# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

#### МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторної роботи №1 з навчальной дисципліни «Мобільні технології»

для студентів РВО «бакалавр» 3 року навчання денної та заочної форми спеціальності 122 «Комп'ютерні науки»

#### Затверджено

на засіданні групи забезпечення спеціальності Протокол № <u>15</u> від « <u>30</u> <u>06</u> 2021р. Голова групи <u>Мили</u> Мещеряков В.І.

Затверджено

на засіданні кафедри \_\_\_\_\_ Протокол № <u>10</u> від 30 » <u>06</u> 2021 р. Завідувач кафедри // Казакова Н.Ф.

ОДЕСА -2021

# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

# МЕТОДИЧНІ ВКАЗІВКИ до лабораторної роботи №1 з навчальної дисципліни «Мобільні технології»

для студентів РВО «бакалавр» 3 року навчання

денної та заочної форми спеціальності 122

«Комп'ютерні науки»

Одеса – 2021

Методичні вказівки до лабораторної роботи з дисципліни "*Мобільні технології* для студентів 3 року навчання денної та заочної форми за спеціальністю 122 «Комп'ютерні науки», рівень вищої освіти бакалавр / Козловська В. П. – Одеса, ОДЕКУ, 2021.

# Лабораторна робота № 1

# Tema: «Установка інструментарію розробника Android. Створення першого додатка на Android »

### Мета лабораторної роботи:

- ознайомитися з інструментами розробки Android-додатків;
- на прикладі найпростіших програм розібрати структуру типового Android-додатку;
- познайомитися з різними середовищами розробки Android-додатків.

# Хід роботи:

### ВСТАНОВЛЕННЯ ІНСТРУМЕНТАРІЮ РОЗРОБНИКА ANDROID

### JDK (Java Development Kit)

Для встановлення JDK необхідно перейти за посиланням <u>http://goo.gl/t3G2k1</u> та завантажити інсталяційний файл відповідно до своєї OC.

Найбільш популярними мовами програмування під Android в даний момент є Java і Kotlin. JDK необхідна для програмування на обох мовах.

#### Що потрібно для розробки додатків?

Варто відзначити, що розробляти програми під Android можна за допомогою різних фреймворків і мов програмування. Так в якості мов програмування можуть застосовуватися Java, Kotlin, Dart (фреймворк Flutter), C++, Python, C# (платформа Xamarin) тощо.

В даному курсі пропонується використовувати мову Java, як найбільш поширену і використовувану, а також за бажанням можна використовувати мову Kotlin.

Кotlin представляє сучасну, статично типізовану мову програмування, яка створена і дуже швидко розвивається компанією JetBrains. Коло платформ, для яких можна створювати додатки на Kotlin, надзвичайно широке – Windows, Linux, Mac OS, iOS, Android. Найпопулярнішим напрямком, де застосовується Kotlin, є, перш за все, розробка під OC Android. Причому настільки популярним, що компанія Google на конференції Google I / О 2017 проголосила Kotlin однією з офіційних мов для розробки під Android (поряд з Java i C ++), а інструменти по роботі з даною мовою були за замовчуванням включені в функціонал середовища розробки Android Strudio починаючи з версії 3.0.

### Середовище розробки

На даний момент для програмування на мові Java рекомендованої IDE  $\epsilon$  Android Studio, в основі якої лежить InelliJ IDEA. Завантажити IDE можна за посиланням <u>http://developer.android.com/sdk/index.html</u> (рис. 1); в процесі установки Android Studio потрібно обов'язково вибрати установку Android SDK. При установці починаючи з версії 3.0 в цьому середовищі можна розробляти Android-додатки як на мові Java, так і на мові Kotlin.



Рисунок 1 – Можливість скачування середовища Android Studio

Якщо при установці виникла помилка, пов'язана з відсутністю JDK (JDK not found), необхідно в змінних середовища вказати шлях до встановленого раніше JDK (докладніше – дивіться посилання <u>http://stackoverflow.com/questions/16574189/androidstudio-installation- on-</u>windows-7-fails-no-jdk-found).

### Установка Android SDK

Крім самого середовища Android Studio для розробки також буде потрібно набір інструментів, який називається Android SDK (Software Development Kit). Наприклад, якщо раніше Android SDK ще не було встановлено, то при першому зверненні до Android Studio вона повідомить, що Android SDK відсутня (рис. 2).



Рисунок 2 – Спроба роботи в Android Studio без установки Android SDK

Можна окремо вручну завантажити Android SDK з офіційного сайту і встановити його. Або можна зробити це безпосередньо з Android Studio. Так, при натисканні на кнопку Next рис. 2) на наступному екрані установки Android Studio буде запропоновано завантажити Android SDK для останньої версії API (рис. 3).

Тут же можна вказати місце для установки Android SDK, якщо не влаштовує шлях за замовчуванням.

Після натискання кнопки Next відображається вікно зі зведенням того, що саме буде встановлено (рис. 4).

🖌 Android Studio Setup Wizard	– 🗆 X
SDK Components Setu	ıp
Check the components you want to update/install. Cl	lick Next to continue. The collection of Android platform APIs, tools
API 30: Android 11.0 (R) – (142 MB)	and utilities that enables you to debug, profile, and compile your apps. The setup wizard will update your current Android SDK installation (if necessary) or install a new version.
Android SDK Location:	Total download size: 452 MB
C:\Users\Eugene\AppData\Local\Android\sdk	Disk space available on drive : 56 GB
A Target folder is neither empty nor does it point to	an existing SDK installation.           Previ         Next         Cancel         Finish

# Рисунок 3 – Установка Android SDK из Android Studio



Рисунок 4 – Повідомлення про обрані установки для Android Studio

Пакет Android SDK містить бібліотеки та інструменти, необхідні для розробки Android-додатків.

Android SDK включає в себе:

- SDK Platform окрема платформа для кожної версії Android;
- SDK Tools інструменти налагодження і тестування, а також інші корисні утиліти. Також включає в себе набір переносних залежних інструментів;
- Приклади додатків містить реальні приклади коду, щоб краще зрозуміти, як користуватися API;
- Документація надає автономний доступ до новітньої документації АРІ;
- Допоміжний інструментарій Android додаткові API, відсутні в стандартній платформі;
- Google Play Billing інтеграція сервісу білінгу в додатки.

За замовчуванням SDK Android включає в себе не всі інструменти, які необхідні для того, щоб приступити до розробки додатків. Інструменти, платформи та інші компоненти представлені в Android SDK у вигляді окремих пакетів, які при необхідності можна завантажити за допомогою менеджера SDK Android. Тому, перш ніж приступити до роботи, в пакет SDK Android необхідно додати деякі додаткові пакети.

Детальніше про встановлення додаткових пакетів для SDK Android -<u>http://developer.android.com/intl/ru/sdk/installing/adding-packages.html</u> (читати обов'язково)

### Налагодження на фізичному пристрої

Для тестування і налагодження Android-додатку на фізичному пристрої необхідно включити на пристрої режим розробника, а також завантажити Google USB Driver (см. <u>http://developer.android.com/sdk/win-usb.html</u>) і установити його (см. <u>http://developer.android.com/tools/extras/oem-usb.html</u>).

Деякі фірми виробники пристроїв (наприклад, Samsung) надають власні драйвера і установник для них для свого пристрою необхідно знайти в мережі Інтернет.

#### Налагодження на стандартному емуляторі

До складу Android SDK входить стандартний емулятор пристрою під керуванням OC Android. Даний емулятор не відрізняється швидкодією, але є безкоштовним і має широкі можливості для емуляції.

### Налагодження на емуляторі Genymotion (бажано)

Альтернативою стандартному емулятору Android є емулятор **Genymotion** – <u>https://www.genymotion.com/#!/</u>. Для установки і скачування пристроїв необхідно зареєструватися.

Після установки Genymotion необхідно встановити плагін для обраної IDE. Інструкція по установці плагіна –

https://www.genymotion.com/#!/download . Запущений емулятор Genymotion визначається як підключений пристрій.

### IntelliJ IDEA – середовище розробки додатків на мові Kotlin

Для розробки додатків на мові Kotlin можна використовувати таке середовище розробки як IntelliJ IDEA від компанії JetBrains. Завантажити її можна за адресою <u>https://www.jetbrains.com/idea/download/</u>. Дане середовище доступне як для Windows, так і для MacOS і Linux. Є безкоштовний випуск – **Community**, і платний – **Ultimate**.

В даному випадку завантажимо і встановимо безкоштовний випуск IntelliJ IDEA Community.

## Установка IntelliJ IDEA



Запустим программу установки:

Рисунок 5 – Початок установки IntelliJ IDEA

Після натискання на першому вікні на кнопку Next (рис. 5) на другому вікні відображається шлях, по якому буде встановлюватися середовище (рис. 6).



Рисунок 6 – Шлях встановлення середовища

Можна оставити шлях за замовчуванням, але можна його змінити. На наступному вікні відображаються деякі конфігураційні налаштування. Можна, наприклад, зв'язати середу з типами файлів або настроїти створення іконок середовища на робочому столі (рис. 7).

IntelliJ IDEA Community Edition Setup		_		$\times$
Installation Option Configure your Intelli	<b>s</b> J IDEA Community Edi	tion insta	llation	
Create Desktop Shortcut          32-bit launcher       64-bit launcher         Update context menu       Add "Open Folder as Project"	Update PATH varia	ble (rest	art neede PATH	d)
Create Associations				
Download and install 32-bit JetBrains Runtime	< Back Next	>	Cance	el

Рисунок 7 – Настроювання конфігурації середовища

Далі відкриється вікно для вибору каталогу в меню **Пуск**, де можна буде знайти програму (рис. 8).

Choose Start Menu Folder Choose a Start Menu folder for the IntelliJ IDEA Community Edition shortcuts. Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. International Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. (International Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. (International Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. (International Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. (International Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. (International Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. (International Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. (International Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. (International Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. (International Start Menu folder in which you would like to create the program's shortcuts. (International Start Menu folder in which you would like to create the program's shortcuts. (International Start Menu folder in which you would like to create the program in the program	IntelliJ IDEA Comm	unity Edition Se	etup			—		$\times$
Choose a Start Menu folder for the IntelliJ IDEA Community Edition shortcuts. Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder. IetBrains 7-Zip Accessibility Accessories Administrative Tools Android SDK Tools Android St Tools		Choose	e Start Me	enu Folder	·			
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.          JetBrains         7-Zip         Accessibility         Accessories         Administrative Tools         Android SDK Tools         Android Studio	1.5 	Choose Edition s	a Start Mer hortcuts.	nu folder for	the Intel	iJ IDEA Co	ommunity	
JetBrains 7-Zip Accessibility Accessories Administrative Tools Android SDK Tools Android Studio	Select the Start Menu can also enter a name	folder in which y to create a new	you would I v folder.	like to creat	e the prog	ram's shor	rtcuts. Yo	u
7-Zip Accessibility Accessories Administrative Tools Android SDK Tools Android Studio	JetBrains							
Accessibility Accessories Administrative Tools Android SDK Tools Android Studio	7-Zip							^
Accessories Administrative Tools Android SDK Tools Android Studio	Accessibility							
Administrative Tools Android SDK Tools Android Studio	Accessories							-
Android Studio	Administrative Tools							
	Android Studio							
Apache Tomcat 9.0 Tomcat9	Apache Tomcat 9.01	Fomcat9						

Рисунок 8 – Вибір каталогу для меню Пуск

Можна оставити значення за замовчуванням і натиснути кнопку **Install**. Після закінчення установки можна запустити середовище. Для цього потрібно відзначити на фінальному вікні пункт **Run IntelliJ IDEA Community Edition** і натиснути на кнопку **Finish**.

## **СТВОРЕННЯ ПЕРШОГО ДОДАТКУ НА ANDROID**

Розробка додатку на мові Java

### Створення нового Android-проекту за допомогою майстра

Для створення нового проекту треба натиснути Start a new Android Studio project (рис. 9) або вибрати пункт меню File $\rightarrow$  New  $\rightarrow$  New Project, якщо середовище розробки вже запущене.

Далі вказуємо назву програми і сайт фірми, яка розробляє додаток. Це необхідно для формування назв пакета. Зверніть увагу, що назва пакета будується з сайту фірми і назви програми. Ім'я пакета повинно бути унікальним, тому що за допомогою пакетів додаток однозначно ідентифікується. Також, внизу необхідно вказати папку, де буде зберігатися проект (рис. 10).

Далі, потрібно вказати цільове пристрій для додатка (в нашому випадку це телефони та планшети), а також мінімальний рівень АРІ (рис. 11).



#### Рисунок 9 – Запуск нового проекту наприкінці інсталяції

Application name:	My Application	
<u>Company</u> Domain:	орилиа	
Package name:	ua.opu.myapplication	Edit
Project location:	C:\Users\NickGodov\OneDrive\Android\Projects\MyApplication	

#### Рисунок 10 – Визначення назви проекту та шляху до нього

Phone and Table	et	
Minimum SDK	API 16: Android 4.1 (Jelly Bean)	<b>•</b>
	Lower API levels target more devices, but have fewer features available.	
	By targeting API 16 and later, your app will run on approximately <b>94,8%</b> of the devices that are active on the Google Play Store.	
	Help me choose	

Рисунок 11 – Визначення цільового пристрою та мінімального рівня АРІ

На наступному етапі потрібно вибрати, як буде виглядати стартове вікно програми. Зазвичай можна вибрати **Empty Activity**.

Після цього відобразиться вікно налаштування нового проекту (рис. 12).

👅 Create New Project		×
Configure Your I	Project	
	<u>N</u> ame	
	HelloApp	
÷	 Package name	
	com.example.helloapp	
	Save location	
	C:\Users\Eugene\AndroidStudioProjects\Java\HelloAp 🚞	
	Language	
	Java	
	Minimum SDK API 21: Android 5.0 (Lollipop)	
Empty Activity	1 Your app will run on approximately <b>94,1%</b> of devices. Help me choose	
Creates a new empty activity	Use legacy android.support libraries 📀	
выберем Java	Previous Next Cancel	sh

Рисунок 12 – Вибір конфігурації нового проекту

У вікні створення нового проекту ми можемо встановити його початкові настройки:

- В поле **Name** вводиться назва програми. Наприклад, можна задати в якості імені назву **HelloApp**
- В поле **Package Name** вказується ім'я пакета, де буде розміщуватися головний клас додатка. В даному випадку для тестових проектів це значення не грає великого значення, тому встановимо **com.example.helloapp**.
- В поле **Save Location** встановлюється розташування файлів проекту на жорсткому диску. Можна залишити значення за замовчуванням.

- В поле Language в якості мови програмування можна вказати Java або Kotlin (будьте уважні, так як за замовчуванням в цьому полі стоїть Kotlin)
- В поле Minimum SDK вказується сама мінімальна підтримувана версія SDK. Залишимо значення за замовчуванням - API 21: Android 5.0 (Lollipop), яка означає, що наш додаток можна буде запустити починаючи з Android 5.0, а це 94% пристроїв. На старіших пристроях запустити буде не можна.

Варто враховувати, що чим вище версія SDK, тим менше діапазон підтримуваних пристроїв.

### Структура Android-проекта

Після натискання на кнопку **Finish**, Android Studio згенерує новий проект зі стартовим екраном (рис. 13)



#### Рисунок 13 – Вікно Android Studio

Як ми бачимо, крім стандартного меню вгорі, а також ряду кнопок з найбільш важливими і часто вживаними операціями, у нас є основна область, де відбувається робота з проектом (написання вихідного коду, редагування макета, XML-файлів, тощо), а також є бічні закладки, які можна показати і заховати кліком миші.

Натиснемо на вкладку Project і виберемо пункт **Project Files**. Вкладка Project дозволяє нам подивитися структуру проекту.

Проект складається з декількох папок та файлів. Проект Android може складатися з різних модулів. За замовчуванням, коли ми створюємо проект, створюється один модуль – арр (рис. 14).



Рисунок 14 – Структура проекту Android

Модуль арр має три папки:

- manifests: зберігає файл маніфесту AndroidManifest.xml, який описує конфігурацію програми та визначає кожен з компонентів цього додатка.
- **java**: зберігає файли коду на мові java, які структуровані по окремих пакетах. Так, в папці **com.android.helloapp** (назва якого було вказано на етапі створення проекту) є за замовчуванням файл MainActivity.java

з кодом на мові Jaba, який представляє клас MainActivity, що запускається за замовчуванням при старті додатка

- **res**: містить використовувані в додатку ресурси. Всі ресурси розбиті на підпапки.
  - о папка **drawable** призначена для зберігання зображень, використовуваних в додатку
  - папка layout призначена для зберігання файлів, що визначають 0 графічний інтерфейс. 3a замовчуванням тут € файл activity main.xml, який визначає інтерфейс класу ДЛЯ MainActivity у вигляді xml
  - о папки **тіртар** містять файли зображень, які призначені для створення іконки програми при різних дозволах екрану.
  - папка values зберігає різні xml-файли, що містять колекції ресурсів різних даних, які застосовуються в додатку. За замовчуванням тут є два файли і одна папка:
    - файл colors.xml зберігає опис квітів, використовуваних в додатку
    - файл strings.xml містить рядкові ресурси, використовувані в додатку
    - папки themes зберігає дві теми програми для світлу (денну) і темну (нічну)

Окремий елемент Gradle Scripts містить ряд скриптів, які використовуються при побудові програми.

У всій цій структурі варто виділити файл **MainActivity.java**, який відкритий в Android Studio і який містить логіку програми та власне з нього починається виконання програми. І також виділимо файл activity\_main.xml, який визначає графічний інтерфейс – по суті те, що побачить користувач на своєму смартфоні після завантаження програми.

Каталог gen в Android-проект містить генеровані значення. Зокрема, файл *R.java* – генерований клас, який містить посилання на ресурси з папки res проекту. Ці ресурси містяться в директорії res і можуть бути XMLфайлами, значеннями, меню, схемами, значками, малюнками або анімаціями.

При створенні нових ресурсів, відповідні посилання будуть автоматично створені в *R.java*. Посилання є статичними значеннямиінтервалами, система Android представляє методи доступу до відповідних ресурсів. Наприклад, для доступу до рядка з ідентифікатором посилання *R.string.yourString* потрібно використовувати метод *getString* (*R.string.yourString*). Не рекомендується вручну змінювати файл *R.java*. *AndroidManifest.xml*. У цьому файлі повинні бути оголошені всі Activity, служби, приймачі і контент-провайдери додатки. Також він повинен містити необхідні додатком дозволи. Наприклад, коли програмі потребується доступ до мережі, це повинно бути визначено тут. AndroidManifest.xml можна розглядати як опис для розгортання Android-додатку.

Програма Hello, World! вже вбудована в будь-який новий проект. Просто потрібно запустити проект і отримати готову програму.

Розкрийте папку *src* і знайдіть файл *MainActivity.java*. Двічі клацніть на файлі, щоб відкрити його в редакторі коду. У файлі *MainActivity.java* вже є мінімальний код, згенерований середовищем розробки.

Раскройте папку *src* и найдите файл *MainActivity.java*. Дважды щелкните на файле, чтобы открыть его в редакторе кода. В файле *MainActivity.java* уже есть минимальный код, сгенерированный средой разработки.

Розглянемо цей код:



Перед вами відкритий файл класу, де ім'я класу збігається з ім'ям файлу з розширенням java (це правило, встановлене мовою Java). У першому рядку йде назва пакета – він задавався при створенні пакета (Package Name). Далі йдуть рядки імпорту необхідних класів для проекту.

Далі йде оголошення самого класу, який успадковується (extends) від класу AppCompatActivity. У самому класі бачимо метод onCreate () – він викликається, коли додаток створює і відображає Activity.

#### Сборка проекта

У складі Android SDK присутній система збирання додатку, за допомогою якої здійснюється складання, запуск, тестування і упаковка додатки. Система збирання збірки може бути запущена або з Android Studio, або за допомогою командного рядка.

Система збирання має такі переваги:

- кастомизація, настройка і розширення процесу складання;
- створення множинних настроюються АРК-файлів для застосування для одного і того ж додатка;
- повторне використання коду і ресурсів.

### Запуск проекту

Для запуска проекту вибираємо в меню *Run / Run*. При спробі запустити проект, Android Studio просить вазати пристрій, на якому буде запущений додаток. Це може бути як реальний фізичний пристрій, так і віртуальний пристрій.

	Nothing to	show	
Launch emulator			
Android virtual device:	Nexus 5 API 23 x86		-

Для прикладу використаємо емулятор Genymotion. Спочатку запускаємо новий пристій від Genymotion, після чого намагаємось запустити Android-проект.

	E	State	Compatible	Serial Number
Genymotion Google Nex	us S - 4.1.1 - API 16 - 48	Online	Yes	192.168.56.1
) Launch emulator				
Launch emulator	Nexus 5 API 23 x86			•

Як бачимо, Android Studio розпізнав запущений емулятор і вказав його в списку доступних пристроїв. Натискаємо ОК. Через деякий час, ми побачимо, що наше додаток було успішно встановлено на віртуальний пристрій.





Знайдіть в проекті папку res\layouts і відкрийте файл activity\_main.xml.

Перед вами відкриється редактор макетів. Він дозволяє налаштовувати макет вікна як в графічному, так і в текстовому режимі.



Використовуйте графічний режим (вкладка Design), знайдіть кнопку і перетягніть її в вікно.



Тепер у нас у вікні є кнопка. Але якщо ми на неї натиснемо, то нічого не станеться, тому що ми не прописали реакцію на натискання кнопки.

Для роботи з елементами GUI в Android використовується стандартний патерн (шаблон) «Слухач».

#### Робимо наступне:

1. В MainActivity.java додаємо наступний метод

Зверніть увагу на сигнатуру методу. Метод повинен бути public void і приймати на вхід об'єкт view класу View.

2. У activity\_main.xml виділяємо справа вгорі текстовий напис і присвоюємо їй який-небудь **id** 

nActivity.java ×		
📲 🔚 Nexus 4 + 🛛 🗧 🕖 AppTheme 🔤 MainActivity + 🌀 + 🎁 23 +	Component Tree	호 축   추· 커
] 꽃 ⊠ •	<ul> <li>Device Screen</li> <li>RelativeLayout</li> </ul>	1
	Ab hello_text (Text	View) - "Hello World!"
My Application	ox button - "New F	3utton".
	Properties	? 5 7
	height	
NEW BUTTON	hint 2	
	hyphenationFrequen	
	id hell	o_text
	importantForAccessibi	
	inputMethod	
	▶ inputType []	
	labelFor	
	letterSpacing	

3. Там же виділяємо нашу додану кнопку і виставляємо властивість onClick. У списку вибираємо наш створений метод



4. Переходимо в MainActivity.java і прописуємо вміст нашого методу

```
public void onNewButtonClick(View view) {
    TextView tv = (TextView) findViewById(R.id.hello_text);
    tv.setText("Hamaro!");
}
```

Щоб поміняти нашу напис Hello World, ми повинні з вихідного коду звернутися до конкретного ресурсу (текстова напис). Щоб це зробити, ми спочатку повинні привласнити цього ресурсу якийсь іd, після чого за допомогою методу **findViewById** класу **Activity** отримати посилання на потрібний ресурс.

Посилання на всі ресурси зберігаються в класі R.java, який генерується за допомогою утиліти аарt. Якщо ми відкриємо файл R.java, то ми зможемо знайти наш ресурс - текстовий рядок з id hello\_text.

```
public static final class id {
    public static final int hello_text=0x7f0c0050;
    public static final int action_bar=0x7f0c0041;
    public static final int action_bar_activity_content=0x7f0c0000;
    public static final int action_bar_container=0x7f0c0040;
    public static final int action_bar_root=0x7f0c003c;
    public static final int action_bar_spinner=0x7f0c0001;
```

Тобто, всередині класу R був згенерований статичний клас id, в якому був згенерований статичний int з iм'ям hello\_text. Таким чином, ми звертаємося до цього ресурсу за допомогою синтаксису **R.id.hello\_text**.

Зверніть увагу, що в коді не вказано – який саме це ресурс - рядок, кнопка, список і тд, просто вказано, що назва hello\_text. Метод **findViewById** () повертає нам посилання на об'єкт класу View. Тому, щоб працювати з цим об'єктом далі, ми має зробити приведення типів - перетворити об'єкт класу **View** в об'єкт класу **TextView**.

```
Tаким чином, рядок
TextView tv = (TextView) findViewById(R.id.hello_text);
```

означає – привласнити посилальної змінної tv класу TextView посилання на об'єкт TextView, який ховається під id hello\_text.

Після виконання цього рядка коду, в змінної tv зберігається посилання на текстове поле. Ми викликаємо метод setText () цього поля. Цей метод встановлює новий текст для нашого текстового поля.

Запускаємо програму, натискаємо на кнопку.



В папці bin проекту можна знайти готовий АРК-файл.

# Завдання на лабораторну роботу:

- 1. Встановити інструментарій для Android-розробника
- 2. Створити додаток, описане в ході роботи.

3. В ході здачі лабораторної роботи, бути готовими відповісти на теоретичні питання.

# Теоретичні питання:

1. Які базові інструменти розробника необхідно встановити для створення додатків для OC Android?

- 2. Що таке Activity?
- 3. Що знаходиться в папках gen, src i res?
- 4. Навіщо потрібен AndroidManifest.xml?
- 5. Що таке ресурси? Які ви знаєте ресурси?
- 6. Що зберігається в файлі R.java?
- 7. Яким чином можна отримати посилання на ресурс у вихідному коді?

# ПЕРЕЛІК ПОСИЛАНЬ

- 1. Программирование под Андроид на Java. URL: <u>https://metanit.com/java/android/</u>
- 2. Освой программирование играючи. Сайт Александра Климова. URL: <u>http://developer.alexanderklimov.ru/android/</u>
- 3. Введение в мобильную разработку для Android: с каких языков начать изучение? URL: <u>https://tproger.ru/articles/introduction-to-mobile-</u> <u>development/</u>
- 4. С чего начать изучение Android-разработки/ URL: <u>https://javarush.ru/groups/posts/2821-s-chego-nachatjh-izuchenie-android-razrabotki</u>