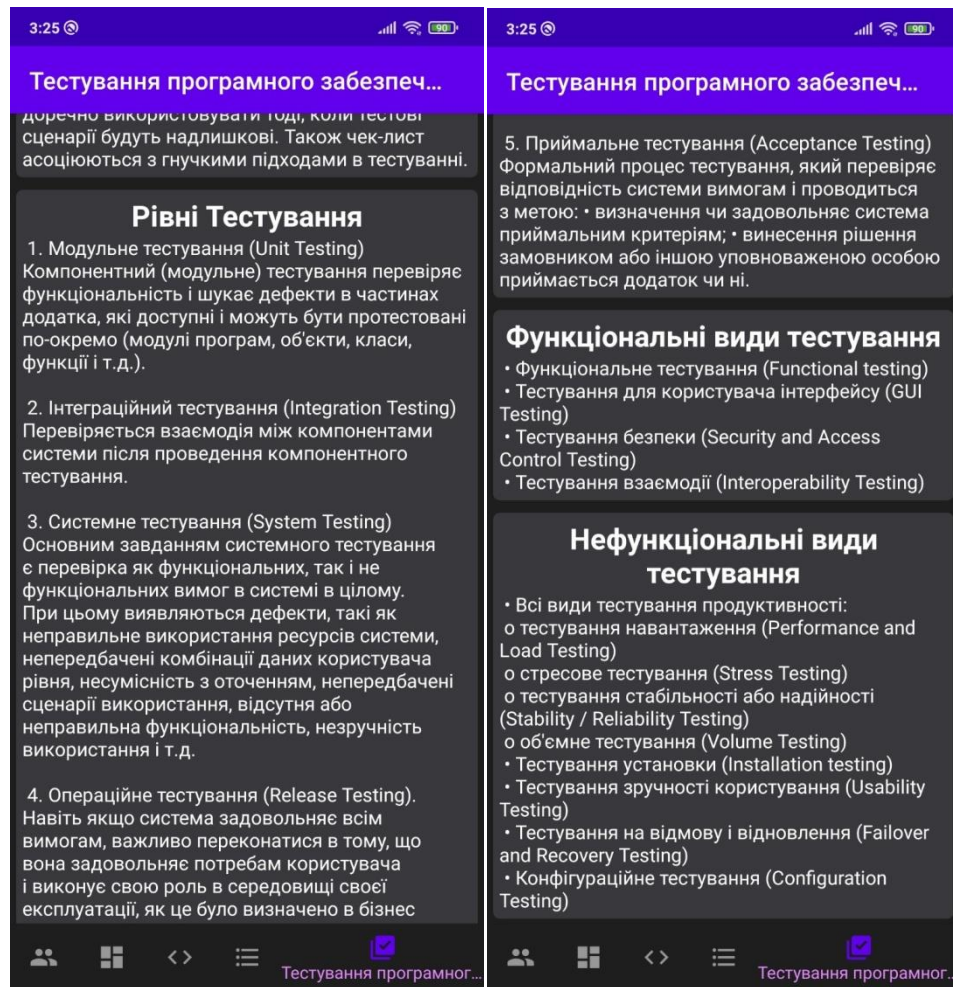


Рисунок 3.26 Тестування

Під час тестування збоїв та недоліків у роботі мобільного програмного комплексу для вивчення основ тестування програмного забезпечення не виявлено, що говорить про високу якість розробки та можливість впровадження за потребою.

## ВИСНОВКИ

У межах даної бакалаврської роботи проведено розробку мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.

Розроблений мобільний програмний комплекс для вивчення основ тестування програмного забезпечення володіє простим і зрозумілим інтерфейсом, в той же час володіє широкими функціональними можливостями.

Актуальність розробки ґрунтується на тому факті, що більшість власників сучасних мобільних телефонів використовують їх у якості своїх помічників у повсякденному житті, працюють з програмами для створення розкладів, написання заміток, установок нагадувань і будильників та є спектр користувачів, цільове застосування мобільних технологій яких, направлено на ігрові цілі. Це передусім молодь та діти, хоча застосування ігрових мобільних технологій контингентом більш старшого віку не є виключенням. Виходячи з того, що майже кожна сучасна людина володіє мобільними технологіями актуальність розробки не виявляється сумнівною.

У роботі вирішено такі завдання:

- здійснено порівняльну характеристику існуючих мобільних програмних комплексів для вивчення основ тестування програмного забезпечення;
- розроблено мобільну гру для вивчення основ тестування програмного забезпечення;
- виведено архітектуру розробки, структури даних, алгоритми ключових модулів та ситуаційного використання.

Результати роботи можуть бути використані при розробці, встановленні та налаштуванні програмного комплексу для вивчення основ тестування програмного забезпечення.

Результати роботи апробовані на науково-практичній конференції.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Global Positioning System [Электронный ресурс]. – Режим доступа: <http://www.navcen.uscg.gov>. – Загл. с экрана.
2. Enabling GPS programatically in Android. [Электронный ресурс]. – Режим доступа: <http://stackoverflow.com/questions/5715257/enabling-gps-programatically-in-android>. – Загл. с экрана.
3. General Information on GPS [Электронный ресурс]. – Режим доступа: <http://www.navcen.uscg.gov>. – Загл. с экрана.
4. Android - Architecture / Tutorialspoint [Офиц. сайт]. [Электронный ресурс]. – Режим доступа: [http://www.tutorialspoint.com/android/android\\_architecture.htm](http://www.tutorialspoint.com/android/android_architecture.htm) 26.10.2020
5. Introduction to Glide, Image Loader Library for Android, recommended by Google / The Cheese Factory: Blog [Офиц. сайт]. [Электронный ресурс]. – Режим доступа: <http://inthecheesefactory.com/blog/get-to-know-glide-recommended-bygoogle/en>. 26.10.2020
6. Material design - Introduction / Google [Офиц. сайт]. [Электронный ресурс]. – Режим доступа: <https://www.google.com/design/spec/materialdesign/introduction.html>. 26.10.2020
7. Android Wear by Google / Android [Офиц. сайт]. [Электронный ресурс]. – Режим доступа: <https://www.android.com/wear/> 26.10.2020
8. Google Play Hits 1 Million Apps [Электронный ресурс]. – Режим доступа: <http://mashable.com/2013/07/24/googleplay-1-million/> 26.10.2020
9. Android App Stats [Электронный ресурс]. – Режим доступа: <http://www.androlib.com/appstats.aspx> 26.10.2020
10. Google: 3 Billion Android Apps Installed; Downloads Up 50 Percent From Last Quarter [Электронный ресурс]. – Режим доступа: <http://techcrunch.com/2011/04/14/google-3-billion-android-appsinstalled-up-50-percent-from-last-quarter/> 26.10.2020

11. Smartphone OS Market Share, Q1 2015 [Электронный ресурс]. – Режим доступа: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> 26.10.2020
12. Apps for Programming on Android [Электронный ресурс]. – Режим доступа: <http://android.appstorm.net/roundups/developer/15-apps-for-programming-onandroid/> 26.10.2020
13. The Perfect Platform for Game Developers: Android [Электронный ресурс]. – Режим доступа: <http://www.developer.com/ws/android/client/the-perfect-platform-for-gamedevelopers-android.html> 16.05.2020
14. Introduction to Android [Электронный ресурс]. – Режим доступа: <https://developer.android.com/guide/index.html> 21.05.2020
15. Jake Wharton, [Web – resource], Exploring RxJava 2 for Android – access [Электронный ресурс]. – Режим доступа: <https://youtu.be/htlXKI5gOQU> GOTO 2016 open access.
16. David Karnok, [Web – resource] RxJava backpressure – access. [Электронный ресурс]. – Режим доступа: <https://github.com/ReactiveX/RxJava/wiki/Backpressure> open access, June 27, 2016.
17. T. Nield [e-book], Learning RxJava, Packt Publishing, 2017, 400 с.
18. App Components [Электронный ресурс]. – Режим доступа: <https://developer.android.com/guide/components/index.html>
19. AndroidSecurityBulletin – January 2017 [Электронный ресурс]. – Режим доступа : <https://source.android.com/security/bulletin /2017-01-01.html> – Назва з екрану
20. Мобильные вирусы и угрозы. Статистика [Электронный ресурс]. – Режим доступа : <http://www.kaspersky.ua/internet-security-center/threats/mobile/>
21. Karpachev I., Kazymur V. Functional security in an ANDROID mobile architecture / I. Karpachev, V. Kazymur // Вісник Чернігівського державного технологічного університету. 2015. № 1 (77).

22. Saad M. H., Serageldin A., Salama G. I. Android spyware disease and medication / M. H. Saad, A. Serageldin, G. I. Salama // Information Security and Cyber Forensics (InfoSec), 2015 Second International Conferenceon. – IEEE, 2015. С. 118–125.
23. Mark L. Murphy. The Busy Coder's Guide to Android Development / Mark L. Murphy // Commonsware, LLC. 2013. 443 p.
24. Understanding security on Android. [Електронний ресурс]. – Режим доступа: <http://www.ibm.com/developerworks/library/xandroidsecurity>.
25. Fedler R. Android OS security: risks and limitations / R. Fedler, C. Banse, C. Kraub, V. Fuesing. 2012. P. 19–27.
26. Mobile application security on android, context on android security / J. Burns // BlackHat. 2009. 27 p.
27. Ratazzietal P. A systematic security evaluation of android's multiuser framework / P. Ratazzietal // Proc. IEEE MoST, 2014. P. 1–10.
28. Fedler R. Effectiveness of malware protection on android and evaluation of android antivirus apps / R. Fedler, J. Schutte, M. Kulicke // Applied and integrated security. 2014. P. 7–13, 26–32.
29. Мелешко О. О. Способи захисту інформації з обмеженим доступом в мобільних пристроях від витоку / О. О. Мелешко, О. С. Болотнікова // Сучасний захист інформації. 2016. № 1.
30. Войтович О. П. Дослідження інцидентів безпеки в ОС Android / О. П. Войтович, М. В. Гурський // Тези доповідей XLV Науково-технічної конф. Вінницького національного технічного університету / ФІТКІ – Вінниця, 2016. С.55-60.

## ДОДАТКИ

### Додаток А

```

<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="AndroidLayouts">
    <shared>
      <config />
    </shared>
  </component>
  <component name="AndroidLogFilters">
    <option name="TOOL_WINDOW_LOG_LEVEL" value="error" />
    <option name="TOOL_WINDOW_CONFIGURED_FILTER" value="Show
only selected application" />
  </component>
  <component name="ChangeListManager">
    <list default="true" id="b0169431-b5af-4171-a135-
bb969a0680a9" name="Default Changelist" comment="" />
    <option name="EXCLUDED_CONVERTED_TO_IGNORED"
value="true" />
    <option name="SHOW_DIALOG" value="false" />
    <option name="HIGHLIGHT_CONFLICTS" value="true" />
    <option name="HIGHLIGHT_NON_ACTIVE_CHANGELIST"
value="false" />
    <option name="LAST_RESOLUTION" value="IGNORE" />
  </component>
  <component name="ExternalProjectsData">
    <projectState path="$PROJECT_DIR$"
    <ProjectState />
  </projectState>
</component>
  <component name="ExternalProjectsManager">
    <system id="GRADLE">
      <state>
        <projects_view />
      </state>
    </system>
  </component>
  <component name="FileTemplateManagerImpl">
    <option name="RECENT_TEMPLATES">
      <list>
        <option value="layoutResourceFile_vertical" />
        <option value="Class" />
        <option value="resourceFile" />
      </list>
    </option>
  </component>
  <component name="Git.Settings">
    <option name="RECENT_GIT_ROOT_PATH"
value="$PROJECT_DIR$/.." />
  </component>

```

```

    <component name="ProjectId"
id="1c60DNNxBJGJxpIBHXzR2FGLoRd" />
    <component name="ProjectLevelVcsManager"
settingsEditedManually="true" />
    <component name="PropertiesComponent">
        <property name="android.sdk.path"
value="$USER_HOME$/AppData/Local/Android/Sdk" />
        <property name="last_directory_selection"
value="$PROJECT_DIR$/app/src/main/res/drawable" />
        <property name="last_opened_file_path"
value="$PROJECT_DIR$/app/src/main/res/drawable" />
        <property name="settings.editor.selected.configurable"
value="AndroidSdkUpdater" />
    </component>
    <component name="RecentsManager">
        <key name="android.template.packageName">
            <recent name="com.modern.codequiz.ui" />
            <recent name="com.modern.codequiz.ui.paradigms_quiz"
/>
            <recent name="com.modern.codequiz" />
        </key>
        <key name="CopyFile.RECENT_KEYS">
            <recent
name="C:\Users\79127\AndroidStudioProjects\programming
outlook\app\src\main\res\drawable" />
            <recent
name="C:\Users\79127\Downloads\2\app\src\main\res\drawable"
/>
        </key>
    </component>
    <component name="RunDashboard">
        <option name="ruleStates">
            <list>
                <RuleState>
                    <option name="name"
value="ConfigurationTypeDashboardGroupingRule" />
                </RuleState>
                <RuleState>
                    <option name="name"
value="StatusDashboardGroupingRule" />
                </RuleState>
            </list>
        </option>
    </component>
    <component name="RunManager">
        <configuration default="true" type="AndroidJUnit"
factoryName="Android JUnit">
            <option name="TEST_OBJECT" value="class" />
            <option name="WORKING_DIRECTORY" value="$MODULE_DIR$"
/>
            <method v="2">
                <option name="Android.Gradle.BeforeRunTask"
enabled="true" />

```

```

        </method>
    </configuration>
    <configuration name="app"
type="AndroidRunConfigurationType" factoryName="Android App"
activateToolWindowBeforeRun="false">
        <module name="app" />
        <option name="DEPLOY" value="true" />
        <option name="DEPLOY_APK_FROM_BUNDLE" value="false" />
        <option name="DEPLOY_AS_INSTANT" value="false" />
        <option name="ARTIFACT_NAME" value="" />
        <option name="PM_INSTALL_OPTIONS" value="" />
        <option name="DYNAMIC_FEATURES_DISABLED_LIST" value=""
/>
        <option name="ACTIVITY_EXTRA_FLAGS" value="" />
        <option name="MODE" value="default_activity" />
        <option name="CLEAR_LOGCAT" value="false" />
        <option name="SHOW_LOGCAT_AUTOMATICALLY" value="false"
/>
        <option name="SKIP_NOOP_APK_INSTALLATIONS"
value="true" />
        <option name="FORCE_STOP_RUNNING_APP" value="true" />
        <option name="TARGET_SELECTION_MODE"
value="DEVICE_AND_SNAPSHOT_COMBO_BOX" />
        <option name="SELECTED_CLOUD_MATRIX_CONFIGURATION_ID"
value="-1" />
        <option name="SELECTED_CLOUD_MATRIX_PROJECT_ID"
value="" />
        <option name="DEBUGGER_TYPE" value="Auto" />
        <Auto>
            <option name="USE_JAVA_AWARE_DEBUGGER" value="false"
/>
            <option name="SHOW_STATIC_VARS" value="true" />
            <option name="WORKING_DIR" value="" />
            <option name="TARGET_LOGGING_CHANNELS" value="lldb
process:gdb-remote packets" />
            <option name="SHOW_OPTIMIZED_WARNING" value="true"
/>
        </Auto>
        <Hybrid>
            <option name="USE_JAVA_AWARE_DEBUGGER" value="false"
/>
            <option name="SHOW_STATIC_VARS" value="true" />
            <option name="WORKING_DIR" value="" />
            <option name="TARGET_LOGGING_CHANNELS" value="lldb
process:gdb-remote packets" />
            <option name="SHOW_OPTIMIZED_WARNING" value="true"
/>
        </Hybrid>
        <Java />
        <Native>
            <option name="USE_JAVA_AWARE_DEBUGGER" value="false"
/>
            <option name="SHOW_STATIC_VARS" value="true" />

```



```

        <option name="WORKING_DIR" value="" />
        <option name="TARGET_LOGGING_CHANNELS" value="lldb
process:gdb-remote packets" />
        <option name="SHOW_OPTIMIZED_WARNING" value="true"
/>
    </Native>
    <Profilers>
        <option name="ADVANCED_PROFILING_ENABLED"
value="false" />
        <option name="STARTUP_CPU_PROFILING_ENABLED"
value="false" />
        <option
name="STARTUP_CPU_PROFILING_CONFIGURATION_NAME"
value="Sample Java Methods" />
    </Profilers>
    <option name="DEEP_LINK" value="" />
    <option name="ACTIVITY_CLASS" value="" />
    <method v="2">
        <option name="Android.Gradle.BeforeRunTask"
enabled="true" />
    </method>
</configuration>
</component>
<component name="SvnConfiguration">
    <configuration />
</component>
<component name="TaskManager">
    <task active="true" id="Default" summary="Default task">
        <changelist id="b0169431-b5af-4171-a135-bb969a0680a9"
name="Default Changelist" comment="" />
        <created>1589833614180</created>
        <option name="number" value="Default" />
        <option name="presentableId" value="Default" />
        <updated>1589833614180</updated>
    </task>
    <servers />
</component>
<component name="Vcs.Log.Tabs.Properties">
    <option name="TAB_STATES">
        <map>
            <entry key="MAIN">
                <value>
                    <State>
                        <option name="COLUMN_ORDER" />
                    </State>
                </value>
            </entry>
        </map>
    </option>
</component>
</project>

```

## Додаток Б

```

<component name="ProjectCodeStyleConfiguration">
  <code_scheme name="Project" version="173">
    <codeStyleSettings language="XML">
      <indentOptions>
        <option name="CONTINUATION_INDENT_SIZE" value="4" />
      </indentOptions>
      <arrangement>
        <rules>
          <section>
            <rule>
              <match>
                <AND>
                  <NAME>xmlns:android</NAME>
                  <XML_ATTRIBUTE />
                  <XML_NAMESPACE>^$</XML_NAMESPACE>
                </AND>
              </match>
            </rule>
          </section>
          <section>
            <rule>
              <match>
                <AND>
                  <NAME>xmlns:.*</NAME>
                  <XML_ATTRIBUTE />
                  <XML_NAMESPACE>^$</XML_NAMESPACE>
                </AND>
              </match>
              <order>BY_NAME</order>
            </rule>
          </section>
          <section>
            <rule>
              <match>
                <AND>
                  <NAME>.*:id</NAME>
                  <XML_ATTRIBUTE />
                </AND>
              </match>
            </rule>
          </section>
          <section>
            <rule>
              <match>
                <AND>
                  <XML_NAMESPACE>http://schemas.android.com/apk/res/android</XML_NAMESPACE>
                </AND>
              </match>
            </rule>
          </section>
          <section>
            <rule>
              <match>
                <AND>
                  <NAME>.*:name</NAME>
                  <XML_ATTRIBUTE />
                </AND>
              </match>
            </rule>
          </section>
        </rules>
      </arrangement>
    </codeStyleSettings>
  </code_scheme>
</component>

```

```

<XML_NAMESPACE>http://schemas.android.com/apk/res/android</X
ML_NAMESPACE>
    </AND>
  </match>
</rule>
</section>
<section>
  <rule>
    <match>
      <AND>
        <NAME>name</NAME>
        <XML_ATTRIBUTE />
        <XML_NAMESPACE>^$</XML_NAMESPACE>
      </AND>
    </match>
  </rule>
</section>
<section>
  <rule>
    <match>
      <AND>
        <NAME>style</NAME>
        <XML_ATTRIBUTE />
        <XML_NAMESPACE>^$</XML_NAMESPACE>
      </AND>
    </match>
  </rule>
</section>
<section>
  <rule>
    <match>
      <AND>
        <NAME>.*</NAME>
        <XML_ATTRIBUTE />
        <XML_NAMESPACE>^$</XML_NAMESPACE>
      </AND>
    </match>
    <order>BY_NAME</order>
  </rule>
</section>
<section>
  <rule>
    <match>
      <AND>
        <NAME>.*</NAME>
        <XML_ATTRIBUTE />
      </AND>
    </match>
  </rule>
</section>
<XML_NAMESPACE>http://schemas.android.com/apk/res/android</X
ML_NAMESPACE>
  </AND>
</match>
<order>ANDROID_ATTRIBUTE_ORDER</order>

```

```
    </rule>
  </section>
  <section>
    <rule>
      <match>
        <AND>
          <NAME>.*</NAME>
          <XML_ATTRIBUTE />
          <XML_NAMESPACE>.*</XML_NAMESPACE>
        </AND>
      </match>
      <order>BY_NAME</order>
    </rule>
  </section>
</rules>
</arrangement>
</codeStyleSettings>
</code_scheme>
</component>
```

**Додаток В**

```

@if "%DEBUG%" == "" @echo off
@rem
#####
#####
@rem
@rem  Gradle startup script for Windows
@rem
@rem
#####
#####

@rem Set local scope for the variables with windows NT shell
if "%OS%"=="Windows_NT" setlocal

set DIRNAME=%~dp0
if "%DIRNAME%" == "" set DIRNAME=.
set APP_BASE_NAME=%~n0
set APP_HOME=%DIRNAME%

@rem Add default JVM options here. You can also use
JAVA_OPTS and GRADLE_OPTS to pass JVM options to this
script.
set DEFAULT_JVM_OPTS=

@rem Find java.exe
if defined JAVA_HOME goto findJavaFromJavaHome

set JAVA_EXE=java.exe
%JAVA_EXE% -version >NUL 2>&1
if "%ERRORLEVEL%" == "0" goto init

echo.
echo ERROR: JAVA_HOME is not set and no 'java' command could
be found in your PATH.
echo.
echo Please set the JAVA_HOME variable in your environment
to match the
echo location of your Java installation.

goto fail

:findJavaFromJavaHome
set JAVA_HOME=%JAVA_HOME:"=%
set JAVA_EXE=%JAVA_HOME%/bin/java.exe

if exist "%JAVA_EXE%" goto init

echo.
echo ERROR: JAVA_HOME is set to an invalid directory:
%JAVA_HOME%
echo.

```

```

echo Please set the JAVA_HOME variable in your environment
to match the
echo location of your Java installation.

goto fail

:init
@rem Get command-line arguments, handling Windows variants

if not "%OS%" == "Windows_NT" goto win9xME_args

:win9xME_args
@rem Slurp the command line arguments.
set CMD_LINE_ARGS=
set _SKIP=2

:win9xME_args_slurp
if "x%~1" == "x" goto execute

set CMD_LINE_ARGS=%*

:execute
@rem Setup the command line

set CLASSPATH=%APP_HOME%\gradle\wrapper\gradle-wrapper.jar

@rem Execute Gradle
"%JAVA_EXE%" %DEFAULT_JVM_OPTS% %JAVA_OPTS% %GRADLE_OPTS% "-
Dorg.gradle.appname=%APP_BASE_NAME%" -classpath
"%CLASSPATH%" org.gradle.wrapper.GradleWrapperMain
%CMD_LINE_ARGS%

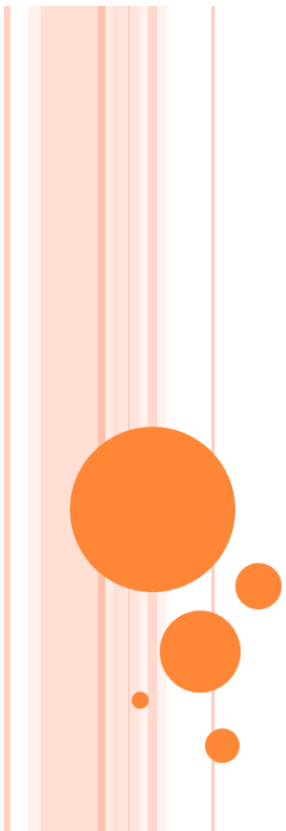
:end
@rem End local scope for the variables with windows NT shell
if "%ERRORLEVEL%"=="0" goto mainEnd

:fail
rem Set variable GRADLE_EXIT_CONSOLE if you need the
_script_ return code instead of
rem the _cmd.exe /c_ return code!
if not "" == "%GRADLE_EXIT_CONSOLE%" exit 1
exit /b 1

:mainEnd
if "%OS%"=="Windows_NT" endlocal

:omega


```



## РОЗРОБКА МОБІЛЬНОГО ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ВИВЧЕННЯ ОСНОВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Дипломна робота

### ***АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ***

- На сьогодні, інформаційні технології, різного виду щільно увійшли в життя сучасної людини. Мобільні пристрої, легкі для перенесення та обігу, грають величезну роль в житті людини. Зростання ринку мобільних пристроїв вже перестало бути лінійним. Кількість великих гравців ринку до сьогоднішнього моменту збільшилася в кілька разів порівняно з початком 2000-х. На початку-середині 2000-х, коли мобільні пристрої стали доступні широким масам, для мобільних пристроїв, стали розроблятися різні програми, націлені на вирішення локальних задач, в більшості своїй не пов'язаних з доступом в мережу Інтернет.
  - Більшість власників сучасних мобільних телефонів використовують їх у якості своїх помічників у повсякденному житті, працюють з програмами для створення розкладів, написання заміток, установок нагадувань і будильників та є спектр користувачів, цільове застосування мобільних технологій яких, направлено на ігрові цілі. Це передусім молодь та діти, хоча застосування ігрових мобільних технологій контингентом більш старшого віку не є виключенням.
- 

## *МЕТА ТА ЗАВДАННЯ РОБОТИ*

- Мета проекту – розробити мобільний програмний комплекс для вивчення основ тестування програмного забезпечення.
- Для досягнення поставленої мети необхідно вирішити такі завдання:
- Здійснити порівняльну характеристику існуючих мобільних програмних комплексів для вивчення основ тестування програмного забезпечення;
- розробити мобільний програмний комплекс для вивчення основ тестування програмного забезпечення;
- вивести архітектуру розробки, структури даних, алгоритми ключових модулів та ситуаційного використання.

## *ОБ’ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ*

- Об’єктом роботи виступає мобільний програмний комплекс для вивчення основ тестування програмного забезпечення.
- Предметом є процес розробки мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.



## ДОСЛІДЖЕННЯ АНАЛОГІВ

Новим напрямком програмного забезпечення тестування є застосування мобільних сервісів, до яких можна віднести такі:

- iSpring Play (раніше iSpring Viewer) - мобільний додаток, що дозволяє переглядати курси і тести.
- Kahoot! - це сервіс не тільки для створення різноманітних форм тестів, але і для проведення онлайн вікторин за допомогою спеціального клієнта, що встановлюється на смартфонах учнів (Android, iOS, Windows Phone).
- Quizlet - це сервіс для створення тестів, в яких вибирається правильна відповідь із запропонованих варіантів.
- Plickers - це сервіс у вигляді мобільного додатку, який дозволяє влаштовувати опитування на смартфонах учнів в аудиторії і аналізувати викладачеві статистику по всіх студентах.
- Easy Test Maker - це сервіс у вигляді мобільного додатку для створення тестів, в яких потрібно вибрати вірні і помилкові твердження.

## ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

Потрібно розробити клієнт-серверний додаток на базі ОС Android, де клієнт може отримати інформацію, про історію програмування, про видатних програмістів, про генезис мов програмування тощо. Пройти тестування за отриманими знаннями. Взяти участь у вікторинах.

Програмний засіб має забезпечувати наступні функціональні можливості:

- 1. відображення історичної інформації
- 2. зчитування інформації з поля введення
- 3. реалізація тестування
- 4. зчитування даних відповідей
- 5. відображення інформації про результат проходження тестування.

## *НАУКОВА НОВИЗНА ОТРИМАНИХ РЕЗУЛЬТАТІВ*

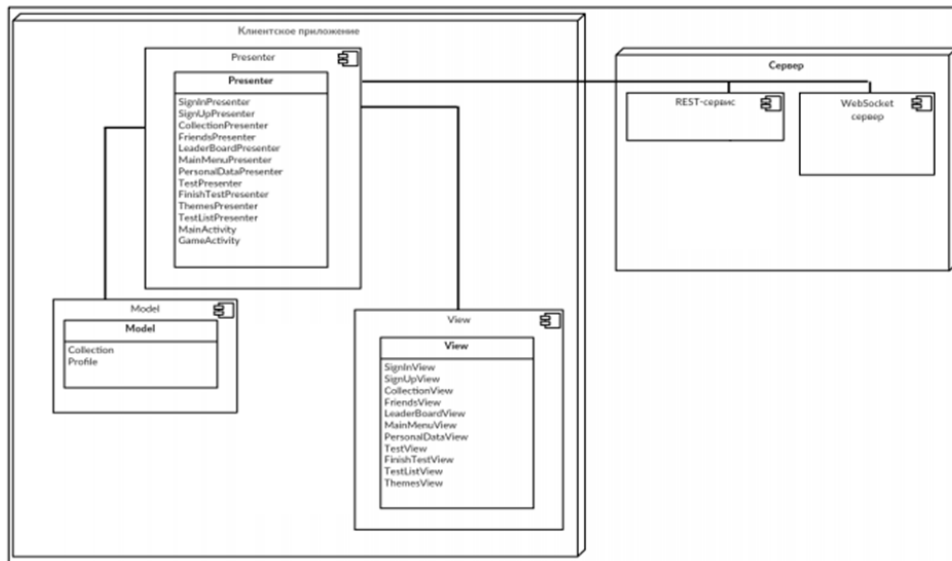
- запропоновано комплексне рішення операційної системи;
- наведено модель мобільного програмного комплексу для вивчення основ тестування програмного забезпечення;
- методи аналізу середовища мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.

## ПРОЕКТУВАННЯ МОБІЛЬНОЇ ГРИ ДЛЯ ВИВЧЕННЯ ОСНОВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

- Функціональні вимоги до проєктованого мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.
- Функціональні вимоги визначають функціональність ПЗ, яку потрібно побудувати, щоб користувачі змогли виконати свої завдання. В ході аналізу були виявлені наступні вимоги:
  - 1) додаток «Для вивчення основ тестування програмного забезпечення» має нести інформацію про історію програмування, мови програмування, розробників та тестування;
  - 2) додаток «Для вивчення основ тестування програмного забезпечення» має надавати можливість брати участь у опитуванні;
  - 3) додаток «Для вивчення основ тестування програмного забезпечення» має надавати можливість проходити тести;
  - 4) додаток «Для вивчення основ тестування програмного забезпечення» має перевіряти відповіді на завдання;

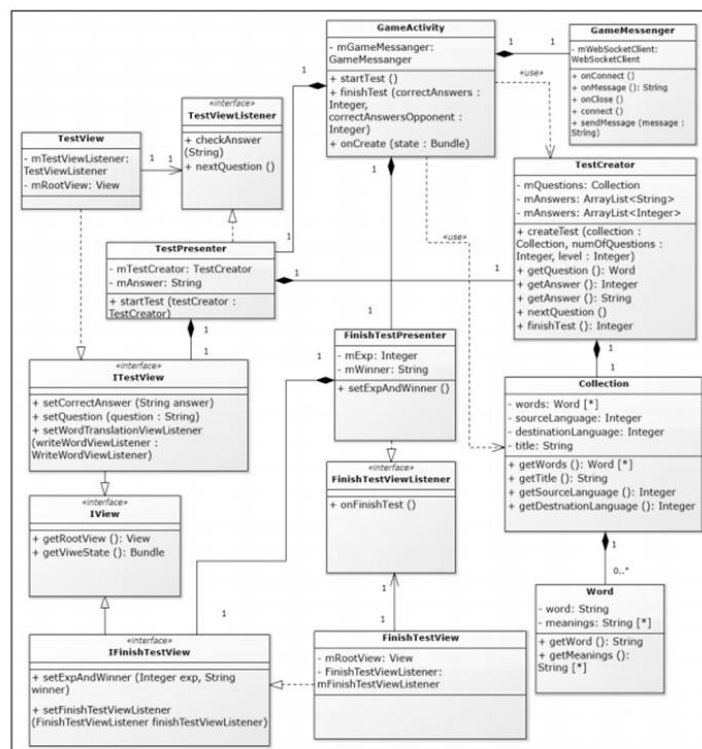


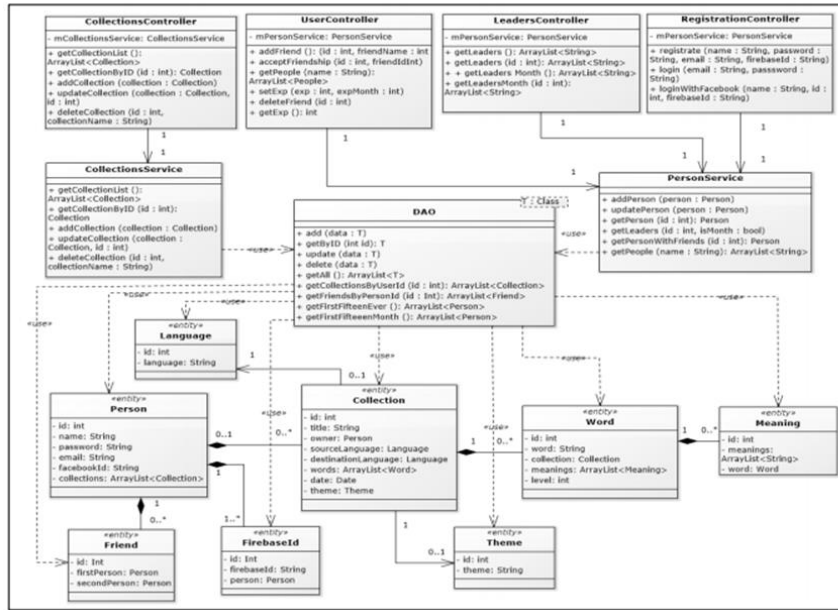
- Діаграма використання



- Архітектура системи «Для вивчення основ тестування програмного забезпечення»

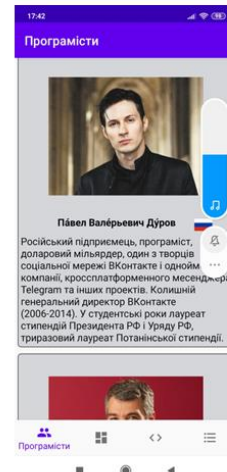
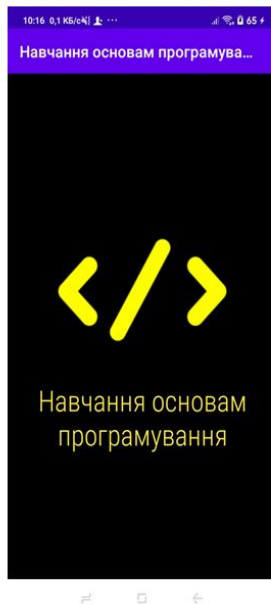
- Діаграма класів режиму тестування



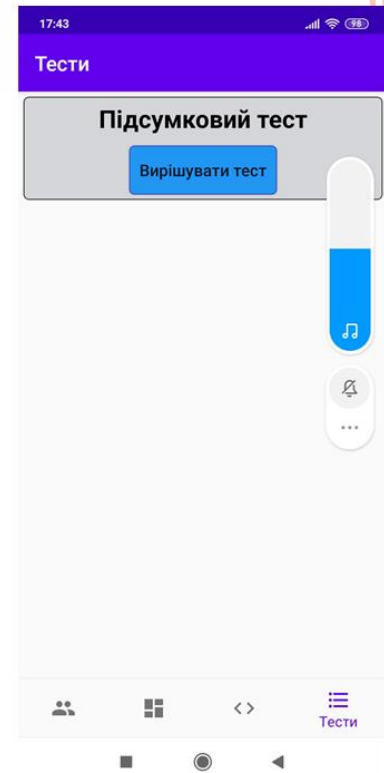
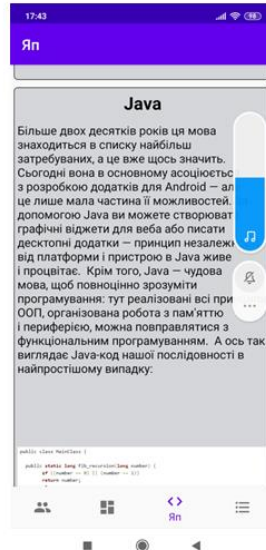
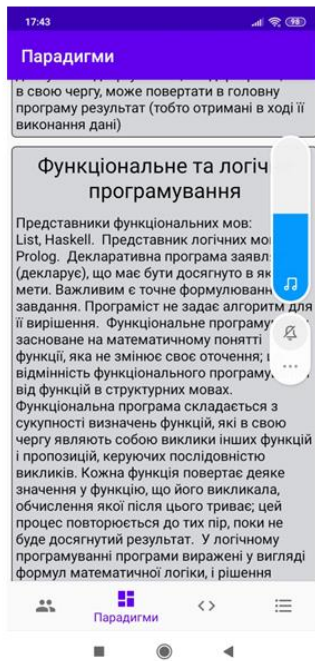


○ Діаграма класів REST-сервісу

## РОЗРОБКА МОБІЛЬНОЇ ГРИ ДЛЯ ВИВЧЕННЯ ОСНОВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



## РЕЗУЛЬТАТИ РОБОТИ



## ВИСНОВКИ

- У межах даної бакалаврської роботи проведено розробку мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.
- Розроблений мобільний програмний комплекс для вивчення основ тестування програмного забезпечення володіє простим і зрозумілим інтерфейсом, в той же час володіє широкими функціональними можливостями.
- Ґрунтуючись на дослідженнях, варто відзначитись, що додаток має такі особливості:
  - мобільність – можна скористатися додатком, де б ви не знаходилися і в будь-який час;
  - можливість ознайомитися з історією програмування та пройти тестування;
  - можливість взяти участь у вікторинах;
  - збереження інформації.
- Проект реалізовано на базі операційної системи Android. Її переваги:
  - - Повна відкритість;
  - - Універсальність;
  - - Зростаюча популярність;
  - - Динамічний розвиток.
- Таким чином, була вивчена предметна область, визначено ряд проблем, які необхідно вирішити, виявлені необхідні складові інформаційної системи, обрана апаратна і програмна основа для реалізації мобільного програмного комплексу для вивчення основ тестування програмного забезпечення.