

Лабораторна робота №9. Життєвий цикл activity

Мета: вивчити порядок роботи з методами життєвого циклу activity в Android.

Теоретичні відомості

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

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. 1.

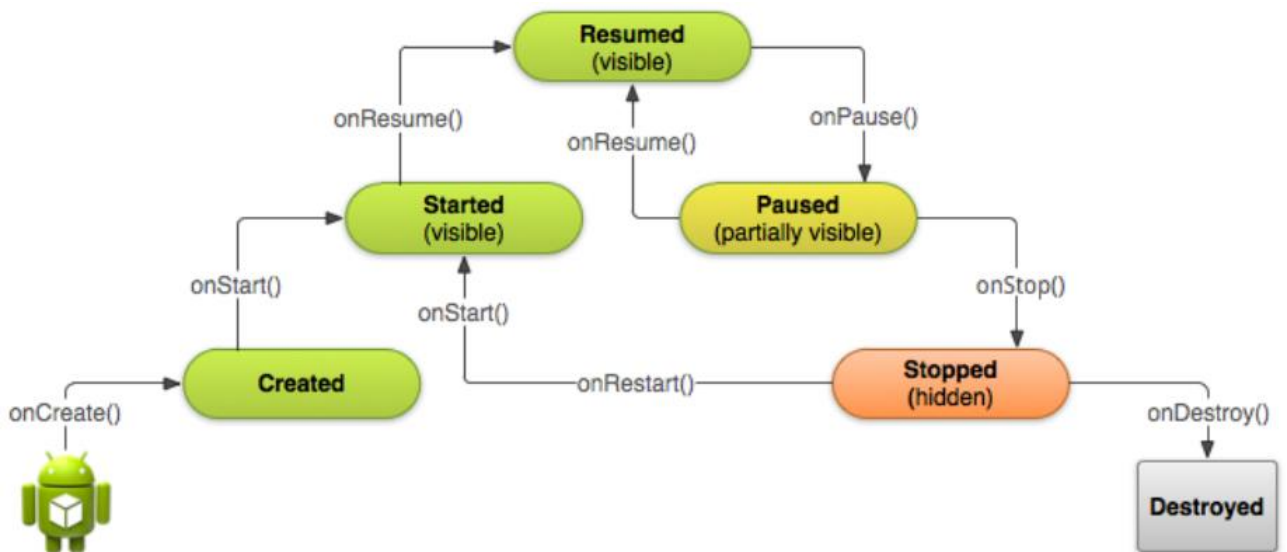


Рис. 1

Або більш докладно у вигляді блок-схеми (рис. 2).

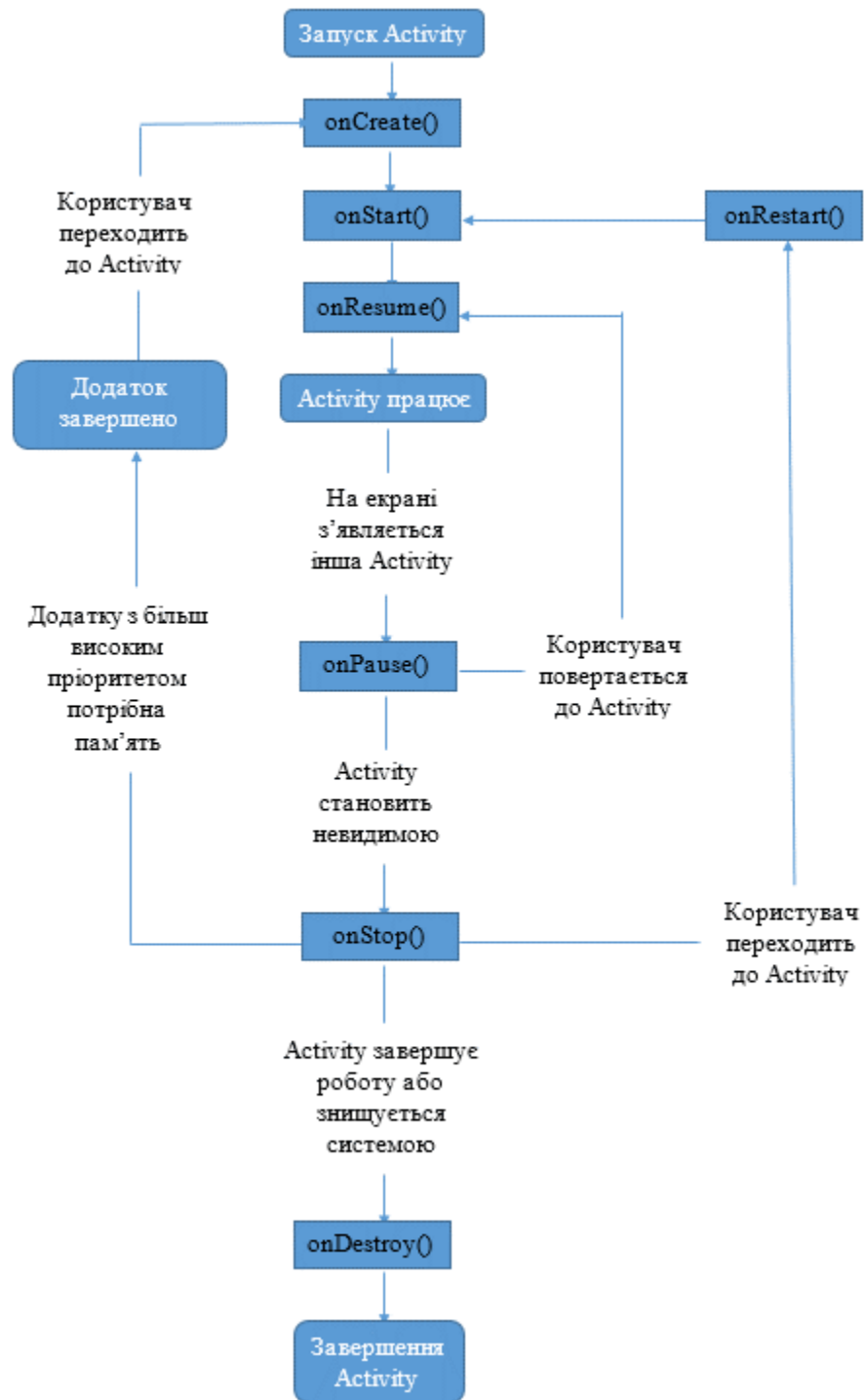


Рис. 2

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.

Загалом можна виділити наступні стани activity.

Resumed (Started). Користувач взаємодіє з activity.

Paused. Користувач не може взаємодіяти з activity, але його може бути видно. Наприклад, activity перекрито вікном іншого activity.

Stopped. Користувач не може взаємодіяти з activity, і його не видно. В цей час activity в фоновому режимі.

Під час роботи додатку саме зміна станів activity викликає ті, чи інші методи зворотного виклику.

Правильна реалізація методів життєвого циклу передбачає виконання наступних умов.

1. Додаток не завершує роботу, якщо користувач отримує дзвінок або перемикається на інший додаток.
2. Додаток не використовує системні ресурси, коли в цьому немає потреби.

3. Зберігається стан додатку, коли користувач виходить з нього і повертається пізніше.
4. Додаток не закривається з помилкою та не губить дані користувача при зміні стану пристрою (наприклад, при повороті).

Головна activity. Після запуску додатку система викликає метод **onCreate()** тієї activity яка була задекларована як основна, тобто є точкою входу в програму.

Для визначення основної активності використовується файл маніфесту **AndroidManifest.xml**. Основна активність додатку декларується в файлі маніфесту за допомогою фільтра **<intent-filter>**, який містить дію **MAIN** і категорію **LAUNCHER** (рис. 3).

```
<activity android:name=".MainActivity" android:label="@string/app_name">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Рис. 3

Завдання до лабораторної роботи

1. До текстового редактору, створеного в лабораторній роботі №8 додати activity з текстом ліцензійної угоди. Передбачити наявність контейнера **ScrollView**. Передбачити наявність двох конопок «Cancel» і «Next». Після натискання кнопки «Cancel» завершити роботу. Після включення прапорця «agree» активувати кнопку «Next» переходу до activity з реєстраційною формою.
2. Реєстраційна форма аналогічна формі завдання 11 лабораторної роботи №5.
3. При переході між різними activity відображати toast повідомлення з назвою активності та викликаного метода життєвого циклу.

Контрольні запитання

1. Назвіть методи життєвого циклу activity?
2. Які методи життєвого циклу відповідають за створення та відображення activity?
3. Які методи життєвого циклу відповідають за призупинення та знищення activity?

