

Лабораторна робота №3. Робота з компонентами Activity та Intent

Мета: створити проект з використанням віджетів та дослідити структуру Android проекту.

Теоретичні відомості

Android проект. Зазвичай Android проект містить два вузла, які можна переглянути у Project window в Android Studio – це **app** і **Gradle Scripts** (рис. 1). Вузол **app** містить всі компоненти проекту. Вузол **Gradle Scripts** містить скрипти **Gradle** (системи автоматичного збирання, яка використовується у Android Studio), які використовуються при компілюванні проекту.

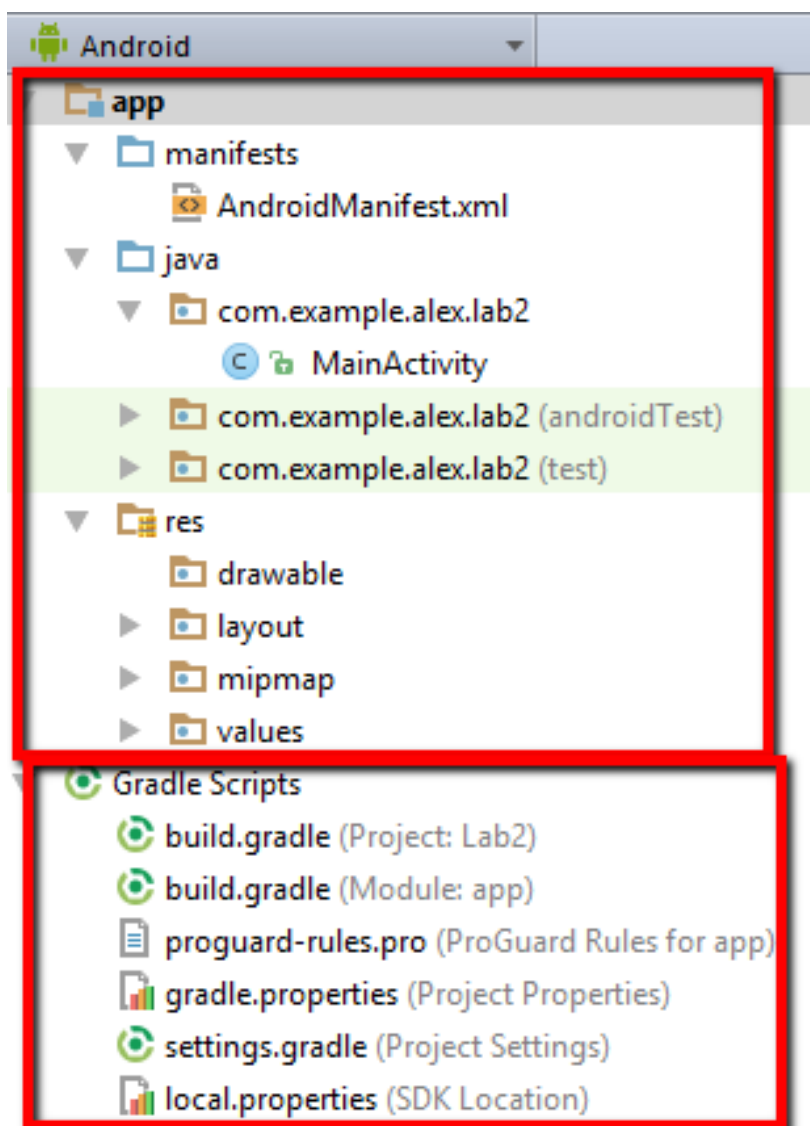


Рис. 1

У вузлі **app** виділяються наступні модулі:

- **manifests**. Містить файл **AndroidManifest.xml**, який описує
- **java**. Містить Java класи проекту.
- **res**. Містить файли ресурсів проекту. Каталог **res** може містити наступні підкаталоги, які відповідають різним типам ресурсів:
 - **animator/**, **anim/**. Файли XML, які визначають анімації.
 - **color/**. Файли XML, які визначають список станів кольорів.
 - **drawable/**. Файли растрових зображень, або файли XML.
 - **ipmap/**. Графічні файли для іконок запуску з різною щільністю.
 - **layout/**. Файли XML, які визначають макет користувацького інтерфейса.
 - **menu/**. Файли XML, які визначають меню додатка, такі як меню параметрів, контекстні меню і т.д.
 - **raw/**. Довільні файли, які треба зберегти у вихідній формі.
 - **values/**. Файли XML, які містять такі значення, як строки, числа, кольори і т.д.
 - **xml/**. Довільні файли XML, які можна читати під час виконання за допомогою метода **Resources.getXML()**.

R клас. Android Studio генерує Java клас, який має назва **R**, його можна знайти у директорії **app/build/generated/source** проекту. В класі **R** описано ID всіх наявних в проекті ресурсів. Кожного разу, коли в проекті додаються, редагуються або видаляються ресурси клас **R** оновлюється.

Життєвий цикл Activity. **Activity** можна описати як вікно, що містить елементи графічного інтерфейсу, з якими може взаємодіяти користувач.

Перше вікно, яке створює додаток називається **main activity**. **main activity** служить точкою входу в додаток. При цьому, додаток може мати декілька **activity** і декларування **main activity** проводиться в файлі маніфесту. В свою чергу, кожна **activity** може запускати інші **activity** для виконання різних задач. Кожного разу, коли запускається нова **activity**, попередня активність зупиняється, однак система зберігає в стек.

Коли користувач викликає додаток, операційна система шукає **main activity** додатку і запускає його. Початок роботи **activity** тягне за собою виклики методів життєвого циклу, які визначені в класі цього **activity**. Нижче перераховані методи життєвого циклу:

- **onCreate**.
- **onStart**.
- **onResume**.
- **onPause**.
- **onStop**.
- **onRestart**.
- **onDestroy**.

Життєвий цикл **Activity** зображено на рис. 2.

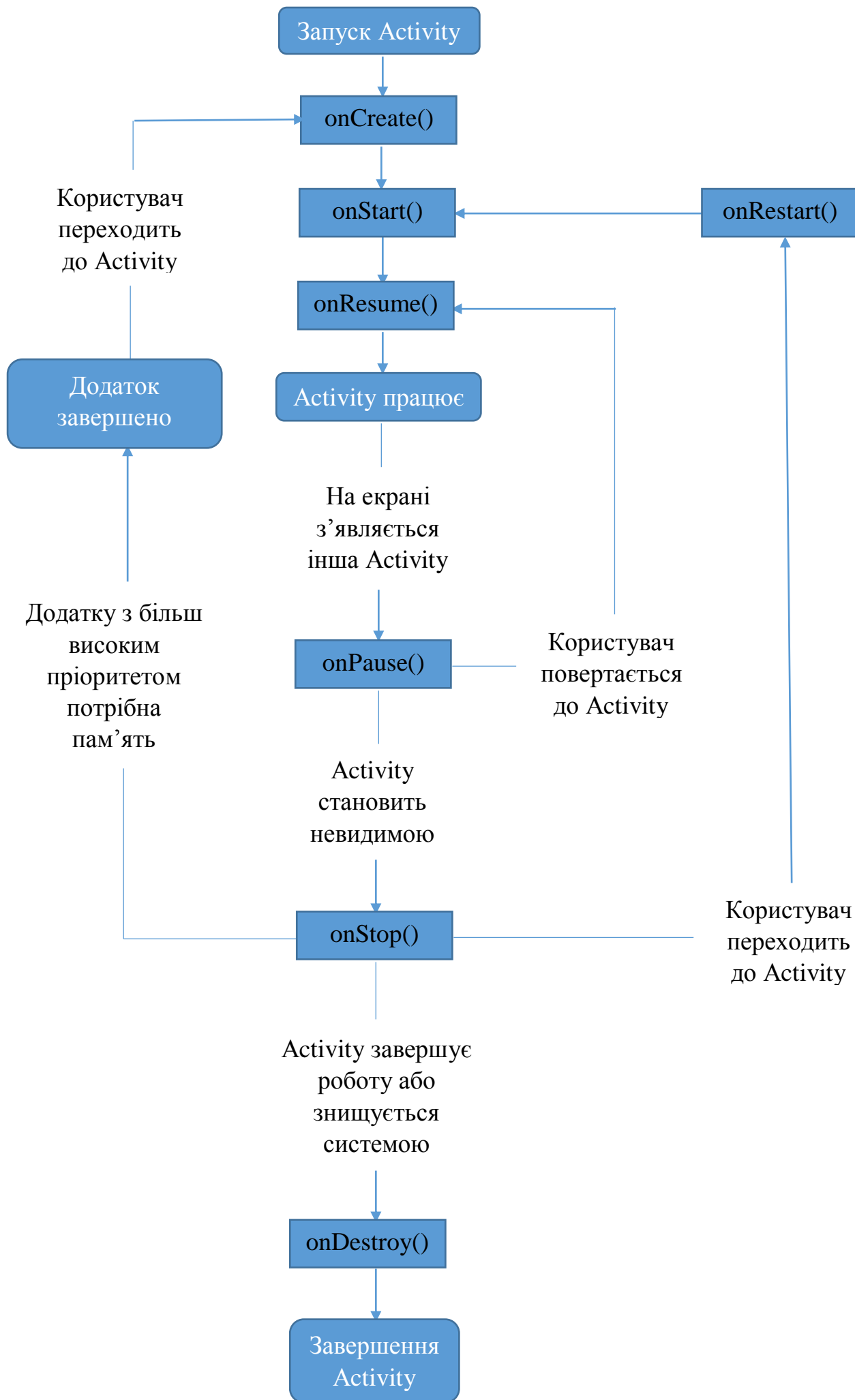


Рис. 2

Система починає з виклику методу **onCreate** для створення activity. У цей метод слід помістити інструкції конструювання UI. Як тільки метод **onCreate** завершено, activity знаходиться у стані **Created**. Цей метод викликається один раз протягом життєвого циклу.

Далі операційна система викликає метод **onStart**. Коли цей метод викликано, activity становиться видимою. Після закінчення цього методу activity має стан **Started**. Цей метод може викликатися декілька разів під час життєвого циклу.

Метод **onResume** викликається перед тим, як activity починає взаємодіяти з користувачем. На цьому етапі activity знаходиться на вершині стека activity і в неї поступають дані, які вводить користувач. Стан activity після цього методу – **Resumed**. Метод **onResume** може викликатися декілька разів.

Отже, методи **onCreated**, **onStart** та **onResume** будуть визиватися послідовно у тому випадку, якщо не буде непередбачених виключень. У стані **Resumed** activity знаходиться у запущеному стані і залишається в ньому до тих пір, як якась подія не змінить його. Це може бути виклик, перехід пристрою у стан сну або запуск іншої activity.

onPause метод викликається тоді, коли activity змінює стан **Resumed**. Після закінчення методу activity переходить у стан **Paused**. **onPause** може викликатися багато разів протягом життєвого циклу.

Що далі відбувається після закінчення методу **onPause** залежить від того, чи буде далі activity повністю невидимою. Якщо так, викликається метод **onStop** і activity переходить до стану **Stopped**. Якщо activity знову викликається після **onPause**, система викликає метод **onResume** і activity знову переходить до стану **Resumed**.

Зі стану **Stopped** activity може бути відновлена, якщо користувач захочу повернутися до activity. В цьому випадку викликаються послідовно методи **onRestart** та **onStart**.

Нарешті, коли activity остаточно виведено з експлуатації, викликається метод **onDestroy**. Цей метод може визиватися один раз протягом життєвого циклу.

Виклик іншої activity. В додатках з декількома активностями виклик однієї activity з іншої виконується за допомогою метода **startActivity**:

```
startActivity(intent);
```

де **intent** – це об'єкт класа **android.content.Intent**.

Припустимо, що треба написати додаток, у якому з головної активності запускається друга активність при натисканні кнопки та відображається відповідне повідомлення у віджеті TextView другої активності.

Файл **AndroidManifest.xml** у випадку використання в проєкті двох activity може мати наступний вигляд:

```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.alex.lab3">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        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>
        <activity android:name=".secondActivity"></activity>
    </application>

</manifest>
```

Для обробки події натискання кнопки в класі MainActivity експортовано клас **OnTouchListener** та перевизначено метод **onTouch**. В цьому методі створюється об'єкт класа **Intent** та поміщається повідомлення методом **intent.putExtra**. Тепер можна використовувати метод **startActivity** для запуску іншої activity

В методі **onCreate** другої activity викликається метод **getIntent** який отримує повідомлення від метода **getStringExtra**. Яке потім можна використовувати у методі **setText** віджета **TextView**. Це можна зробити за допомогою наступної команди:

```
((TextView)findViewById(R.id.textView1)).setText(message)
```

Об'єкти Intent, пов'язані з Activity. Таким чином для запуску нового activity потрібно:

```
Intent intent = new Intent(this, SecondActivity.class);
startActivity(intent);
```

Якщо треба передати

```
Intent intent = new Intent(this, SecondActivity.class);
intent.putExtra("message", "Message from First Screen");
startActivity(intent);
```

Intent представляє собою об'єкт обміну повідомленнями, за допомогою якого можна видати запит на виконання дії у іншого компонента або компонента іншого додатку.

Об'єкти **Intent** зазвичай використовуються у наступних випадках:

- Для виклику activity. Використовуються методи startActivity() або startActivityForResult()/onActivityResult().
- Для виклику служб (Service). Використовуються методи startService() та bindService().
- Для розсилки широкомовних повідомлень. Використовуються методи sendBroadcast(), sendOrderedBroadcast() или sendStickyBroadcast().

Є два типа об'єктів **Intent**:

Явні об'єкти **Intent** вказують компонент, який потрібно запустити по імені класа. У попередньому прикладі використовувався явний Intent, тобто такий, у якому явно прописана activity, якій він спрямований.

Неявні об'єкти **Intent** не містять імені конкретного компонента. Замість цього, вони визначають дію, яку потрібно виконати. Система сама вирішує, для яких компонентів спрямовано Intent шляхом порівняння змісту об'єкта Intent з фільтрами Intent, які визначені в файлах маніфеста інших додатків. Якщо об'єкт Intent співпадає з фільтром Intent, система запускає цей компонент і передає йому об'єкт Intent.

Фільтр Intent представляє собою вираз в файлі маніфеста додатку, який вказує типи об'єктів Intent, які може прийняти компонент.

Таблиця 1 містить назви можливих дій, які можуть вказуватись в об'єкті Intent спрямованому activity.

Дія	Опис
ACTION_MAIN	Запускає activity та визначає її як точку входу в додаток.
ACTION_VIEW	Переглядає дані, додані до Intent
ACTION_ATTACH_DATA	Додає дані, які були додані до Intent у іншому компоненті.
ACTION_EDIT	Редагує дані додані до Intent.

ACTION_PICK	Вибрати пункт з даних.
ACTION_CHOOSER	Показати всі додатки, які реагують на Intent.
ACTION_GET_CONTENT	Дозволяє користувачу вибрати певні дані та передати їх.
ACTION_DIAL	Набрати номер, доданий до Intent.
ACTION_CALL	Зателефонувати персоні, вказаній у Intent.
ACTION_SEND	Відіслати дані, додані до Intent.
ACTION_SENDTO	Відіслати повідомлення персоні, вказаній у Intent.
ACTION_ANSWER	Відповісти на вхідний дзвінок.
ACTION_INSERT	Вставити пустий об'єкт у контейнер.
ACTION_DELETE	Видалити вказані дані з контейнера.
ACTION_RUN	Запустити додані дані.
ACTION_SYNC	Виконати синхронізацію даних.
ACTION_PICK_ACTIVITY	Вибрати activity зі списку activity.
ACTION_SEARCH	Виконати пошук, використовуючи строки пошуку з Intent.
ACTION_WEB_SEARCH	Виконати пошук у WEB, використовуючи строки пошуку з Intent.
ACTION_FACTORY_TEST	Визначити як точку входу для factory test.

Завдання до лабораторної роботи

1. Створити додаток у Android Studio з двома activity.

Вимоги до інтерфейсу:

- MainActivity містить кнопку “Start second activity”.
- SecondActivity містить віджет Plain TextView.

Вимоги до функціоналу:

- При натисканні на кнопку “Start second activity” відкривається SecondActivity та в Plain TextView відображається повідомлення, що було надіслане в intent.

Контрольні запитання

1. Які типи Intent ви знаєте?
2. Яку роль у додатку відіграють екземпляри класа Intent?
3. Призначення класа OnClickListener?