

Лабораторна робота №10. Життєвий цикл activity2

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

Теоретичні відомості

Activity Stack. Для кожного додатку на Android система підтримує стек активностей. Після запуску додатку, перша з activity заноситься на вершину стеку. Після запуску другої активності, вона поміщається на вершину стека і т.д. При натисканні кнопки «Назад» попередня активність з стеку відновлюється (рис. 1).

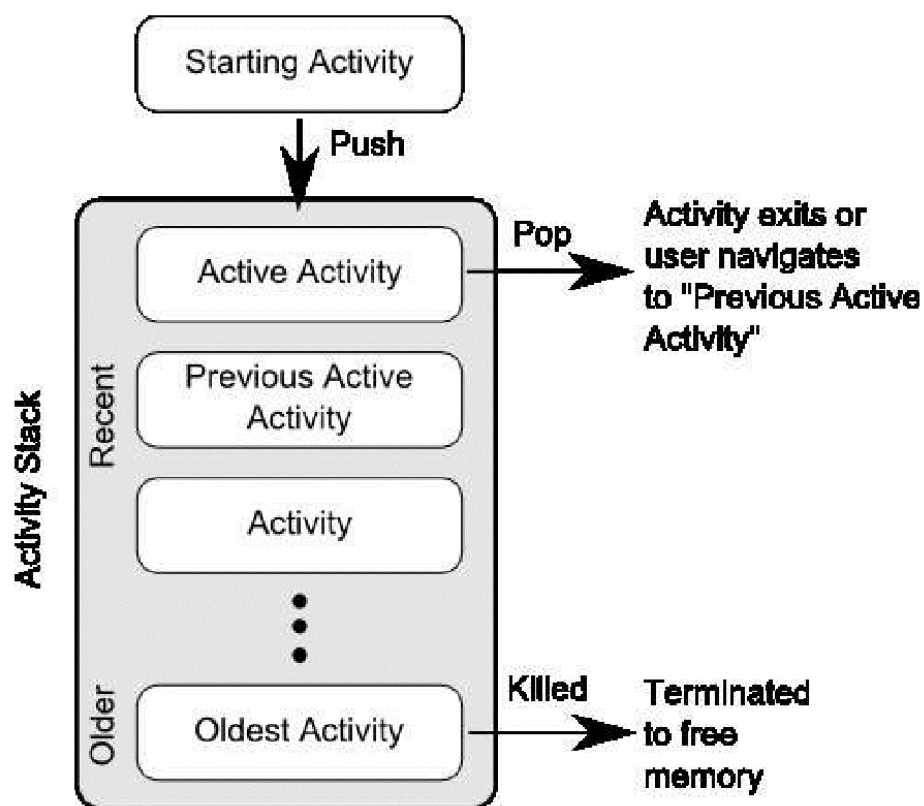


Рис. 1

Збереження activity. Активність може завершити свою роботу планово – в результаті виклику метода **finish()**, або система може знищити activity, якщо вона зупинена і не використовується довгий час, або якщо активності потрібно ресурсів більше ніж наявно в системі в даний момент.

У разі, якщо Android примусово завершує роботу додатку, система запам'ятовує стан активності та при відновленні додатка, відновлює збережений стан.

Системні дані, що використовуються системою для відновлення попереднього стану зберігаються у вигляді набору пар «ключ-значення» та містяться у об'єкті **Bundle**.

Наприклад, activity знищується та відновлюється системою при повороті екрана. Це пов'язано з потенційною необхідністю завантажити інше представлення та пов'язані ресурси при зміні орієнтації екрана. Аналогічна ситуація виникає при будь-якій зміні конфігурації пристрою або зовнішніх перериваннях.

Для збереження додаткових параметрів стану активності необхідно перевизначити метод зворотного виклику **onSaveInstanceState()**. Система викликає цей метод коли завершується виконання активності та передає йому об'єкт **Bundle** у якому збережено необхідні параметри. При відновленні activity той же самий **Bundle** передається методам **onRestoreInstanceState()** та **onCreate()** (рис. 2).

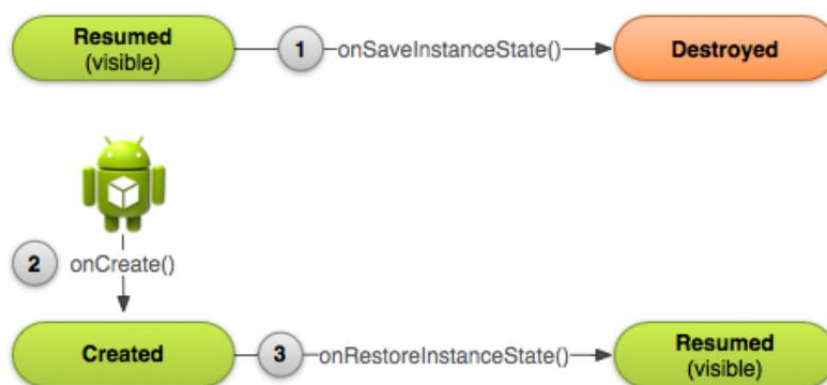


Рис. 2

Для збереження стану активності необхідно перевизначити метод **onSaveInstanceState()** та додати до об'єкта **Bundle** необхідні пари «ключ-значення» (рис. 3).

```
static final String STATE_SCORE = "playerScore";
static final String STATE_LEVEL = "playerLevel";
...

@Override
public void onSaveInstanceState(Bundle savedInstanceState) {
    // Save the user's current game state
    savedInstanceState.putInt(STATE_SCORE, mCurrentScore);
    savedInstanceState.putInt(STATE_LEVEL, mCurrentLevel);

    // Always call the superclass so it can save the view hierarchy state
    super.onSaveInstanceState(savedInstanceState);
}
```

Рис. 3

Відновлення activity. При відновленні стану активності, попередній стан можна відновити з об'єкту **Bundle**, куди система передає дані activity. Методи **onRestoreInstanceState()** та **onCreate()** отримують той самий **Bundle**.

Оскільки метод **onCreate()** викликається при створенні нового екземпляру activity або при його відновленні, спочатку потрібно перевірити чи порожній **Bundle** (рис. 4).

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState); // Always call the superclass first

    // Check whether we're recreating a previously destroyed instance
    if (savedInstanceState != null) {
        // Restore value of members from saved state
        mCurrentScore = savedInstanceState.getInt(STATE_SCORE);
        mCurrentLevel = savedInstanceState.getInt(STATE_LEVEL);
    } else {
        // Probably initialize members with default values for a new instance
    }
    ...
}
```

Рис. 4

Альтернативним способом є перевизначення метода **onRestoreInstanceState()**, який викликається системою після метода **onStart()**. При цьому, система викликає **onRestoreInstanceState()** тільки при наявності збереженого стану. Тобто, відпадає необхідність перевірки чи порожній **Bundle** (рис. 5).

```
public void onRestoreInstanceState(Bundle savedInstanceState) {
    // Always call the superclass so it can restore the view hierarchy
    super.onRestoreInstanceState(savedInstanceState);

    // Restore state members from saved instance
    mCurrentScore = savedInstanceState.getInt(STATE_SCORE);
    mCurrentLevel = savedInstanceState.getInt(STATE_LEVEL);
}
```

Рис. 5

Завдання до лабораторної роботи

1. До текстового редактору, створеного в лабораторній роботі №8,9 додати можливість відновлювати роботу activity таким чином, щоб після повороту екрана всі налаштування форматування тексту та фону відновлювались.
2. При відновленні використовувати 2 способи: відновлення в методі **onRestoreInstanceState()** та **onCreate()**

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?