

Київський національний торговельно-економічний університет

Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

*«Модель навчального мобільного додатку інтеграції керування
знаннями і безперервним навчанням»*

Студента 2м курсу, 2м групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Шапочки Дениса
Вікторовича

Науковий керівник
доктор технічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Криворучко Олена
Володимирівна

Гарант освітньої програми
доктор економічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Токар Володимир
Володимирович

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії

програмного забезпечення та
кібербезпеки

Криворучко О. В.

"10" листопада 2020 р.

Завдання на випускний кваліфікаційний проєкт студентіві

Шапочці Денису Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Модель навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням»

Затверджена наказом ректора від «30» листопада 2020 р. № 3224

2. Строк здачі студентом закінченої роботи 25 листопада 2021

3. Цільова установка та вихідні дані до роботи

Мета роботи випускного кваліфікаційного проєкту є застосування мобільних додатків для безперервного навчання та їх створення

Об'єкт дослідження мобільні додатки

Предмет дослідження є методи та алгоритми створення мобільного додатку

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ОСНОВНІ ПОЛОЖЕННЯ МОБІЛЬНОГО НАВЧАННЯ ТА НАВЧАННЯ ПРОТЯГОМ ЖИТТЯ

1.1. Основні положення та переваги мобільного навчання(M-learning)

1.2. Поняття та принципи навчання протягом життя(LifeLongLearning).....

1.3. Висновки до Розділу 1

РОЗДІЛ 2. ЗАСОБИ РОЗРОБКИ НАВЧАЛЬНОГО МОБІЛЬНОГО ДОДАТКУ ІНТЕГРАЦІЇ КЕРУВАННЯ ЗНАННЯМИ І БЕЗПЕРЕРВНИМ НАВЧАННЯМ

2.1. Опис та переваги ОС Android

2.2. Переваги використання мови програмування Java.....

2.3. Опис та обґрунтування вибору бібліотеки FBReader SDK.....

2.4. Висновки до Розділу 2

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1. Створення проєкту та підключення FBReader SDK

3.2. Написання додатку.....

3.3. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....

ДОДАТКИ.....

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2020	21.09.2020
2.	<i>Розробка та затвердження завдання на проєкт магістра</i>	06.11.2020	22.12.2020
3.	<i>Вступ та перелік літературних джерел</i>	27.02.2021	27.02.2021
4.	<i>Розробка технічного завдання</i>	20.03.2021	20.03.2021
5.	<i>Розділ 1. «Основні положення мобільного навчання та навчання протягом життя»</i>	16.04.2021	16.04.2021
6.	<i>Розділ 2. «Засоби розробки навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням»</i>	24.05.2021	24.05.2021
7.	<i>Розділ 3. «Розробка програмного продукту»</i>	21.06.2021	21.06.2021
8.	<i>Розробка програми та методики тестування</i>	18.10.2021	18.10.2021
9.	<i>Написання наукової статті</i>	22.05.2021	22.05.2021
10.	<i>Керівництво користувача</i>	21.10.2021	21.10.2021
11.	<i>Висновки та пропозиції</i>	01.11.2021	01.11.2021
12.	<i>Здача випускного кваліфікаційного проєкту на кафедру (перша перевірка)</i>	03.11.2021	03.11.2021
13.	<i>Підготовка автореферату та презентації доповіді</i>	03.11.2021	03.11.2021
14.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	22.11.2021- 25.11.2021	22.11.2021
15.	<i>Здача зброшурованого випускного кваліфікаційного проєкту</i>	25.11.2021	25.11.2021
16.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	26.11.2021	26.11.2021
17.	<i>Підготовка до публічного захисту випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «10» листопада 2020 р.

8. Науковий керівник випускного кваліфікаційного проєкту _____
Криворучко О.В.
(прізвище, ініціали, підпис)

9. Гарант освітньої програми _____ Токар В.В.
(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент _____ Шапочка Д.В.
(прізвище, ініціали, підпис)

11. Відгук керівника випускного кваліфікаційного проєкту

Науковий керівник випускного кваліфікаційного проєкту

(підпис, дата)

Відмітка про попередній захист _____

(ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проєкт

Випускний кваліфікаційний проєкт студента _____ Шапочки Д.В.

(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми _____

Токар В. В.

(прізвище, ініціали, підпис)

Завідувач кафедри _____

Криворучко О. В.

(підпис, прізвище, ініціали)

« _____ » _____ 20 ____ р.

АНОТАЦІЯ

випускної кваліфікаційного проєкту Шапочки Дениса Вікторовича
на тему: *«Модель навчального мобільного додатку інтеграції керування
знаннями і безперервним навчанням»*

Даний випускний кваліфікаційний проєкт присвячений розробці мобільного додатку інтеграції керування знаннями і безперервним навчанням. Метою роботи є розробка Модель навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням. У роботі було досліджено основні теоретичні відомості про технології та методики керування знаннями та безперервним навчанням. Проведено аналіз та дослідження використання навчальних програм та методик, а також проаналізовано їх загальні характеристики та стан в світі. Створено — технічне завдання та постановка задачі. Також, було наведено доцільність використання для даної роботи таких компонентів, як: мова Java, графічна бібліотека FBReader SDK та середовища розробки Android Studio. Детально описано процес створення програми, та її головних функціональних моментів.

Загальний обсяг роботи: 50 сторінок, 3 розділи, 5 рисунків, 23 джерела.

Ключові слова: мобільне навчання(M-learning), навчання протягом життя(LifeLongLearning), мобільний додаток, додаток для читання.

ANNOTATION

final qualifying project of Denis Viktorovich Shapochka
on the topic: "Model of educational mobile application for integration of
knowledge management and lifelong learning"

This final qualifying project is devoted to the development of a mobile application for the integration of knowledge management and lifelong learning. The aim of the work is to develop a model of educational mobile application for the integration of knowledge management and lifelong learning. The paper investigates

the basic theoretical information about technologies and methods of knowledge management and lifelong learning. The analysis and research of the use of educational programs and methods, as well as their general characteristics and the state of the world are analyzed. Created - technical task and problem statement. Also, it was advisable to use components for this work, such as: Java language, FBReader SDK graphics library and Android Studio development environment. The process of creating the program and its main functional moments are described in detail.

Total volume of work: 50 pages, 3 sections, 5 figures, 23 sources.

Keywords: mobile learning (M-learning), lifelong learning (LifeLongLearning), mobile application, reading application.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. ОСНОВНІ ПОЛОЖЕННЯ МОБІЛЬНОГО НАВЧАННЯ ТА НАВЧАННЯ ПРОТЯГОМ ЖИТТЯ	5
1.1. Основні положення та переваги мобільного навчання(M-learning).....	5
1.2. Поняття та принципи навчання протягом життя(LifeLongLearning).....	10
1.3. Висновки до Розділу 1.....	14
РОЗДІЛ 2. ЗАСОБИ РОЗРОБКИ НАВЧАЛЬНОГО МОБІЛЬНОГО ДОДАТКУ ІНТЕГРАЦІЇ КЕРУВАННЯ ЗНАННЯМИ І БЕЗПЕРЕРВНИМ НАВЧАННЯМ	15
2.1. Опис та переваги ОС Android.....	15
2.2. Переваги використання мови програмування Java.....	19
2.3. Опис та обґрунтування вибору бібліотеки FBReader SDK.....	24
2.4. Висновки до Розділу 2.....	25
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	26
3.1. Створення проєкту та підключення FBReader SDK.....	26
3.2. Написання додатку.....	29
3.3. Висновки до розділу 3.....	45
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТКИ	50

					КНТЕУ 121-02-22.МР			
Зм.	Аркуш	№ докум	Підпис	Дата				
<i>Зав. кафедри</i>		Криворучко О.В.		01.11.2021	<i>Модель навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>		Криворучко О.В.		01.11.2021		3	2	49
<i>Гарант</i>		Токар В.В.		01.11.2021		Факультет інформаційних технологій, 2м курс, 2 група		
<i>Розроб.</i>		Шапочка Д.В..		01.11.2021	Зміст			

ВСТУП

Актуальність: Мобільні додатки стали одним з головних трендів у розвитку інформаційних технологій за останні роки. Перевагою мобільних додатків є те, що користувач може отримати доступ до них, перебуваючи в будь-якому місці та в будь-якій ситуації.

Об'єктом дослідження є інформатизація аспекту безперервного навчання.

Предметом дослідження є методи та алгоритми створення мобільного додатку.

Мета роботи полягає у створенні мобільного додатку за допомогою якого полегшується процес безперервного навчання.

Розробка мобільних додатків - це ядро основних бізнес-стратегій; традиційна освітня інфраструктура також завадила корінного перегляду процесу електронного навчання. Ключовою рушійною силою зростання онлайн-освіти є зростаючий попит з боку учнів, які бажають підвищити свій рівень навчання.

Мобільні додатки, що ростуть, технологічні досягнення і простота доступу в Інтернет допомагають забезпечити індивідуальний підхід до навчання для учнів усіх типів. Крім студентів, мобільні додатки електронного навчання також розробляються для підтримки навчання та навчання користувачів на корпоративному рівні.

Студенти або корпоративні учні тепер можуть гнучко використовувати кращі функції мобільного додатка, пройти навчання і отримати інтерактивний досвід навчання. Найкраще, що ми можемо зробити з

					<i>КНТЕУ 121-02--22.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>		<i>27.02.2021</i>	<i>Модель навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>		<i>Криворучко О.В.</i>		<i>27.02.2021</i>		<i>В</i>	<i>3</i>	<i>49</i>
<i>Гарант</i>		<i>Токар В.В.</i>		<i>27.02.2021</i>		<i>Факультет інформаційних технологій, 2м курс, 2 група</i>		
<i>Розроб.</i>		<i>Шапочка Д.В.</i>		<i>27.02.2021</i>				
					<i>Вступ</i>			

додатками електронного навчання, - це зруйнувати стіни віддаленості, вийти за рамки демографічних проблем, отримати доступ до зручного навчання (в будь-який час і в будь-якому місці) на будь-якому пристрої і внести свій вклад в розвиток віртуального середовища навчання.

Мобільні технології можуть вивести електронне навчання на новий рівень. Ці і пристрої можуть забезпечити кращу і всебічну підтримку корпоративного навчання. Крім того, очікується, що вони додадуть імерсивності моделювання, досвіду навчання, що дозволить учням поринути в історію, засвоїти поліпшені елементи навчання і застосовувати їх в сценаріях в реальному часі. Іноді фахівці по ланцюжках поставок, логістики та продажу можуть скористатися пост-тренінговою підтримкою в режимі реального часу, використовуючи пристрої і усуваючи проблеми. В цьому випадку можна використовувати мобільні технології.

Ці технологічні інтеграції в мобільні додатки змінять сценарій електронного навчання і зроблять його більш ефективним для учнів. UdeMy, Coursera і індійський гігант електронного навчання BYJU вже потрясли ринок освіти і традиційну концепцію викладання і навчання; ця еволюція не відбулася раптово. Ці технологічні тенденції надали безліч переваг як для вчених, так і для корпорацій, зробивши додатки електронного навчання важливим доповненням, що дозволяє студентам і співробітникам навчатися і ефективно утримувати знання.

Можна сказати, що технології змінюють правила гри в освіті або корпоративному навчанні. Поки організація не дізнається про тенденції, їй буде важко лідирувати в конкурентній боротьбі. Мобільні додатки для електронного навчання, які забезпечують зростання учнів, а також заохочують реальне професійний розвиток, можуть сприяти кращим результатам навчання.

					КНТЕУ 121-02-22.МР	Аркуш
						4
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 1

ОСНОВНІ ПОЛОЖЕННЯ МОБІЛЬНОГО НАВЧАННЯ ТА НАВЧАННЯ ПРОТЯГОМ ЖИТТЯ

1.1. Основні положення та переваги мобільного навчання (M-learning)

Сьогодні люди витрачають значну кількість їх вільного часу в мережі. Дуже часто, вони грають в ігри на мобільних пристроях, в тому числі; планшети або смартфони. ці розробки неминуче призвели до нового зсуву парадигми; як навчання через гру або використання мобільних технологій в освіті. Змінилось те, як студенти думають і обробляють інформацію. Дуже часто, технології, в тому числі ігри, забезпечують імерсивні, добровільні і приємні моменти в діяльності. У той же час ці інновації все більше задовольняють основні вимоги освітніх програм шкіл оскільки вони прагнуть забезпечити захоплююче навчальне середовище для своїх учнів. Повсюдно поширені технології можуть використовуватися поза контекстом розваги; і вважаються частиною продуманого просування до навчання, заснованого на відкриттях.

M-Learning включає використання мобільних технологій окремо або разом з іншими інформаційно-комунікаційними технологіями (ІКТ). Таким чином, цей тип технології може надати студентам можливості будувати та вдосконалювати знання в будь-який час або в будь-якому місці. Відповідно до цього, M-Learning може відбуватися в ситуаціях, коли технології можуть запропонувати студенту засоби побудови своїх знань. Однак просте,

					<i>КНТЕУ 121-02-22.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>		<i>16.04.2021</i>	<i>Модель навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>		<i>Криворучко О.В.</i>		<i>16.04.2021</i>		<i>P1</i>	<i>5</i>	<i>49</i>
<i>Гарант</i>		<i>Токар В.В.</i>		<i>16.04.2021</i>		<i>Факультет інформаційних технологій, 2м курс, 2 група</i>		
<i>Розроб.</i>		<i>Шапочка Д.В.</i>		<i>16.04.2021</i>				
					<i>Основні положення мобільного навчання та навчання протягом життя</i>			

випадкове використання мобільного пристрою для виконання ізольованої діяльності в класі не є мобільним навчанням. Для того, щоб зрозуміти, вчитель повинен інтегрувати використання технологій з педагогічним плануванням, яке передбачає вивчення змісту, навчальних матеріалів, стратегій та заходів реалізації. На додаток до підтримки академічної діяльності, цей тип навчання може також сприяти взаємодії та спілкування серед тих, хто бере участь у навчальному процесі.

M-Learning надає можливості об'єднати людей у реальному та віртуальному світах, створюючи навчальні спільноти серед викладачів та студентів. Це відбувається з метою інтеграції процесу викладання та навчання з використанням мобільних технологій. Тому виникає необхідність у створенні одного або декількох стратегій викладання для підтримки цього навчального процесу або можливого комплексу освітніх заходів, які може застосовуватися відповідно до індивідуальних та / або колективних потреб студентів. Смартфони мають великий потенціал для навчання. Студенти все частіше використовують їх у повсякденному житті.

Рівень користувачів смартфона серед дорослих у 2019 році був 65%. Серед молоді (віком 18–29 років) вона становила 85%. 88% підлітків (у віці від 13 до 17) мали доступ до якогось мобільного телефону в 2019 році. Викладачі можуть скористатися наявністю смартфонів для створення інтерактивного та цікавого досвіду навчання. Використовуючи спеціальні особливості телефонів, вчитель може створити новий досвід навчання і залучити студентів до нього в класі та поза ним, а отже, підвищити мотивацію навчання серед учнів. Смартфони можуть збагачувати навчання, забезпечуючи автентичні та контекстні умови навчання.

Навчання за допомогою мобільного пристрою можуть бути спонтанними та вимагати певні умови. Він пропонує нові можливості для навчання: навчання поза класом, навчання в будь-який час і в будь-якому

					КНТЕУ 121-02-22.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		6

місці та навчання на ходу. Єдиними обмеженнями, що обмежують використання мобільного телефону, є погані умови зв'язку, оскільки зв'язок все ще неможливий у певних місцях та тривалість роботи акумулятора. Організація мобільного навчання це нова технологія в освіті. Відповідно до результатів дослідження, ми можемо вказати на основні вимоги до організації мобільного навчання в системі освіти:

- Комунікація: можливість спілкування за допомогою різноманітних програмних засобів: електронної пошти, форумів, відеоконференцій;
- Співпраця: мобільні засоби навчання та програмне забезпечення повинні бути ергономічними та простими у використанні;
- Створення: Інструменти та послуги електронного навчання повинні відповідати творчим потребам студентів, таких як програмування, редагування музики, відео, зображень та тексту;
- Зміст: наявність необхідного змісту для ефективного процесу викладання та навчання та цілісність дисципліни.

Найважливіші переваги впровадження M-learning у вищому освітньому процесі

- Повсюдність: можливість отримання інформації в Інтернеті;
- Доступність: доступ до персональних та навчальних ресурсів через бездротову мережу;
- Зручність: можливість зберігати персональні дані та необхідні навчальні матеріали на мобільному пристрої, миттєве підключення до Інтернету через мобільний телефон;
- Персоналізація: можливість персоналізованого навчання.

В останні роки особлива увага приділяється концепції, названій як BYOD («Принесіть свій пристрій»), який тісно пов'язаний з технологією мобільного навчання. Впровадження BYOD у навчальний процес передбачає впровадження в університетське освітнє середовище технологій та послуг,

					КНТЕУ 121-02-22.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		7

орієнтованих на надання персоналізований доступ до інформаційних ресурсів через мобільні пристрої (навчальні програми та програми, звіти про успіхи та відвідуваність, результати сесії, навантаження, графік для студента та вчителя тощо); надання розподіленого онлайн-доступу до вмісту (трансляція підкастів, вебінари, електронні журнали, особиста бібліотека освітніх та наукових ресурсів, соціальні медіа тощо). Успішна реалізація BYOD як складової ІТ-стратегії вищого навчального закладу зумовлена наступними факторами:

- високий рівень та динаміка поширення мобільних пристроїв у студентському та навчальному середовищі та стійкий інтерес до їх застосування, сформований зовнішніми соціально-психологічними факторами;

- Значний когнітивний потенціал студентів вищого навчального закладу, які є досить гнучкими та адекватно реагують на зміни в усталеній практиці організації навчального процесу та легко адаптуються до нових підходів та технологій;

- Навчальні матеріали порівняно легко можна перетворити на медіа-вміст та вміст для інтерактивних мобільних послуг;

–Мобільні послуги та контент можуть бути досить легко інтегровані в інфраструктуру освітнього та дослідницького простору як технологічно, так і методологічно.

Мобільні пристрої показали свій вплив у теперішньому та майбутньому наступним чином:

1) Форм-фактор: простий в користуванні, якщо порівнювати з ноутбуками та ноутбуками.

2) Тривалий час автономної роботи та миттєве увімкнення: триває довше без підключення до мережі та забезпечує миттєве підключення.

3) Ціна: коштує вдвічі менше і різко зменшується.

						КНТЕУ 121-02-22.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			8

- 4) Сенсорний інтерфейс: немає незграбнішої миші або тачпада.
- 5) Покращене цифрове читання: чітка якість дисплея.
- 6) Вбудована мультимедіа: аудіо / відео, зображення.
- 7) Співпраця: освітнє значення соціальних мереж.

У наш час Інтернет став частиною життя кожного студента, на якому потрібно шукати інформацію будь-коли і використання мобільних телефонів для доступу до Інтернету стало звичним явищем. Кількість мобільних доступних споживачів до Інтернету збільшується з кожним днем. Зростання попиту на телефон Android доступ до Інтернету з високошвидкісним переглядом дав можливість забезпечити можливості в освітньому секторі. Які надають можливість кінцевим користувачам використовувати свій смартфон для отримання освітніх переваг в обмежений час, незалежно від їх місцезнаходження. Дистанційна освіта - це механізм навчання, який зосереджується на звільненні учнів від обмеження часу та місця для гнучких можливостей в освіті. Це також дозволяє студентам використовувати свій час таким чином, щоб вони могли продовжувати навчання без шкоди для їх роботи та сімейного життя. Смартфон у класі та поза ним полегшує роботу студентів та викладачів до співпраці. Студенти можуть пропустити лекції з будь-яких причин та відвідувати заняття через свій смартфон і вести запис пропущеного заняття без жодних проблем щоб тримати їх окремо.

Поряд із перевагами, він має деякі обмеження, наприклад, смартфон надає доступ до іншої діяльності такі як Whatsapp, сайти соціальних мереж, електронні листи, онлайн-ігри і телевізійні канали. Це може ввести студента в оману, зосередитися на інших перспективах, що призводять до порушень інших учнів навколо них і навіть час від часу весь клас. Крім того, студентам можуть просто дзвонити під час іспитів, щоб обдурити, або студентам у класі чи екзаменаційному залі може бути легко використовувати свої смартфони для доступу до інформації в Інтернеті, щоб обдурити на іспитах. Насправді є

					КНТЕУ 121-02-22.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

теми про використання смартфонів для обману в класі. Залежність від смартфона може бути використана для повідомлень з іншими студентами, пошуку рішення в Інтернеті, використання наукових калькуляторів та різних програм, відповіді також зберігаються на їхніх телефонах, щоб допомогти в іспиті. Смартфони також можуть застосовувати до переслідування та загрози. В смартфон вбудований високо профільні аудіо / відео системи, які можна використовувати для запису та фотографування домагань у школи та коледжі.

1.2. Поняття та принципи навчання протягом життя (LifeLongLearning)

Осліплюючі зміни в науці і техніці сьогодні призвели до швидкого збільшення обсягу інформації і призвели до того, що певна інформація про сайт втратила свою актуальність за досить короткий час. Тепер не достатньо покластись на обмежене джерело інформації та намагатися застосувати цю інформацію в майбутньому у цьому постійно мінливому світі. Для того, щоб йти в ногу з цією зміною, важливо навчитися слідкувати за інформацією та деякі навички. Сьогодні для дорослого - набуття правильного та ефективного досвіду навчання здається найважливішим скарбом, який він чи вона міг би принести з минулого життя на сьогоднішній день і принести на майбутнє. В додаток, існує загальноприйнята думка, що обов'язкові освітні заходи можуть бути недостатніми для забезпечення особи необхідною інформацією та навичками, які їй можуть знадобитися протягом усього життя та освіта що тривала протягом усього життя. З цієї причини одна з нових парадигм прийнята як у вищих навчальних закладів і, загалом, у всій освітній системі сьогодні - це «навчання впродовж життя». Навчання впродовж життя

					КНТЕУ 121-02-22.MP	Аркуш
						10
Зм.	Аркуш	№ докум	Підпис	Дата		

здійснюється різними міжнародними установами з цілісним підходом і має філософію "освіта від колиски до могили" та, має такі загальні цілі як такі: узагальнення дошкільної освіти як за кількістю, так і за якістю, що становить реальну базу навчання в обов'язковій освіті, сприяючи прогресу від школи до бізнесу, життя, заохочуючи дорослих до навчання, оновлюючи ресурси системи та встановлюючи узгодженість між частинами системи.

Швидкі зміни в інформаційних технологіях постійно змінюють суспільні структури.

Найбільше відображення змін у двадцять першому столітті, без сумніву, було в освіті з часу існування навчальних закладів полягає у вихованні людей, які мають високу кваліфікацію для нового віку. Поняття, що формують традиційну освіту, перевертаються в інформаційному суспільстві. В минулому, освіта означала офіційні навчальні заклади, які обслуговували людей протягом певного періоду часу; навпаки, в інформаційному віці дає можливість існування освітнього / навчального середовища незалежно від часу та місця. основною причиною такого середовища є оновлення наявної інформації та навичок, а отже потреба у навчанні протягом усього життя. Швидко розвиваються технології та інтенсивне збільшення обсягу знання значно сприяли глобалізації. Сьогодні, в умовах глобалізації, виробляється діюча інформація і його застосування до нових доменів було основною силою, яка визначає національна та міжнародна конкуренція. Інформація була основним джерелом багатства для країн. Виробництво та використання дійсної інформації, яка буде джерелом багатств, відповідно поклала нові обов'язки на освіту систему.

Найбільш очевидні переваги навчання впродовж життя - це освіта, підтримка потреб, цілей та інтересів людей протягом їхнього життя. У цьому сенсі навчання протягом усього життя не є нічим новим. Вивчення нового інструменту, нової мови, нової навички, як малювання, кулінарія чи письмо,

					КНТЕУ 121-02-22.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		11

ніколи не обмежувалось лише навчанням у дитинстві. Окрім цих освітніх переваг, сучасне навчання впродовж життя також пропонує безліч економічних стимулів.

Пристаювання до постійно мінливого ринку важливе. У наш час маркетологи повинні знати, як використовувати інструменти без коду, менеджери продуктів, щоб знати про UX, і підприємці ... Щоб знати трохи про все. Деякі робочі місця стають автоматизованими, а інші створюються. Розробник додатків, SEO-спеціаліст, виробник підкастів ... Це всі робочі місця, яких не було не так давно. Навчання протягом усього життя означає, що ви зможете адаптуватися до змін ринку, незалежно від того, стане ваша робота менш потрібною або зміниться в природі.

Творчість носить комбінаційний характер. Виставляючи більше знань у різних галузях, ми збільшуємо наші шанси виявити потребу або прогалину, яку слід заповнити. Знання - це справді сила, коли йдеться про висування нових ідей. Рідко буває добре, коли засновники стартапів намагаються придумати випадковий продукт, не знаючи багато про ринок, користувачів та технологічні обмеження.

Розвиток сусідніх навичок - це чудовий спосіб розширити свою кар'єру перспективу. Знову ж таки, дуже мало ймовірно, що ви будете залишатися на одній роботі всю свою кар'єру. Навчання впродовж життя - це спосіб зробити подорож цікавішою, відкривши нові -двері під час вашої кар'єри.

Загалом, будучи учнем протягом усього життя, ваш профіль стане більш привабливим для потенційних роботодавців, а ваші навички стануть потужнішими як фрілансер чи підприємець. Дизайнери, які вміють кодувати, маркетологи, які знають про психологію, та творці вмісту, які знають про управління проектами, швидше за все досягнуть своїх цілей.

					КНТЕУ 121-02-22.МР	Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата		

Нарешті, але не менш важливим є те, що навчання протягом усього життя може бути справжньою можливістю отримати радість через інтелектуальні виклики, заведення нових друзів та виховання своєї цікавості.

Хоча існує багато університетів, які пропонують безперервну освіту, навчання протягом усього життя не вимагає переходу від педагогічного (керованого інструкторами) до андрагогічного (керованого дорослими) підходу. Існує кілька кроків до навчання протягом усього життя:

- Визначення з поточною метою навчання. Навчальні цілі не обов'язково встановлювати, але, як правило, корисно зосередитись на одній чи двох цілях за один раз, особливо якщо ви зараз не є студентом денної форми навчання і маєте інші обов'язки у своєму житті. Крім того, ваші навчальні цілі можуть бути професійними чи особистими.
- Навчальні обмеження. Визначившись із ціллю навчання, стає можливим розпочати дослідження. Можливо, що навчальна тема є досить нішевою, і існує лише один онлайн-курс, який її охоплює. Можливо, це популярна тема, і буде безліч варіантів на вибір. Який ваш бюджет? Скільки часу ви можете виділити? Це всі важливі навчальні обмеження, які визначатимуть, як розроблятиметься навчальне середовище.
- Створення навчального середовища. Тепер можна зрозуміти, що доступно з точки зору змісту, бюджету та часу. Обраний курс, книга, репетитор або будь-який інший спосіб, для досягнення навчальної мети. Можливо комбінувати різні варіанти, виходячи з потреб та проблем. Обрання інструментів, такі як додаток для записів або планшет. Залежно від того, яка стратегія обрана для вивчення теми, можливо, доведеться отримати конкретні інструменти.

					КНТЕУ 121-02-22.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

Оскільки немає фіксованої навчальної програми, можливо змішувати та поєднувати різні засоби, щоб зробити це цікавим. Це також може бути більш спонукальним для навчання разом з іншими: навчальні групи, навчальні спільноти або навчання з другом. І якщо через деякий час тема буде не до вподоби або неактуальною, хороша новина: жоден учитель не змусить продовжувати. Абсолютно добре відмовитись від навчальної мети, коли вона вже не підходить. Ви можете просто обрати нову.

1.3. Висновки до Розділу 1

M-навчання вже почало відігравати дуже важливу роль у електронному навчанні у світі. M-навчання - це шлюз до електронного навчання для більшість тих, хто навчається в, оскільки стрімко зростаюча бездротова інфраструктура все більше задовольняє потреби у доступі. Дистанційна освіта невід'ємно пов'язана з технологіями в освіті. Мобільні телефони – це найпоширеніша технологія. Їх 1,5 мільярда на 6 мільярдів населення світу. Їх використання для навчання та навчання є надзвичайно важливим. Всі проблеми, пов'язані з представленням повних мобільних навчальних курсів на КПК, були вирішено. Не повинно виникати труднощів при розробці навчальних матеріалів для цих пристроїв та проведенні повних курсів з них. Розвиток мобільного навчання - це новий передовий рівень у розвитку, наступне покоління після електронного навчання.

Навчання впродовж життя - це безперервний процес, у якому люди зберігають і розвивають своє життя: поведінка, знання та вміння. Головною метою навчання впродовж життя є зміцнення та поліпшення якості життя людей даючи їм можливість довести власний потенціал до максимального рівня.

					КНТЕУ 121-02-22.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		14

РОЗДІЛ 2

ЗАСОБИ РОЗРОБКИ НАВЧАЛЬНОГО МОБІЛЬНОГО ДОДАТКУ ІНТЕГРАЦІЇ КЕРУВАННЯ ЗНАННЯМИ І БЕЗПЕРЕРВНИМ НАВЧАННЯМ

2.1. Опис та переваги ОС Android

Сьогодні існує низка мобільних операційних систем, які намагалися бути найкращими у світі, але жодна з них не зуміла піднятися на вершину та зробити такий вплив, як Android. Android - це набір програмних підсистем, необхідних для забезпечення повноцінного рішення для мобільних пристроїв. Обидві платформи містять подібність у способі використання та використовуваних пристроях, але є багато відмінностей, які виникають внаслідок різних операційних систем: Android. На ринку мобільних телефонів поява різних платформ для смартфонів, таких як Android, не змушує розробників постійно розробляти програми з однаковим вмістом, щоб забезпечити відповідність кожній з доступних платформ. Потім кожний додаток, розроблений на одній платформі для використання на іншій платформі, має бути перепрограмований. Як результат, для аналізу та перетворення вмісту мобільного додатка для використання платформи на смартфоні потрібен значний час та витрати. За допомогою автоматичного перетворення наявного вмісту програми в вміст іншої програми, наявний вміст програми можна швидко завантажити на різні платформи. Як

					КНТЕУ 121-02-22.МР			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Зав. кафедри</i>	Криворучко О.В.			24.05.2021	<i>Модель навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	Криворучко О.В.			24.05.2021		P2	15	49
<i>Гарант</i>	Токар В.В.			24.05.2021		Факультет інформаційних технологій, 2м курс, 2 група		
<i>Розроб.</i>	Шапочка Д.В.			24.05.2021				
					<i>Засоби розробки навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням</i>			

результат, їх використання збільшиться, а отже, зменшиться робоча сила, час і витрати. Для вирішення цієї проблеми вивчено автоматичне перетворення та перетворювачі, призначені для автоматичного перекладу вмісту Android та навпаки.

Android - це мобільна операційна система, розроблена Google, яка в першу чергу призначена для мобільних пристроїв, таких як смартфони. Операційна система використовує сенсорні входи, такі як перетягування, натискання та стискання для маніпулювання об'єктами відображення та віртуальною клавіатурою. У 2015 році Android мав найбільшу встановлену базу серед усіх мобільних операційних систем. У червні 2017 року Google Play Mobile App Store випустив понад 3 мільйони додатків для Android, а також завантажено понад 80 мільярдів програм. У 2017 році компанія Google виявила, що у них було понад 2 мільярди активних користувачів Android на місяць, на відміну від попереднього року, приблизно 1,5 мільярда активних користувачів. Починаючи з 2008 року і дотепер, компанія Android мала багато оновлень, які поступово вдосконалювали свою операційну систему, додаючи нові функції та виправляючи помилки в попередніх версіях. Кожна нова версія названа на честь десерту в алфавітному порядку: Cupcake 1.5; Donut 1.6; Eclair 2.0; Froyo 2.2; Gingerbread 2.3; Honeycomb 3.0; Ice Cream Sandwich 4.0; Jelly Bean 4.1; KitKat 4.4; Lollipop 5.0; Marshmallow 6.0; Nougat 7.0; Oreo 8.0.

Щодо аналізу, проведеного в різних категоріях, можна зробити висновок, що в розділі інтерфейсів переконливо переважає Android, завдяки своєму більш уніфікованому, стильному та зручному інтерфейсу, тоді як з точки зору користувальницького досвіду Android надається невелика перевага завдяки величезному набору вибору меню, простоти їх використання та ідеальної клієнтської клавіатури QWERTY. У розділі доступності додатків переважає Android завдяки простішому інтерфейсу,

					КНТЕУ 121-02-22.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

вищому придбанню дорогих додатків, а отже, більшій прибутковості та більшій кількості рекомендацій щодо нових додатків. Після розділу стабільності операційної системи та дослідження, проведеного в першому кварталі 2020 року, було встановлено, що 68% пристроїв iOS зазнали збою щодо кількості проблем, відмови додатків, перегріву пристрою та падіння якості підключення тощо, тоді як для пристроїв Android цей показник становить 50%. В ході дослідження результати показали, що додатки для Android викликають помилку на 18% менше ніж iOS, тоді як Samsung, Xiaomi, Motorola, Sony і Lenovo згадуються як найбільш стандартні виробники Android. В якості остаточного висновку про стабільність систем можна сказати, що Android є більш серйозною і стабільною системою з точки зору системи відносно його аналогу iOS.

Що стосується оновлення програмного забезпечення, платформа регулярно та своєчасно оновлюються, і існує великий контроль над програмним забезпеченням. Але на Android, від мобільного оператора користувача, який тестує нові версії, залежить, коли нові оновлення будуть надані. Пристрої Android мають різні розміри батарей але кожен пристрій має змінну батарею і, таким чином, збільшує використання пристрою Android. Нові моделі Android, за винятком стандартної дротової зарядки акумулятора, також включають бездротову зарядку, яка повільніша за стандартну, але надає велику зручність користувачам. У цьому розділі Android має велику перевагу.

Аналіз, зроблений з точки зору периферійних пристроїв з уподобаннями Android і таким чином має більшу кількість периферійних пристроїв. Що стосується маршрутизації, запуску операційної системи та захисту, Android перемагає завдяки можливості маршрутизації мобільного пристрою. Завдяки маршрутизації мобільний пристрій має доступ до декількох програм і доступний для останніх оновлень системи, більших естетичних удосконалень

					КНТЕУ 121-02-22.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

операційної системи, збільшення швидкості роботи пристрою, збільшення часу автономної роботи та багатьох інших. У налаштуваннях Android є багато різних елементів, тож користувач може багато в чому налаштувати свій досвід, є можливість встановити різні пускові установки для зміни зовнішнього вигляду інтерфейсу користувача, можливість налаштувати більше фонів у фоновому режимі та ярлики, а також можливість вибрати клавіатуру за замовчуванням, підтримує Bluetooth та Wi-Fi. Android використовує NFC, тобто бездротовий зв'язок у короткому полі. Вибираючи пристрої, Android наймовірніше лідирує завдяки величезному різноманіттю мобільних пристроїв. Основна функціональність дзвінків та обміну повідомленнями ефективна на платформах. Android пропонує Hangouts, Messenger, Allo та Duo, за допомогою яких користувачі можуть надсилати повідомлення через Інтернет або SMS. Що стосується стандартних електронних програм, платформи використовують ряд сторонніх електронних програм. Однак програма Gmail на Android є більш популярною та потужною, і вона пропонує користувачеві можливість додати більше адрес електронної пошти від різних постачальників. Платформа використовує власний веб-браузер, Google Chrome на Android. І Android, пропонує хороші карти та точні оцінки, Google Maps є неперевершеною. Він містить найбільше пунктів і є найбільш детальним. Безпека на платформах знаходиться на задовільному рівні. У ряді пристроїв Android використовується програмне забезпечення старіше 3 років, і тому воно піддається серйозному злому, що загрожує безпеці. І нарешті, з точки зору створення та публікації додатків, Android та Play Store перемагають завдяки більшій свободі та дешевшому внеску за реєстрацію.

Що стосується дизайну, Android має більшу точність, чіткість і природність. З точки зору програмування додатків, програмування Android означало написання програм, які будуть використовуватися на великій

					КНТЕУ 121-02-22.МР	Аркуш
						18
Зм.	Аркуш	№ докум	Підпис	Дата		

кількості пристроїв Android, з різними розмірами екрану та різним співвідношенням. З цих причин існує фрагментація версій Android, і через це написання програм для Android дуже складне для розробників.

Завдяки аналізу обох платформ Android та iOS як переможця у цій "битві", Android виходить з дуже великою різницею. Android є більш вигідною операційною системою завдяки своїм перемогам у дизайні, безпеці, індивідуальності налаштувань, різноманіттю програм.

2.2. Переваги використання мови програмування Java

Java - це загальна, надійна, безпечна та об'єктно-орієнтована мова програмування. Це мова високого рівня, тобто її синтаксис використовує англійську мову. Він був розроблений Sun Microsystems у 1995 році. Зараз він підтримується та розповсюджується Oracle. Java має своє середовище виконання та API тому його ще називають платформою.

Протягом багатьох років Java використовується у великій кількості програм. Однак він має різні переваги та недоліки.

Переваги:

1. Простий. Java - це проста мова програмування, оскільки її легко вивчити та зрозуміти. Його синтаксис базується на C ++, і він використовує автоматичний збір сміття; отже, нам не потрібно видаляти об'єкти без посилань із пам'яті. Java також видала такі функції, як явні вказівники, перевантаження оператора тощо, що полегшує читання та запис.

2. Об'єктно-орієнтований. Java використовує об'єктно-орієнтовану парадигму, що робить її більш практичною. Все в Java - це об'єкт, який опікується як даними, так і поведінкою. Java використовує об'єктно-

						КНТЕУ 121-02-22.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			19

орієнтовані концепції, такі як об'єкт, клас, успадкування, інкапсуляція, поліморфізм та абстракція.

3. Забезпечений. Java є захищеною мовою програмування, оскільки вона не використовує явні вказівники. Крім того, програми Java працюють всередині пісочниці віртуальної машини. JRE також надає завантажувач класів, який використовується для динамічного завантаження класу в JVM. Він відокремлює пакети класів локальної файлової системи від тих, які імпортуються з мережі.

4. Надійний. Java - це надійна мова програмування, оскільки вона використовує потужне управління пам'яттю. Також можна обробляти винятки за допомогою коду Java. Крім того, можна використовувати перевірку типу, щоб зробити наш код більш безпечним. Він не надає явних покажчиків, щоб програміст не міг отримати доступ до пам'яті безпосередньо з коду.

5. Незалежна платформа. Код Java може працювати безпосередньо на декількох платформах, тобто нам не потрібно кожного разу компілювати його. Це правило одного разу, працює в будь-якій мові (WORA), яка може бути перетворена в байт-код під час компіляції. Байт-код - це незалежний від платформи код, який може працювати на декількох платформах.

6. Багатопоточність. Java використовує багатопотокове середовище, в якому більші завдання можна перетворити на різні потоки і запустити окремо. Головною перевагою багатопоточності є те, що нам не потрібно надавати пам'ять кожному поточному потоку.

Недоліки:

1. Продуктивність. Java повинна інтерпретуватися під час виконання, що дозволяє їй працювати в будь-якій операційній системі, але це також робить її повільнішою, ніж такі мови, як C та C ++. З іншого боку, програму C ++

						КНТЕУ 121-02-22.МР	Аркуш
							20
Зм.	Аркуш	№ докум	Підпис	Дата			

потрібно компілювати в кожній операційній системі, безпосередньо в бінарну, а отже, працює швидше.

2. Споживання пам'яті. Програма Java споживає більше пам'яті, оскільки працює на віртуальній машині Java.

3. Вартість. Мова програмування Java трохи дорожча через вищі вимоги до обробки та пам'яті. Нам потрібно краще апаратне забезпечення для запуску програми Java.

4. Менш інтерактивна машина. Java бракує безпосередньої взаємодії з машинами, що робить її менш життєздатною для програмного забезпечення, яке потрібно швидко запускати та запускати безпосередньо з машиною, оскільки явні вказівники також відсутні в Java.

5. Видалення сміття. Java забезпечує автоматичний збір сміття, яким не може керувати програміст. Він не надає таких методів, як `delete ()` та `free ()`, щоб звільнити пам'ять.

Однак не дивлячись через різні недоліки, Java є однією з найбільш використовуваних мов програмування завдяки своїм перевагам, що робить її незалежною від платформи, безпечною та надійною мовою програмування.

Мобільні пристрої, такі як ноутбуки, особисті цифрові помічники та мобільні телефони, стали інструментом навчання з великим потенціалом як у класах, так і на відкритому повітрі. У цьому контексті використання мобільних пристроїв у навчальних середовищах можна розглядати як фактор, що може підвищити ефективність процесу навчання та навчання.

У цьому процесі передбачається розробляти середовища мобільного навчання, щоб успішно досягти освітньої інтеграції мобільних технологій навчання та мобільних додатків. Однак розробка програмного забезпечення для мобільних додатків є слабкою, і методології, які вважаються доречними для прогресу таких мобільних додатків, недостатньо. Все ще бракує методів дослідження, і в той же час недостатньо розуміння та аналізу проблем та

					КНТЕУ 121-02-22.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		21

труднощів, які можуть виникнути в процесі розробки мобільних додатків. Є кроки, яких слід дотримуватися при розробці додатків.

Розробка мобільних додатків для Android за допомогою Java та Eclipse включають такі чотири стадії розробки:

1. Налаштування: На цьому етапі ми встановлюємо та налаштовуємо середовище розробки. Ми також створюємо віртуальні пристрої Android (AVD) та підключаємо апаратні пристрої, на які ми можемо встановлювати наші програми.

2. Розробка: На цьому етапі ми створюємо та розробляємо наш проєкт Android, який містить усі вихідні коди та файли ресурсів для нашої програми.

3. Налаштування та тестування: На цьому етапі ви вбудовуєте свій проєкт у пакет debuggable.apk, який можна встановити та запустити на емуляторі або пристрої на базі Android.

4. Публікація: На цьому етапі ми налаштовуємо та створюємо нашу програму для випуску та розповсюдження нашої програми серед користувачів.

В Android SDK було використано ряд інструментів. Після завантаження та встановлення SDK доступ до цих інструментів здійснюється безпосередньо з IDE Eclipse, через плагін ADT або з командного рядка. Розробка за допомогою Eclipse є найкращим методом, оскільки ми можемо безпосередньо шукати необхідні інструменти при розробці програм.

У міру зростання інтересу до мобільних додатків розробка мобільних додатків стала складним процесом. Процес розробки програмного забезпечення розроблених додатків має велике значення. Для контролю цих процесів та отримання найкращих результатів необхідно детально пояснити етапи розробки додатків. Переважна більшість досліджень у цій галузі дають результати функціонування процесів розробки програмного забезпечення. Заслуговує на увагу відсутність досліджень, що описують, як розробляються

						КНТЕУ 121-02-22.МР	Аркуш
							22
Зм.	Аркуш	№ докум	Підпис	Дата			

ці програми. Однак відомо, що розробка програмного забезпечення для мобільних додатків є слабкою, і прийнятих тверджень про прогрес мобільних додатків недостатньо. Необхідно звернути увагу на той факт, що методів дослідження все ще бракує, і одночасно визначити проблеми та труднощі, які можуть виникнути в процесі розробки мобільних додатків. Оскільки розробка програмного забезпечення зазвичай дає абстрактні пропозиції, необхідно відображати процес розробки програмного забезпечення таким, яким він є. Під час розробки програми описуються кроки, яких слід дотримуватися, та інструменти, які потрібно завантажити, згадуються один за одним.

Платформа Eclipse, Android ADT та Android SDK можуть бути використані для розробки мобільних додатків для операційної системи Android. Ці системи мають приємний інтерфейс і безперебійну роботу. Java на Eclipse IDE зараз є кращим вибором для розробки програмного забезпечення. Дослідники програмної інженерії розробили програми для різних цілей у цьому середовищі та продовжили їх аналіз. Ці програми забезпечують чудовий досвід користування. Окрім популярності в операційній системі Android, додатки на базі Android SDK привернули більше уваги. У цій роботі в Eclipse було використано ряд інструментів під час розробки програми для Android. По-перше, використовуються інструменти в Android SDK. Після завантаження та встановлення SDK доступ до цих інструментів здійснюється безпосередньо з IDE Eclipse, через плагін ADT або з командного рядка. Після встановлення Eclipse ADT налаштування каталогу Android SDK та налаштування Android SDK та AVD manager. За допомогою налаштування віртуального пристрою для Android програма відображається на віртуальних пристроях.

					КНТЕУ 121-02-22.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		23

2.3. Опис та обґрунтування вибору бібліотеки FBReader SDK.

В даний час найбільш популярними та зручними способами читання є електронні книги. Їх можна взяти із собою куди завгодно, вони заощаджують папір, відображення тексту в них можна налаштовувати так, як зручно читачеві. З цих причин з'явилося багато форматів електронних книг, основні з яких це EPUB, FB2 та MOBI. Всім відомі DOC і TXT також є форматами електронних книг, проте їх можливості порівняно з вищезазначеними дуже обмежені. У кожного з форматів є свої переваги, проте найчастіше використовуються EPUB і FB2, причому другий популярний тільки на території СНГ. Однією з особливостей цих форматів є те, що їм потрібна спеціальна програма, яка може працювати з конкретним форматом. Тому в App Store або Google Play можна знайти багато різноманітних читалок під всі відомі формати.

У вересні минулого року FBReader випустили свій власний SDK для Android, що дозволяє створювати власні читалки на його основі. FBReader SDK має великий функціонал, який включає наступні можливості:

- Відкриття будь-яких файлів електронних книг (ePub, fb2, mobi тощо)
- Читання метаданих із файлу книги
- Особливості движка FBReader
- Можливість змінювати стиль тексту книги

Крім того, бібліотека надає різні додаткові можливості для реалізації в додатках, такі як:

- Виділення тексту;
- Закладки;
- Пошук за текстом;

						КНТЕУ 121-02-22.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			24

- Зміст;
- Додаткова навігація за текстом;
- Підтримка виносок та гіперпосилань;
- Налаштування яскравості.

Окремо варто відзначити, що, хоч FBReader SDK і можна завантажити з сайту безкоштовно, після вбудовування в додаток вона працюватиме в демо-режимі, який має повний функціонал, але показує лише кілька перших сторінок книги. Для того, щоб отримати повну версію бібліотеки, потрібно купити її на сайті розробника і, дотримуючись інструкцій, отримати спеціальний ключ SDK, який потім додається до програми.

2.4. Висновки до Розділу 2

Android - це мобільна операційна система, розроблена Google, яка в першу чергу призначена для мобільних пристроїв. Android є більш вигідною операційною системою завдяки своїм перемогам у дизайні, безпеці, індивідуальності налаштувань, різноманіттю програм та користувацькій екосистем. Java, платформа Eclipse, Android ADT та Android SDK доступні з найосновніших мов програмування для розробки мобільних додатків в операційній системі Android. Ці системи мають приємний інтерфейс і безперебійну роботу. Ці програми забезпечують чудовий досвід користування. Окрім популярності в операційній системі Android, додатки на базі Android SDK також привертають більше уваги. В останні роки набуває популярності тенденція електронних книг, як засобу до саморозвитку та навчальних цілей. Завдяки зручності використання такого формату книг з'являється все більше користувачів. Тому дана бібліотека підходить під критерії цього проєкту.

						КНТЕУ 121-02-22.МР	Аркуш
							25
Зм.	Аркуш	№ докум	Підпис	Дата			

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1. Створення проєкту та підключення FBReader SDK

У цьому розділі описано, як на основі FBReader SDK створити додаток електронних книг і буде розглянуто його можливості.

Почнемо з додавання SDK до свого проєкту. Для цього створимо новий проєкт із порожньою активністю. Як мінімальний API вкажемо API 16.

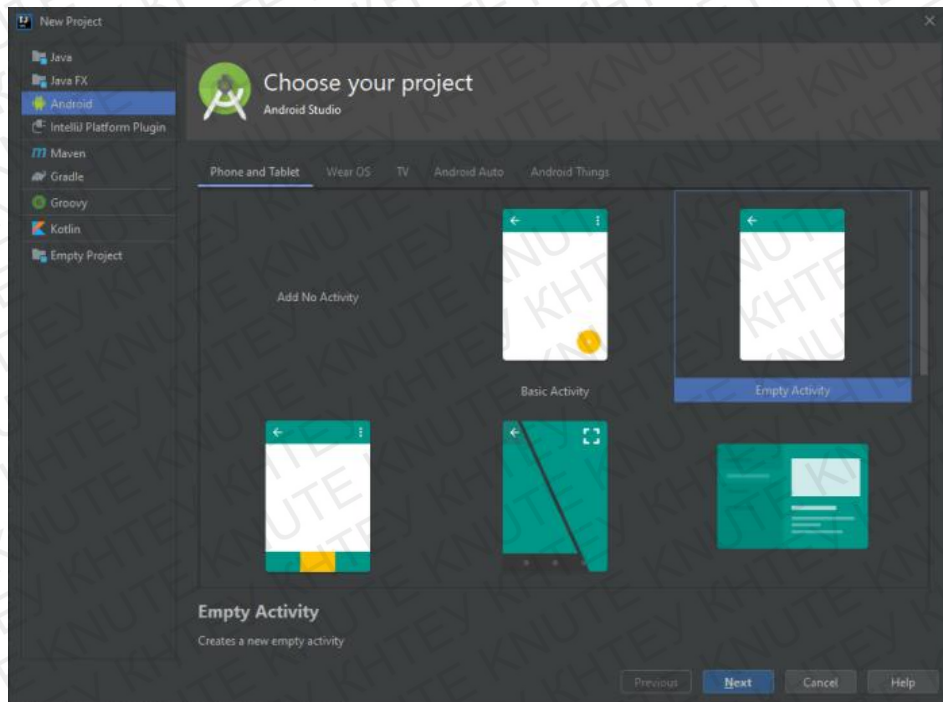


Рис. 3.1. Меню створення проєкту в Android studio.

У файлі `build.gradle` модуля проєкту додамо до блоку всіх проєктів посилання для завантаження потрібних нам бібліотек.

Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121-02-22.МР</i>			
Зав. кафедри		Криворучко О.В.		21.06.2021	Модель навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		21.06.2021		РЗ	26	49
Гарант		Токар В.В.		21.06.2021		Факультет інформаційних технологій, 2м курс, 2 група		
Розроб.		Шапочка Д.В.		21.06.2021				
					Розробка програмного продукту			


```

allprojects {
    repositories {
        ...
        maven {
            url "https://sdk.fbreader.org/maven"
        }
    }
}

```

Потім у файлі build.gradle модуля програми додамо залежності з бібліотеками до блоку dependencies.

```

dependencies {
    ...

    implementation 'com.googlecode.json-simple:json-simple:1.1'
    implementation 'org.fbreader:book:1.0.0-rc01'
    implementation 'org.fbreader:config_client:1.0.0-rc01'
    implementation 'org.fbreader:config_provider:1.0.0-rc01'
    implementation 'org.fbreader:filesystem:1.0.0-rc01'
    implementation 'org.fbreader:fontentry:1.0.0-rc01'
    implementation 'org.fbreader:format_interface:1.0.0-rc01'
    implementation 'org.fbreader:image:1.0.0-rc01'
    implementation 'org.fbreader:intent:1.0.0-rc01'
    implementation 'org.fbreader:language:1.0.0-rc01'
    implementation 'org.fbreader:loader:1.0.0-rc01'
    implementation 'org.fbreader:options:1.0.0-rc01'
    implementation 'org.fbreader:text_client:1.0.0-rc01'
    implementation 'org.fbreader:text_provider:1.0.0-rc01'
    implementation 'org.fbreader:text_util:1.0.0-rc01'
    implementation 'org.fbreader:text_view:1.0.0-rc01'
    implementation 'org.fbreader:toc:1.0.0-rc01'
    implementation 'org.fbreader:util:1.0.0-rc01'
    implementation 'org.fbreader:view:1.0.0-rc01'
    implementation 'org.fbreader:vingadgets:1.0.0-rc01'
    implementation 'org.fbreader:zip-amse:1.0.0-rc01'
}

```

Як можна помітити, тут також є бібліотека «com.googlecode.json-simple:json-simple:1.1». Вона стане в нагоді трохи пізніше для роботи з JSON-файлами.

					<i>КНТЕУ 121-02-22.МР</i>	<i>Аркуш</i>
						27
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

Бібліотек, що входять до FBReader SDK, досить багато, однак вони розділені так, щоб можна було підключати тільки ті, які потрібні для конкретного випадку.

У цьому ж файлі до блоку android додамо наступні рядки, які мають забезпечити успішне складання проєкту.

```
android {  
    ...  
  
    compileOptions {  
        sourceCompatibility JavaVersion.VERSION_1_8  
        targetCompatibility JavaVersion.VERSION_1_8  
    }  
}
```

На сайті SDK розробники також надають код різних компонентів, які спрощують реалізацію інтерфейсу. Для потрібно скачати та перенести звідти папку fbreader до папки свого проєкту. Після цього у файлі settings.gradle потрібно прописати цю папку та шляхи до потрібних модулів.

```
include 'fbreader_extras'  
  
project(':fbreader_extras').projectDir = new File('fbreader/extras')  
  
include 'fbreader_styles'  
  
project(':fbreader_styles').projectDir = new File('fbreader/styles')  
  
include 'fbreader_text_extras'  
  
project(':fbreader_text_extras').projectDir = new File('fbreader/text_extras')
```

Тепер додати залежності з ними у файлі build.gradle модуля програми.

					КНТЕУ 121-02-22.МР	<i>Аркуш</i>
						28
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

```
dependencies {
    ...

    implementation project(':fbreader_extras')
    implementation project(':fbreader_styles')
    implementation project(':fbreader_text_extras')
}
```

Коли IDE перезбере проект, можна побачити додані модулі у списку ліворуч.

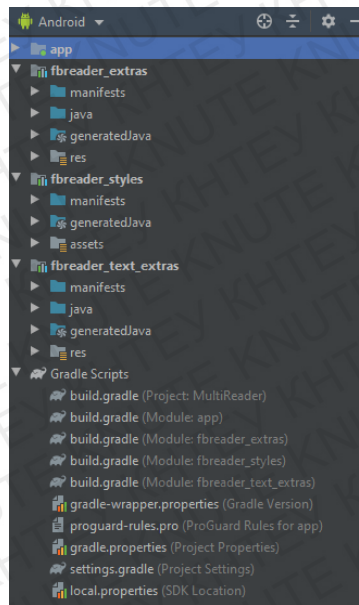


Рис 3.2. Список модулів додатку

На цьому підключення SDK завершено і можна розпочати безпосередньо створення програми.

3.2. Написання додатку

Має бути проект з двома активностями, одна з яких відображатиме список електронних книг на пристрої, а друга відкриватиме вибраний файл.

					<i>КНТЕУ 121-02-22.MP</i>	<i>Аркуш</i>
						29
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

Почнемо з розмітки. Для MainActivity додамо на екран компонент RecyclerView, який міститиме власне список файлів. Для цього додамо до файлу layout/activity_main.xml наступний код.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:paddingRight="12dp"
    android:paddingLeft="12dp"
    >

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="6dp"
        android:id="@+id/rv_files"
        />

</RelativeLayout>
```

					КНТЕУ 121-02-22.МР	Аркуш
						30
Зм.	Аркуш	№ докум	Підпис	Дата		

Тепер ініціалізуємо цей об'єкт у кодї активності. Відкриємо файл MainActivity.java та додамо наступний код.

```
public class MainActivity extends AppCompatActivity {  
  
    private RecyclerView rvFiles;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        rvFiles = findViewById(R.id.rv_files);  
        rvFiles.setLayoutManager(new LinearLayoutManager(this));  
    }  
}
```

Для того, щоб працювати з файлами, що зберігаються на пристрої, нам потрібно буде запитувати спеціальну роздільну здатність. Додамо до маніфесту програми AndroidManifest.xml такі дозволи.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
  
    package="ru.androidtools.multireader">  
  
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>  
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
  
    <application  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="@string/app_name"  
        android:roundIcon="@mipmap/ic_launcher_round"  
        android:supportsRtl="true"  
        android:theme="@style/AppTheme">  
        <activity android:name=".MainActivity">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN"/>  
  
                <category android:name="android.intent.category.LAUNCHER"/>  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>
```

								Аркуш
								31
Зм.	Аркуш	№ докум	Підпис	Дата				

KHTEY 121-02-22.MP

Починаючи з API 23, цей дозвіл необхідно запитувати у користувача в рантаймі. Для цього при запуску програми перевірять версію Android на пристрої і при необхідності відправляти користувачеві запит на отримання дозволу.

```
public class MainActivity extends AppCompatActivity {  
    ...  
  
    private static final int REQUEST_PERMISSION = 101;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        rvFiles = findViewById(R.id.rv_files);  
        rvFiles.setLayoutManager(new LinearLayoutManager(this));  
  
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
            checkPermission();  
        } else {  
            generateList();  
        }  
    }  
  
    @TargetApi(Build.VERSION_CODES.M) private void checkPermission() {  
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE_EXTERNAL_STORAGE)  
            == PackageManager.PERMISSION_GRANTED) {  
            generateList();  
        } else {  
            ActivityCompat.requestPermissions(this,  
                new String[] { Manifest.permission.WRITE_EXTERNAL_STORAGE }, REQUEST_PERMISSION);  
        }  
    }  
}
```

Отримати результат запиту можна, перевизначивши метод активності `onRequestPermissionsResult()`. Якщо дозвіл додатку надано, продовжуємо подальшу роботу.

```
@Override public void onRequestPermissionsResult(int requestCode, @NonNull String[]  
permissions,  
  
    @NonNull int[] grantResults) {  
  
    switch (requestCode) {
```

						Аркуш
						32
Зм.	Аркуш	№ докум	Підпис	Дата	KHTEY 121-02-22.MP	

```

case REQUEST_PERMISSION: {

    if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

        generateList();

    }

    break;

}

}
}

```

У методі generateList() ми будемо шукати потрібні файли на пристрої та відображати їх у списку програми. Для цього створимо AsyncTask, який виконуватиме всю роботу в окремому потоці та повертатиме готовий список із знайденими файлами. Додамо наступний код до MainActivity.

```

private void generateList() {
    List<StorageBean> storageBeans = StorageUtils.getStorageData(this); // получение списка
    смонтированных карт памяти на устройстве
    List<String> paths = new ArrayList<>();
    if (storageBeans != null) {

        for (StorageBean storageBean : storageBeans) {
            paths.add(storageBean.getPath());
        }
    } else {
        String path = Environment.getExternalStorageDirectory().getAbsolutePath();
        paths.add(path);
    }
    ListFilesTask listFilesTask = new ListFilesTask(paths);
    listFilesTask.setListener(new ListFilesTask.ListFilesListener() {
        @Override public void onTaskCompleted(List<File> files) {
        }
    });
    listFilesTask.execute();
}

static class ListFilesTask extends AsyncTask<Void, Void, List<File>> {
    public interface ListFilesListener {
        void onTaskCompleted(List<File> files);
    }

    private ListFilesListener listener;
    private List<String> startPaths;
    private List<File> files;
    private boolean completed;
}

```

						<i>Аркуш</i>
						<i>KHTEY 121-02-22.MP</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		33

```

public ListFilesTask(List<String> startPaths) {
    this.startPaths = new ArrayList<>(startPaths);
    this.files = new ArrayList<>();
    this.completed = false;
}

public void setListener(ListFilesListener listener) {
    this.listener = listener;
    if (completed && listener != null && files != null) {
        listener.onTaskCompleted(files);
    }
}

@Override protected List<File> doInBackground(Void... voids) {
    List<File> fileList = new ArrayList<>();
    for (String s : startPaths) {
        searchFiles(fileList, new File(s));
    }
    return fileList;
}

@Override protected void onPostExecute(List<File> files) {
    completed = true;
    if (listener != null) {
        listener.onTaskCompleted(files);
    } else {
        this.files = new ArrayList<>(files);
    }
}

private void searchFiles(List<File> list, File dir) {
    String epubPattern = ".epub";
    String fb2Pattern = ".fb2";

    File[] listFiles = dir.listFiles();
    if (listFiles != null) {
        for (File listFile : listFiles) {
            if (listFile.isDirectory()) {
                searchFiles(list, listFile);
            } else {
                if (listFile.getName().endsWith(epubPattern) || listFile.getName().endsWith(fb2Pattern)) {
                    list.add(listFile);
                }
            }
        }
    }
}
}

```

Отриманий результат слід відобразити. Для цього нам знадобиться адаптер для RecyclerView та модель, в якій зберігатимуться дані елементів списку. Створимо простий клас BookFile, в якому ми зберігатимемо ім'я файлу і шлях до нього на пристрої.

```

public class BookFile {
    private String filename, path;
}

```

						<i>Аркуш</i>
						КНТЕУ 121-02-22.МР
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		34


```

public BookFile(String filename, String path) {
    this.filename = filename;
    this.path = path;
}

public String getFilename() {
    return filename;
}

public String getPath() {
    return path;
}

@Override public int hashCode() {
    return filename.hashCode() + path.hashCode();
}

@Override public boolean equals(Object obj) {
    if (obj instanceof BookFile) {
        BookFile bookFile = (BookFile) obj;
        return this.filename.equals(bookFile.getFilename()) &&
this.path.equals(bookFile.getPath());
    }
    return false;
}

```

Тепер нам потрібен адаптер, до якого ми завантажуватимемо список. Додамо розмітку для списку, для цього створимо файл layout/file_list_item.xml і додамо в нього наступний код.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/item_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:layout_marginTop="2dp"
    android:orientation="vertical"
    android:paddingBottom="6dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="6dp"
    >
    <TextView
        android:id="@+id/tv_filename"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="filename"
        android:textSize="16sp"
        />
    <TextView
        android:id="@+id/tv_path"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="2dp"

```

						<i>Аркуш</i>
						КНТЕУ 121-02-22.МР
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		35

```

        android:text="path"
        android:textSize="14sp"
    />
</LinearLayout>

```

Тепер створимо клас BooksAdapter, який успадковуватиме від RecyclerView.Adapter<>. У ньому нам потрібно ініціалізувати елементи списку, завантажуючи в них шляхи та імена файлів електронних книг. Таким чином, код адаптера буде виглядати так.

```

public class BooksAdapter extends RecyclerView.Adapter<BooksAdapter.BookHolder> {
    public interface BookListener {
        void onBookOpen(BookFile bookFile);
    }

    private BookListener listener;
    private List<BookFile> books;

    public BooksAdapter(List<BookFile> books, BookListener listener) {
        this.books = new ArrayList<>(books);
        this.listener = listener;
    }

    @NonNull @Override public BookHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View view =
            LayoutInflater.from(parent.getContext()).inflate(R.layout.file_list_item,
parent,
false);
        return new BookHolder(view);
    }

    @Override public void onBindViewHolder(@NonNull BookHolder holder, int position) {
        BookFile bookFile = books.get(position);
        holder.bind(bookFile);
    }

    @Override public int getItemCount() {
        return books.size();
    }

    class BookHolder extends RecyclerView.ViewHolder {
        private LinearLayout itemLayout;
        private TextView tvPath, tvName;

        public BookHolder(@NonNull View itemView) {
            super(itemView);
            itemLayout = itemView.findViewById(R.id.item_layout);
            tvName = itemView.findViewById(R.id.tv_filename);
            tvPath = itemView.findViewById(R.id.tv_path);
        }

        public void bind(BookFile bookFile) {
            tvPath.setText(bookFile.getPath());
            tvName.setText(bookFile.getFilename());
            itemLayout.setOnClickListener(new View.OnClickListener() {
                @Override public void onClick(View view) {
                    listener.onBookOpen(bookFile);
                }
            });
        }
    }
}

```

						<i>Аркуш</i>
						KHTEY 121-02-22.MP
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		36

```

    }
  });
}
}
}

```

Крім того, між активністю та адаптером прокидається інтерфейс, за допомогою якого ми будемо передавати в активність файл, який користувач хоче відкрити.

Повернемося до головної активності. У методі `generateList()` ми запускаємо `AsyncTask` і прив'язуємо до нього слухач, який повинен викликати метод `onTaskCompleted()` при завершенні роботи. У цьому методі нам потрібно скласти список, відсортувати його на ім'я та передати в адаптер. Додамо сюди наступний код:

```

listFilesTask.setListener(new ListFilesTask.ListFilesListener() {
    @Override public void onTaskCompleted(List<File> files) {
        if (!isFinishing()) {
            List<BookFile> bookFiles = new ArrayList<>();
            for (File f : files) {
                BookFile bookFile = new BookFile(f.getName(), f.getAbsolutePath());
                if (!bookFiles.contains(bookFile)) bookFiles.add(bookFile);
            }
            Collections.sort(bookFiles, new Comparator<BookFile>() {
                @Override public int compare(BookFile bookFile, BookFile t1) {
                    return bookFile.getFilename().compareToIgnoreCase(t1.getFilename());
                }
            });
            rvFiles.setAdapter(new BooksAdapter(bookFiles, new BooksAdapter.BookListener() {
                @Override public void onBookOpen(BookFile bookFile) {
                }
            }));
        }
    }
});
}
});

```

Тепер, запустивши програму, можемо побачити список файлів електронних книг, які є на пристрої.

					<i>КНТЕУ 121-02-22.МР</i>	<i>Аркуш</i>
						37
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

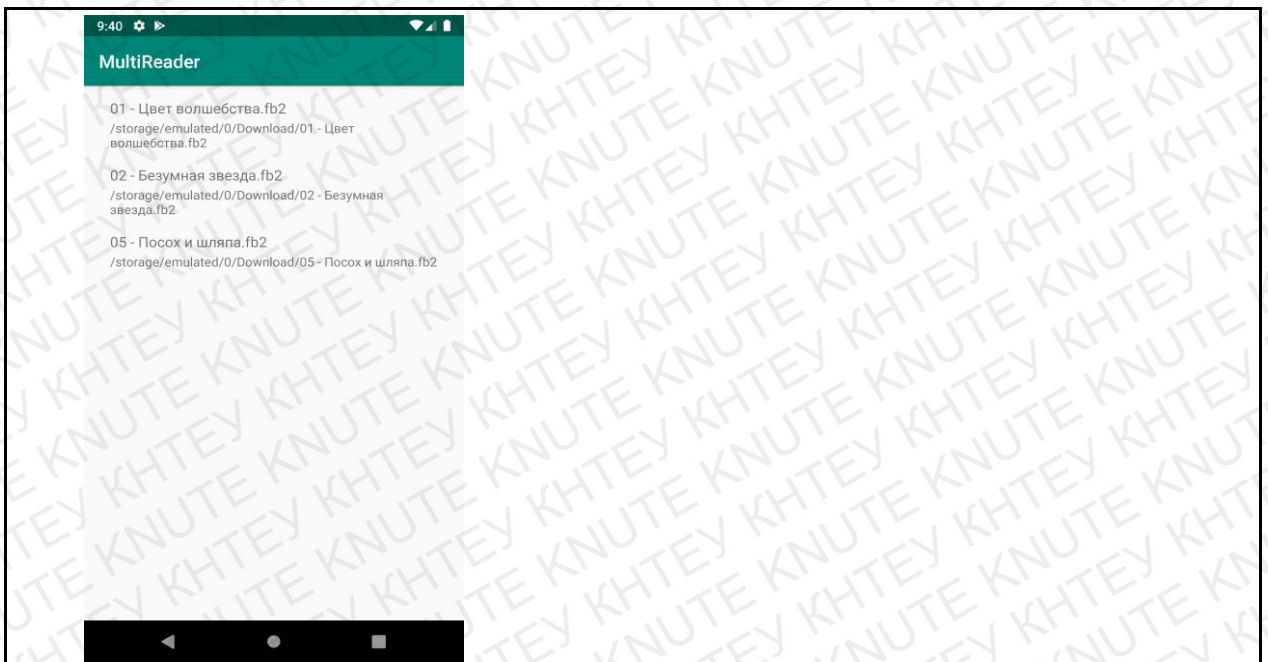


Рис 3.3.Список файлів додатку.

Однак нам потрібно цей файл відкрити та прочитати, у цьому нам і допоможе FBReader SDK. Створимо нову активність, назовемо її ReaderActivity. Тепер із завантаженого вихідного коду нам знадобиться кілька класів. Відкриємо архів з сайту FBReader, з вихідним кодом samples/extensions/src/main та скопіюємо собі в проєкт повністю папку res, за винятком res/layout/main.xml, та з java/org/fbreader/sample/extensions скопіюємо всі класи, крім MainActivity.java. Ці класи містять готові компоненти, які за бажанням можна модифікувати відповідно до потреб.

Тепер відкриємо layout/reader_activity.xml та додамо до нього наступний код розмітки.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/widget_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >

    <ru.androidtools.multireader.TextWidgetExt
        android:id="@+id/text_widget"
        android:layout_width="match_parent"
```

							Аркуш
							38
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121-02-22.МР		

```

        android:layout_height="match_parent"
        android:focusable="true"
        android:scrollbars="vertical"
        android:scrollbarAlwaysDrawVerticalTrack="true"
        android:fadeScrollbars="false"
    />

    <TextView
        android:id="@+id/error_message"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="The app couldn't open the book"
        android:visibility="gone"
    />

</RelativeLayout>

```

TextWidgetExt є обгорткою класу бібліотеки TextWidget, цей клас є основним і саме він відображає вміст книги.

Ініціалізуємо тепер ці об'єкти у кодї активності ReaderActivity.java.

```

public class ReaderActivity extends AppCompatActivity {
    private TextWidgetExt widget;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_reader);

        widget = findViewById(R.id.text_widget);
        View errorView = findViewById(R.id.error_message);
        widget.setVisibility(View.VISIBLE);
        errorView.setVisibility(View.GONE);
    }
}

```

Ми повинні отримати від головної активності шлях до файлу, який користувач бажає відкрити. Для цього ми передаватимемо інтент з однієї активності в іншу, що містить потрібний шлях. Повернемося до MainActivity.java і додамо код створення інтенту в метод onBookOpen() інтерфейсу адаптера.

```

rvFiles.setAdapter(new BooksAdapter(bookFiles, new BooksAdapter.BookListener() {

```

					КНТЕУ 121-02-22.МР	<i>Аркуш</i>
						39
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

```

@Override public void onBookOpen(BookFile bookFile) {

    Intent intent = new Intent(MainActivity.this, ReaderActivity.class);
    intent.putExtra(ReaderActivity.EXTRA_PATH, bookFile.getPath());
    startActivity(intent);
}
}));

```

Після цього надісланий інтеніт нам потрібно отримати в ReaderActivity.java при старті активності. Додамо в onCreate() запис шляху файлу.

```

...

public final static String EXTRA_PATH = "EXTRA_PATH";
private String filepath;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_reader);

    widget = findViewById(R.id.text_widget);
    View errorView = findViewById(R.id.error_message);
    widget.setVisibility(View.VISIBLE);
    errorView.setVisibility(View.GONE);

    filepath = getIntent().getStringExtra(EXTRA_PATH);
}
}

```

Отриманий шлях ми передаємо метод setBook() у об'єкта TextWidgetExt. Після цього він повинен повернути нам об'єкт Book, який можна отримати за допомогою методу TextWidgetExt controller(). Якщо об'єкт не дорівнюватиме null, значить завантаження вмісту пройшло успішно і можна показати книгу користувачу.

```

...
private TextWidgetExt widget;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_reader);

    ...

    try {

```

							Аркуш
							40
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121-02-22.МР		

```

widget.setBook(BookLoader.fromFile(filepath, this, 1L));
Book book = widget.controller().book;
if (book != null) {
    widget.invalidate();
    widget.post(new Runnable() {
        @Override public void run() {
            widget.gotoPage(0);
            setTitle(book.getTitle());
        }
    });
} else {
    errorView.setVisibility(View.VISIBLE);
}
} catch (BookException e) {
    e.printStackTrace();
    errorView.setVisibility(View.VISIBLE);
}
}

```

Тепер, якщо запустити програму, ми вже можемо бачити вміст книги, перегортати його та працювати з ним.

Однак це не весь функціонал, на який здатна ця SDK. Раніше ми додавали файли ресурсів: різні розмітки, іконки та меню. Додамо меню в нашу програму, для цього в ReaderActivity.java перевизначимо метод onCreateOptionsMenu() і додамо до нього наступний код.

```

@Override
public boolean onCreateOptionsMenu(final Menu menu) {
    getMenuInflater().inflate(R.menu.app, menu);
    return true;
}

```

Також перевизначимо методи onPrepareOptionsMenu() та onOptionsItemSelected() для роботи з цим меню.

```

@Override
public boolean onPrepareOptionsMenu(final Menu menu) {
    SearchView searchView = (SearchView) menu.findItem(R.id.menu_search).getActionView();
}

```

					КНТЕУ 121-02-22.МР	Аркуш
						41
Зм.	Аркуш	№ докум	Підпис	Дата		

```

searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextChange(String query) {
        return true;
    }

    @Override
    public boolean onQueryTextSubmit(String query) {
        widget.searchInText(query);
        menu.findItem(R.id.menu_search).collapseActionView();
        return true;
    }
});

menu.findItem(R.id.menu_table_of_contents).setEnabled(TableOfContentsUtil.isAvailable(widget));
;
String name = widget.colorProfile().name;
menu.findItem(R.id.menu_color_profile_light).setChecked("defaultLight".equals(name));
menu.findItem(R.id.menu_color_profile_dark).setChecked("defaultDark".equals(name));
menu.findItem(R.id.menu_color_profile_dark_with_bg).setChecked("darkWithBg".equals(name));
menu.findItem(R.id.menu_color_profile_pink).setChecked("pink".equals(name));
return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    BaseStyle baseStyle = widget.baseStyle();

    switch (item.getItemId()) {
        case R.id.menu_table_of_contents: {
            final Intent intent = TableOfContentsUtil.intent(widget);
            if (intent != null) {
                startActivityForResult(intent, REQUEST_TABLE_OF_CONTENT);
            }
            break;
        }
        case R.id.menu_zoom_in:
            baseStyle.fontSize.setValue(baseStyle.fontSize.getValue() + 2);
            break;
        case R.id.menu_zoom_out:
            baseStyle.fontSize.setValue(baseStyle.fontSize.getValue() - 2);
            break;
        case R.id.menu_color_profile_light:
            widget.setColorProfileName("defaultLight");
            break;
        case R.id.menu_color_profile_dark:
            widget.setColorProfileName("defaultDark");
            break;
        case R.id.menu_color_profile_dark_with_bg:
            widget.setColorProfileName("darkWithBg");
            break;
        case R.id.menu_color_profile_pink:
            widget.setColorProfileName("pink");
            break;
    }
    widget.clearTextCaches();
    widget.invalidate();
    return true;
}

```

Таким чином ми додамо додатку меню, що дозволяє нам:

					КНТЕУ 121-02-22.МР	<i>Аркуш</i>
						42
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

- Міняти розмір тексту;
- Змінювати стиль тексту;
- Шукати за текстом;
- Відкривати зміст.

Щодо змісту варто сказати окремо. У кодї, взятому з бібліотеки, зміст подано у вигляді окремої активності, ми її скопіювали раніше, але не додали до маніфесту. Тому відкриємо файл маніфесту `AndroidManifest.xml` та зареєструємо у ньому нову активність.

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    ...
    <activity android:name=".TableOfContentsActivity"/>
</application>
```

Тепер при відкритті змісту через меню у нас буде запускатися нова активність, що містить зміст відкритої книги. Варто відзначити, що не всі книги містять такий зміст, тому в таких випадках кнопка меню буде неактивною.

При виборі розділу назад у `ReaderActivity` буде повертатися номер сторінки, яку потрібно перейти. Щоб отримати цей результат, в `ReaderActivity.java` ми перевизначимо метод `onActivityResult()`, який додамо наступний код.

```
public class ReaderActivity extends AppCompatActivity {
    private final int REQUEST_TABLE_OF_CONTENT = 1;
    ...
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        switch (requestCode) {
            case REQUEST_TABLE_OF_CONTENT:
                if (resultCode == RESULT_OK) {
                    int ref = data.getIntExtra(String.valueOf(TableOfContentsUtil.Key.reference), -1);
                    if (widget != null && ref != -1) {
```

						<i>Аркуш</i>
					<i>KHTEY 121-02-22.MP</i>	43
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

```

        widget.jumpTo(new FixedPosition(ref, 0, 0));
    }
}
default:
    super.onActivityResult(requestCode, resultCode, data);
}
}

```

Як можна побачити, бібліотека надає розробникам весь свій функціонал, єдиним обмеженням безкоштовної версії тут є те, що вона відкриває лише перші кілька сторінок книги, про що йдеться під час відкриття книги.

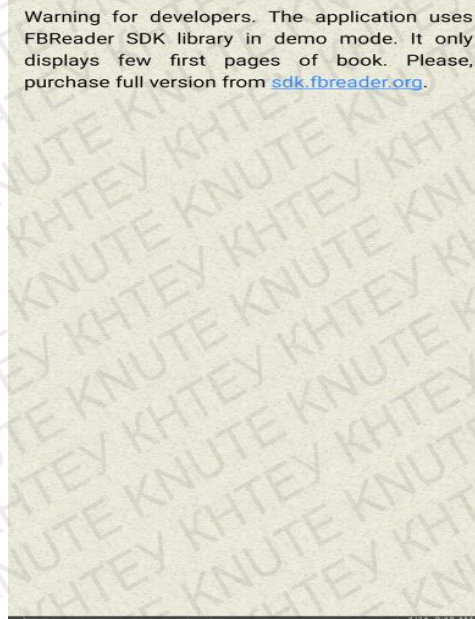


Рис 3.4. Повідомлення від FBReader

Якщо в наявності є ключ SDK можна додати його до своєї програми, знадобиться у файлі `build.gradle` модуля програми додати в блоці `buildTypes` наступний код.

```

buildTypes {
    ...

    all {
        resValue 'string', 'fbreader_sdk_key', 'put your fbreader sdk key here'
    }
}

```

Незважаючи на те, що двигун FBReader досить потужний, а також враховуючи графічні ресурси, що додаються разом з бібліотекою,

					КНТЕУ 121-02-22.МР	Аркуш
						44
Зм.	Аркуш	№ докум	Підпис	Дата		

підсумковий APK вийшов невеликого розміру, що говорить про хорошу оптимізацію бібліотеки.

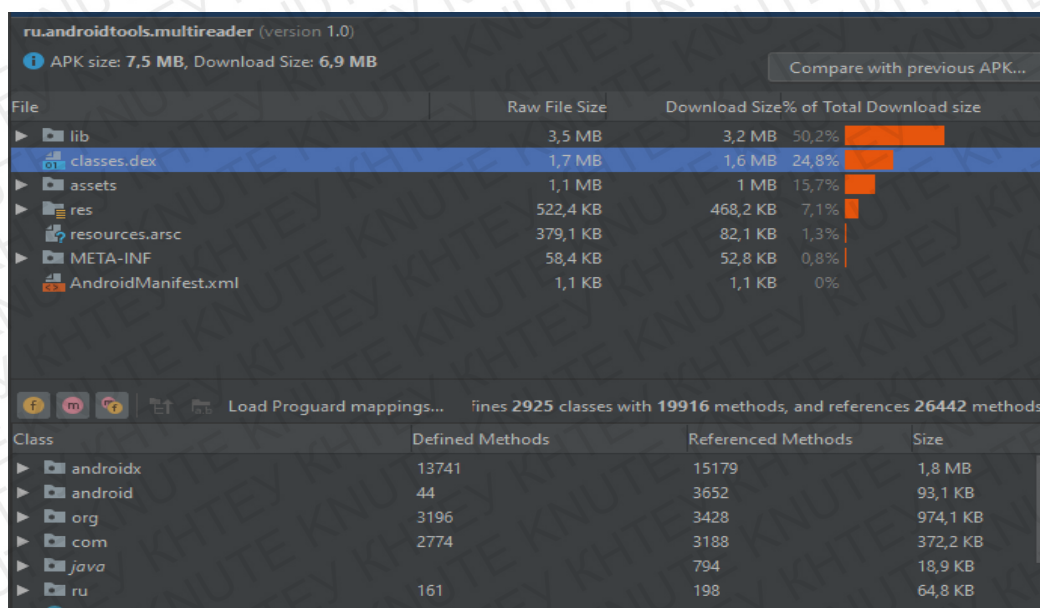


Рис. 3.5. Оптимізація додатку

3.3. Висновки до розділу 3

Отже, за допомогою FBReader SDK та Android Studio було створено додаток для читання електронних книг. Можна вже залишити як є або продовжити модифікувати, змінюючи інтерфейс і дизайн так, як потрібно. Додаток оптимізований та простий у використанні, а завдяки можливості створення тем для додатку та умовної безкоштовності в користуванні, додаток отримає більше поширення.

						Аркуш
						45
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121-02-22.МР	

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У ході написання випускної кваліфікаційної роботи, було досліджено основні теоретичні відомості про навчання впродовж життя та мобільні додатки які можна використати для такого навчання. Також було виявлено що даний додаток є гарним інструментом при навчанні завдяки мобільності, доступності та універсальності. Мобільні додатки одні з найпоширеніших засобів, що застосовуються в системах навчання впродовж життя. Розглянуті засоби створення програмного забезпечення є оптимальними для створення даного додатку є вигідними для універсальності та персоналізації додатку.

					<i>КНТЕУ 121-02-22.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>		<i>01.11.2021</i>	<i>Модель навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>		<i>Криворучко О.В.</i>		<i>01.11.2021</i>		<i>ВП</i>	<i>46</i>	<i>49</i>
<i>Гарант</i>		<i>Токар В.В.</i>		<i>01.11.2021</i>		<i>Факультет інформаційних технологій, 2м курс, 2 група</i>		
<i>Розроб.</i>		<i>Шапочка Д.В.</i>		<i>01.11.2021</i>				
					<i>Висновки та пропозиції</i>			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Department of Education and Science (2000). Learning for Life: Paper on Adult Education. Dublin: Stationery Office.
2. Commission of the European Communities: "Adult learning: It is never too late to learn". COM(2006) 614 final. Brussels, 23.10.2006.
3. Fischer, Gerhard (2000). "Lifelong Learning - More than Training" in Journal of Interactive Learning Research, Volume 11 issue 3/4 pp 265–294.
4. O'Grady, Anne (2013). Lifelong Learning in the UK: An introductory guide for Education Studies. Oxon: Routledge.
5. Jarvis, Peter (2009). The Routledge International Handbook of Lifelong Learning. Oxon: Routledge.
6. Qinhua, Zheng; Dongming, Ma; Zhiying, Nian; Hao, Xie (2016). Adult Competencies for Lifelong Learning. Aalborg: River Publishers.
7. Sharma, Tara Chand (2004). Meaning Of Lifelong Learning. New Delhi: Sarup & Sons.
8. Keith, Davies, W.; Norman, Longworth (2013). *Lifelong Learning*. Oxon: Routledge.
9. Aspin, David N. & Chapman, Judith D. (2007) "Lifelong Learning Concepts and Conceptions" in: David N. Aspin, ed.: Philosophical Perspectives on Lifelong Learning, Springer.
10. Barbara Liskov with John Guttag (2000). Program Development in Java - Abstraction, Specification, and Object-Oriented Design. USA, Addison Wesley.

					<i>КНТЕУ 121-02-22.МР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. кафедри		Криворучко О.В.		27.02.2021	Модель навчального мобільного додатку інтеграції керування знаннями і безперервним навчанням	Стадія	Аркуш	Аркушів
Керівник		Криворучко О.В.		27.02.2021		СВД	47	49
Гарант		Токар В.В.		27.02.2021		Факультет інформаційних технологій, 2м курс, 2 група		
Розроб.		Шапочка Д.В.		27.02.2021				
					Список використаних джерел			

11. Object-oriented programming "The History of Java Technology". Sun Developer Network. c. 1995.
12. Gosling, James; Joy, Bill; Steele, Guy; Bracha, Gilad; Buckley, Alex (2014). The Java® Language Specification (PDF) (Java SE 8 ed.).
13. Lindholm, Tim; Yellin, Frank (1999). The Java Virtual Machine Specification (2nd ed.). Addison-Wesley.
14. Gosling, James; Joy, Bill; Steele, Guy L., Jr.; Bracha, Gilad (2005). The Java Language Specification (3rd ed.). Addison-Wesley.
15. Binstock, Andrew (May 20, 2015). "Java's 20 Years of Innovation". Forbes [Електронний ресурс]:
(<https://web.archive.org/web/20160314102242/http://www.forbes.com/sites/oracle/2015/05/20/javas-20-years-of-innovation/>).
16. "Write once, run anywhere?". Computer Weekly. May 2, 2002 [Електронний ресурс]:
(<https://web.archive.org/web/20210813193857/https://www.computerweekly.com/feature/Write-once-run-anywhere>).
17. "Android Language Breakdown". Open Hub. October 25, 2017 [Електронний ресурс]:
(https://web.archive.org/web/20171214124807/https://www.openhub.net/p/android/analyses/latest/languages_summary).
18. "Google's Android OS: Past, Present, and Future". PhoneArena. August 18, 2011. [Електронний ресурс]:
(https://web.archive.org/web/20170313044444/http://www.phonearena.com/news/Googles-Android-OS-Past-Present-and-Future_id21273).
19. Chu, Eric (April 13, 2011). "Android Developers Blog: New Carrier Billing Options on Android Market". android-developers.blogspot.com. [Електронний ресурс]:
(<http://archive.wikiwix.com/cache/20110628054436/http://android->

					<i>KHTEY 121-02-22.MP</i>	<i>Аркуш</i>
						48
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

developers.blogspot.com/2011/04/new-carrier-billing-options-on-android.html)

20. Випущені вихідні коди Android [Електронний ресурс]. - Режим доступу: URL : <http://habrahabr.ru/post/132603/>
21. Основи безпеки операційної системи Android [Електронний ресурс]. -Режим доступу : URL : <http://habrahabr.ru/post/176131/>
22. Annual number of global mobile app downloads 2017-2022 [Електронний ресурс] // Statista. – 2019. – Режим доступу: <https://bit.ly/2zTtudv>.
23. Лубко Д. В. Методологія проектування та інструментарій для створення мобільних додатків / Д. В. Лубко // Вісник Нац. техн. ун-ту "ХПІ" : зб. наук. пр. Темат. вип. : Нові рішення в сучасних технологіях. – Харків : НТУ "ХПІ". – 2013. – № 56 (1029). – С. 117-122.

					КНТЕУ 121-02-22.МР	<i>Аркуш</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		49

ДОДАТКИ

Додаток А

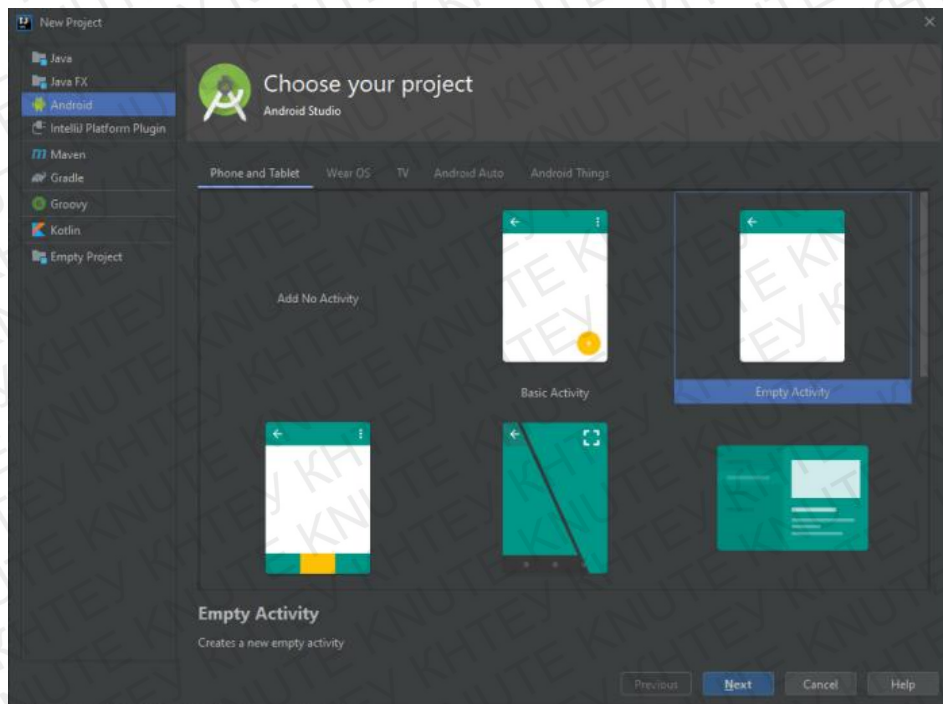


Рис. 3.1. Меню створення проекту в Android studio

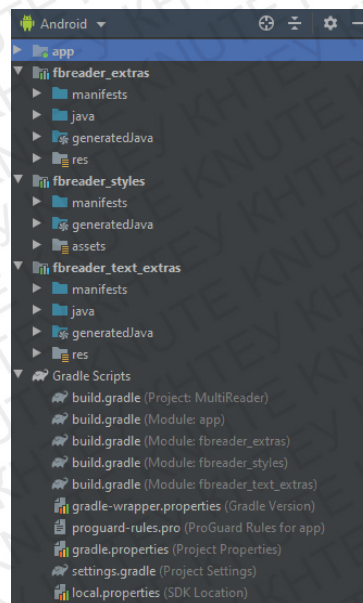


Рис. 3.2. Список модулів додатку

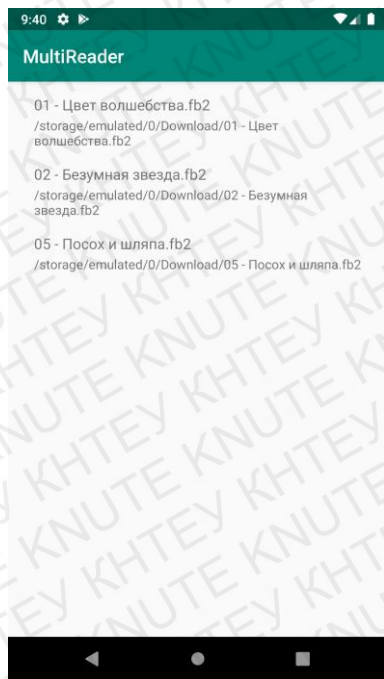


Рис. 3.3. Список файлів додатку

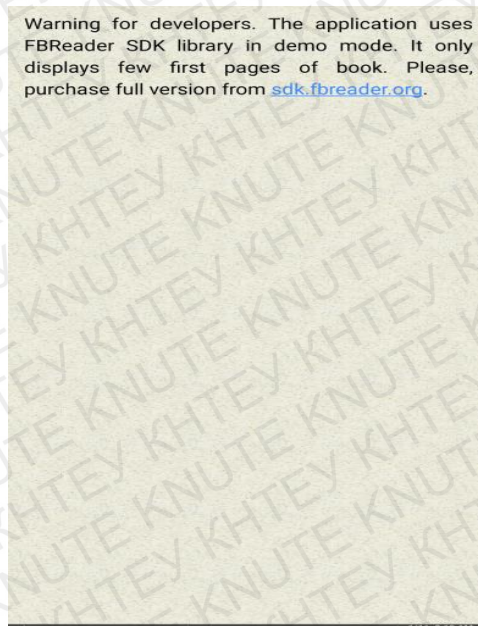


Рис. 3.4. Повідомлення від FBReader

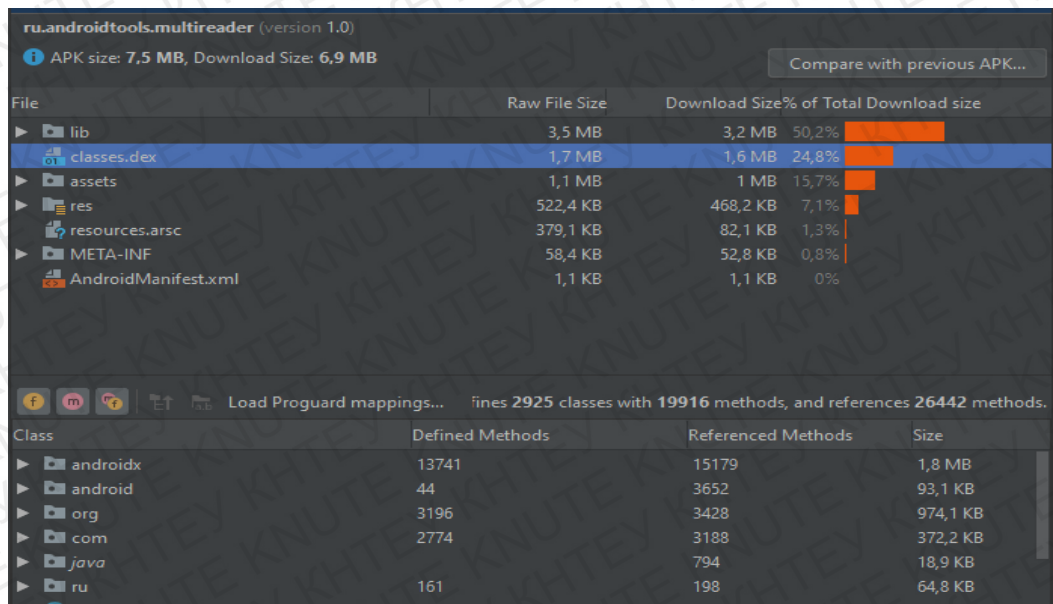


Рис. 3.5. Оптимізація додатку

```
allprojects {  
    repositories {  
        ...  
        maven {  
            url "https://sdk.fbreader.org/maven"  
        }  
    }  
}  
  
dependencies {  
    ...  
  
    implementation 'com.googlecode.json-simple:json-simple:1.1'  
    implementation 'org.fbreader:book:1.0.0-rc01'  
    implementation 'org.fbreader:config_client:1.0.0-rc01'  
    implementation 'org.fbreader:config_provider:1.0.0-rc01'  
    implementation 'org.fbreader:filesystem:1.0.0-rc01'  
    implementation 'org.fbreader:fontentry:1.0.0-rc01'  
    implementation 'org.fbreader:format_interface:1.0.0-rc01'  
    implementation 'org.fbreader:image:1.0.0-rc01'  
    implementation 'org.fbreader:intent:1.0.0-rc01'  
    implementation 'org.fbreader:language:1.0.0-rc01'  
    implementation 'org.fbreader:loader:1.0.0-rc01'  
    implementation 'org.fbreader:options:1.0.0-rc01'  
    implementation 'org.fbreader:text_client:1.0.0-rc01'  
    implementation 'org.fbreader:text_provider:1.0.0-rc01'  
    implementation 'org.fbreader:text_util:1.0.0-rc01'  
    implementation 'org.fbreader:text_view:1.0.0-rc01'  
    implementation 'org.fbreader:toc:1.0.0-rc01'  
    implementation 'org.fbreader:util:1.0.0-rc01'  
    implementation 'org.fbreader:view:1.0.0-rc01'  
    implementation 'org.fbreader:vimgadgets:1.0.0-rc01'  
    implementation 'org.fbreader:zip-amse:1.0.0-rc01'  
}  
  
android {  
    ...  
  
    compileOptions {  
        sourceCompatibility JavaVersion.VERSION_1_8  
        targetCompatibility JavaVersion.VERSION_1_8  
    }  
}  
  
include 'fbreader_extras'  
  
project(':fbreader_extras').projectDir = new File('fbreader/extras')
```

```

include 'fbreader_styles'

project(':fbreader_styles').projectDir = new File('fbreader/styles')

include 'fbreader_text_extras'

project(':fbreader_text_extras').projectDir = new File('fbreader/text_extras')

dependencies {
    ...

    implementation project(':fbreader_extras')
    implementation project(':fbreader_styles')
    implementation project(':fbreader_text_extras')
}

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:paddingRight="12dp"
    android:paddingLeft="12dp"
    >
    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="6dp"
        android:id="@+id/rv_files"
    />

```

```
</RelativeLayout>
```

```
public class MainActivity extends AppCompatActivity {  
  
    private RecyclerView rvFiles;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        rvFiles = findViewById(R.id.rv_files);  
        rvFiles.setLayoutManager(new LinearLayoutManager(this));  
    }  
}
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
  
    package="ru.androidtools.multireader">  
  
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>  
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
  
    <application  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="@string/app_name"  
        android:roundIcon="@mipmap/ic_launcher_round"  
        android:supportsRtl="true"  
        android:theme="@style/AppTheme">  
        <activity android:name=".MainActivity">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN"/>  
  
                <category android:name="android.intent.category.LAUNCHER"/>  
            </intent-filter>  
        </activity>  
    </application>  
  
</manifest>
```

```
public class MainActivity extends AppCompatActivity {  
  
    ...  
  
    private static final int REQUEST_PERMISSION = 101;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        rvFiles = findViewById(R.id.rv_files);  
        rvFiles.setLayoutManager(new LinearLayoutManager(this));  
  
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
            checkPermission();  
        } else {  
            generateList();  
        }  
    }  
}
```

```

    }
}

@TargetApi(Build.VERSION_CODES.M) private void checkPermission() {
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE_EXTERNAL_STORAGE)
        == PackageManager.PERMISSION_GRANTED) {
        generateList();
    } else {
        ActivityCompat.requestPermissions(this,
            new String[] { Manifest.permission.WRITE_EXTERNAL_STORAGE }, REQUEST_PERMISSION);
    }
}

@Override public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions,

    @NonNull int[] grantResults) {

    switch (requestCode) {

        case REQUEST_PERMISSION: {

            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

                generateList();

            }

            break;

        }

    }

}

private void generateList() {
    List<StorageBean> storageBeans = StorageUtils.getStorageData(this); // получение списка
монтированных карт памяти на устройстве
    List<String> paths = new ArrayList<>();
    if (storageBeans != null) {

        for (StorageBean storageBean : storageBeans) {
            paths.add(storageBean.getPath());
        }
        else {
            String path = Environment.getExternalStorageDirectory().getAbsolutePath();
            paths.add(path);
        }
        ListFilesTask listFilesTask = new ListFilesTask(paths);
        listFilesTask.setListener(new ListFilesTask.ListFilesListener() {
            @Override public void onTaskCompleted(List<File> files) {
            }
        });
        listFilesTask.execute();
    }

}

static class ListFilesTask extends AsyncTask<Void, Void, List<File>> {

```

```

public interface ListFilesListener {
    void onTaskCompleted(List<File> files);
}

private ListFilesListener listener;
private List<String> startPaths;
private List<File> files;
private boolean completed;

public ListFilesTask(List<String> startPaths) {
    this.startPaths = new ArrayList<>(startPaths);
    this.files = new ArrayList<>();
    this.completed = false;
}

public void setListener(ListFilesListener listener) {
    this.listener = listener;
    if (completed && listener != null && files != null) {
        listener.onTaskCompleted(files);
    }
}

@Override protected List<File> doInBackground(Void... voids) {
    List<File> fileList = new ArrayList<>();
    for (String s : startPaths) {
        searchFiles(fileList, new File(s));
    }
    return fileList;
}

@Override protected void onPostExecute(List<File> files) {
    completed = true;
    if (listener != null) {
        listener.onTaskCompleted(files);
    } else {
        this.files = new ArrayList<>(files);
    }
}

private void searchFiles(List<File> list, File dir) {
    String epubPattern = ".epub";
    String fb2Pattern = ".fb2";

    File[] listFiles = dir.listFiles();
    if (listFiles != null) {
        for (File listFile : listFiles) {
            if (listFile.isDirectory()) {
                searchFiles(list, listFile);
            } else {
                if (listFile.getName().endsWith(epubPattern) || listFile.getName().endsWith(fb2Pattern)) {
                    list.add(listFile);
                }
            }
        }
    }
}

public BookFile(String filename, String path) {
    this.filename = filename;
    this.path = path;
}

public String getFilename() {
    return filename;
}

```

```

public String getPath() {
    return path;
}

@Override public int hashCode() {
    return filename.hashCode() + path.hashCode();
}

@Override public boolean equals(Object obj) {
    if (obj instanceof BookFile) {
        BookFile bookFile = (BookFile) obj;
        return this.filename.equals(bookFile.getFilename())
            &&
            this.path.equals(bookFile.getPath());
    }
    return false;
}
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/item_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:layout_marginTop="2dp"
    android:orientation="vertical"
    android:paddingBottom="6dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="6dp"
    >

    <TextView
        android:id="@+id/tv_filename"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="filename"
        android:textSize="16sp"
        />

    <TextView
        android:id="@+id/tv_path"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="2dp"
        android:text="path"
        android:textSize="14sp"
        />

</LinearLayout>
public class BooksAdapter extends RecyclerView.Adapter<BooksAdapter.BookHolder> {
    public interface BookListener {
        void onBookOpen(BookFile bookFile);
    }

    private BookListener listener;
    private List<BookFile> books;

    public BooksAdapter(List<BookFile> books, BookListener listener) {
        this.books = new ArrayList<>(books);
        this.listener = listener;
    }

    @NonNull @Override public BookHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View view =
            LayoutInflater.from(parent.getContext()).inflate(R.layout.file_list_item,
parent,
false);
        return new BookHolder(view);
    }
}

```



```

    }

    @Override public void onBindViewHolder(@NonNull BookHolder holder, int position) {
        BookFile bookFile = books.get(position);
        holder.bind(bookFile);
    }

    @Override public int getItemCount() {
        return books.size();
    }

    class BookHolder extends RecyclerView.ViewHolder {
        private LinearLayout itemLayout;
        private TextView tvPath, tvName;

        public BookHolder(@NonNull View itemView) {
            super(itemView);
            itemLayout = itemView.findViewById(R.id.item_layout);
            tvName = itemView.findViewById(R.id.tv_filename);
            tvPath = itemView.findViewById(R.id.tv_path);
        }

        public void bind(BookFile bookFile) {
            tvPath.setText(bookFile.getPath());
            tvName.setText(bookFile.getFilename());
            itemLayout.setOnClickListener(new View.OnClickListener() {
                @Override public void onClick(View view) {
                    listener.onBookOpen(bookFile);
                }
            });
        }
    }
}

listFilesTask.setListener(new ListFilesTask.ListFilesListener() {

    @Override public void onTaskCompleted(List<File> files) {

        if (!isFinishing()) {
            List<BookFile> bookFiles = new ArrayList<>();
            for (File f : files) {
                BookFile bookFile = new BookFile(f.getName(), f.getAbsolutePath());
                if (!bookFiles.contains(bookFile)) bookFiles.add(bookFile);
            }
            Collections.sort(bookFiles, new Comparator<BookFile>() {
                @Override public int compare(BookFile bookFile, BookFile t1) {
                    return bookFile.getFilename().compareToIgnoreCase(t1.getFilename());
                }
            });
            rvFiles.setAdapter(new BooksAdapter(bookFiles, new BooksAdapter.BookListener() {
                @Override public void onBookOpen(BookFile bookFile) {

                }
            }));
        }
    });
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/widget_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >

    <ru.androidtools.multireader.TextWidgetExt

```

```

        android:id="@+id/text_widget"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:focusable="true"
        android:scrollbars="vertical"
        android:scrollbarAlwaysDrawVerticalTrack="true"
        android:fadeScrollbars="false"
    />

    <TextView
        android:id="@+id/error_message"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="The app couldn't open the book"
        android:visibility="gone"
    />

</RelativeLayout>
public class ReaderActivity extends AppCompatActivity {
    private TextWidgetExt widget;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_reader);

        widget = findViewById(R.id.text_widget);
        View errorView = findViewById(R.id.error_message);
        widget.setVisibility(View.VISIBLE);
        errorView.setVisibility(View.GONE);
    }
}

rvFiles.setAdapter(new BooksAdapter(bookFiles, new BooksAdapter.BookListener() {

    @Override public void onBookOpen(BookFile bookFile) {

        Intent intent = new Intent(MainActivity.this, ReaderActivity.class);
        intent.putExtra(ReaderActivity.EXTRA_PATH, bookFile.getPath());
        startActivity(intent);
    }
}));

...

public final static String EXTRA_PATH = "EXTRA_PATH";
private String filepath;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_reader);

    widget = findViewById(R.id.text_widget);
    View errorView = findViewById(R.id.error_message);
    widget.setVisibility(View.VISIBLE);
    errorView.setVisibility(View.GONE);

    filepath = getIntent().getStringExtra(EXTRA_PATH);
}
...
private TextWidgetExt widget;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_reader);
    ...

    try {
        widget.setBook(BookLoader.fromFile(filepath, this, 1L));
        Book book = widget.controller().book;
        if (book != null) {
            widget.invalidate();
            widget.post(new Runnable() {
                @Override public void run() {
                    widget.gotoPage(0);
                    setTitle(book.getTitle());
                }
            });
        } else {
            errorView.setVisibility(View.VISIBLE);
        }
    } catch (BookException e) {
        e.printStackTrace();
        errorView.setVisibility(View.VISIBLE);
    }
}

@Override

public boolean onCreateOptionsMenu(final Menu menu) {

    getMenuInflater().inflate(R.menu.app, menu);

    return true;
}

@Override
public boolean onPrepareOptionsMenu(final Menu menu) {
    SearchView searchView = (SearchView) menu.findItem(R.id.menu_search).getActionView();
    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextChange(String query) {
            return true;
        }

        @Override
        public boolean onQueryTextSubmit(String query) {
            widget.searchInText(query);
            menu.findItem(R.id.menu_search).collapseActionView();
            return true;
        }
    });

    menu.findItem(R.id.menu_table_of_contents).setEnabled(TableOfContentsUtil.isAvailable(widget))
    ;
    String name = widget.colorProfile().name;
    menu.findItem(R.id.menu_color_profile_light).setChecked("defaultLight".equals(name));
    menu.findItem(R.id.menu_color_profile_dark).setChecked("defaultDark".equals(name));
    menu.findItem(R.id.menu_color_profile_dark_with_bg).setChecked("darkWithBg".equals(name));
    menu.findItem(R.id.menu_color_profile_pink).setChecked("pink".equals(name));
    return true;
}

```

```

    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        BaseStyle baseStyle = widget.baseStyle();

        switch (item.getItemId()) {
            case R.id.menu_table_of_contents: {
                final Intent intent = TableOfContentsUtil.intent(widget);
                if (intent != null) {
                    startActivityForResult(intent, REQUEST_TABLE_OF_CONTENT);
                }
                break;
            }
            case R.id.menu_zoom_in:
                baseStyle.fontSize.setValue(baseStyle.fontSize.getValue() + 2);
                break;
            case R.id.menu_zoom_out:
                baseStyle.fontSize.setValue(baseStyle.fontSize.getValue() - 2);
                break;
            case R.id.menu_color_profile_light:
                widget.setColorProfileName("defaultLight");
                break;
            case R.id.menu_color_profile_dark:
                widget.setColorProfileName("defaultDark");
                break;
            case R.id.menu_color_profile_dark_with_bg:
                widget.setColorProfileName("darkWithBg");
                break;
            case R.id.menu_color_profile_pink:
                widget.setColorProfileName("pink");
                break;
        }
        widget.clearTextCaches();
        widget.invalidate();
        return true;
    }
}
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    ...
    <activity android:name=".TableOfContentsActivity"/>
</application>
public class ReaderActivity extends AppCompatActivity {
    private final int REQUEST_TABLE_OF_CONTENT = 1;
    ...

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        switch (requestCode) {
            case REQUEST_TABLE_OF_CONTENT:
                if (resultCode == RESULT_OK) {
                    int ref = data.getIntExtra(String.valueOf(TableOfContentsUtil.Key.reference), -1);
                    if (widget != null && ref != -1) {
                        widget.jumpTo(new FixedPosition(ref, 0, 0));
                    }
                }
            default:
                super.onActivityResult(requestCode, resultCode, data);
        }
    }
}
buildTypes {
    ...
}

```

```
all {  
  resValue 'string', 'fbreader_sdk_key', 'put your fbreader sdk key here'  
}
```