**ACTIVITY**

**4.1 Компоненти Android-додатку**

Кожен додаток під управлінням ОС Android запускається в своєму власному процесі. Тому додаток ізольовано від інших запущених додатків, і некоректно працюючий додаток не може нашкодити іншим запущеним на виконання додаткам.

Проте, важливою властивістю додатку є можливість використовувати компоненти інших додатків, якщо вони дають на це відповідні права. Припустимо, нам потрібен якийсь компонент з прокруткою для відображення тексту, і схожий компонент вже реалізований в іншому додатку. Тоді у нас є можливість використовувати реалізований компонент. У цьому випадку наш додаток не копіює необхідний код до себе і не створює посилання на нього. Замість цього додаток робить запит на виконання частини коду іншої програми, де є потрібний нам компонент.

Виділяють чотири типи компонентів Android-програми: *Activities*, *Services*,

*Broadcast receivers* і *Content providers*.

Також важливо відзначити об’єкти *Intents*, в Android-додатках майже все працює завдяки їм. *Intent* - це механізм для опису однієї операції (вибрати фотографію, відправити лист, зробити дзвінок, запустити браузер і перейти за вказаною адресою тощо). Найбільш поширений сценарій використання *Intent* – запуск іншого *Activity* в додатку.

**4.1.1 Services**

Service – це певний процес, який запускається у фоновому режимі. Наприклад, Service може отримувати дані по мережі, виконувати будь-які тривалі обчислення. Хорошим прикладом Service служить програвач музики. Користувач може вибрати будь-яку пісню в програвачі, включити її і закрити плеєр. Музика буде програватися в фоновому процесі. Service для програвання музики буде працювати, навіть якщо Activity плеєра буде закритою.

Подібно до Activity, Service має свої методи життєвого циклу:

 *void onCreate();*

 *void onStart(Intent intent);*

 *void onDestroy().*

У повному життєвому циклі Service існує два вкладених цикли:

1. Повне життя Service – це проміжок між моментом виклику методу *onCreate()* і моментом повернення *onDestroy()*. Подібно до Activity, для Services здійснюють початкову ініціалізацію в *onCreate()* і звільняють всі зайняті ним ресурси в *onDestroy()*;

2. Активне життя Service – починається з виклику методу *onStart()*. Цьому методу передається об’єкт Intent, який передавався в *startService()*.

Як і Activities, Services запускаються в головному потоці процесу додатку. Тому їх слід запускати в окремому потоці, щоб вони не блокували ні інші компоненти ні інтерфейс користувача.

**4.1.2 Broadcast receivers**

Broadcast receiver – це компонент, який розсилає і реагує на широкомовні повідомлення. Прикладом широкомовних компонентів можуть бути: повідомлення про перехід на літній/зимовий час, повідомлення про мінімальний заряді батареї і т. д.

Broadcast receiver не відображує на екрані користувацький інтерфейс, але може запустити Activity при отриманні певного повідомлення або використовувати NotificationManager для залучення уваги користувача. Привернути увагу користувача можна, наприклад, вібрацією пристрою, програванням звуку або миготінням спалаху.

Приймач широкомовних повідомлень має єдиний метод життєвого циклу: *onReceive*(). Коли широкомовне повідомлення прибуває до одержувача, Android викликає його методом *onReceive*() і передає в нього об’єкт Intent, що містить саме повідомлення. Приймач широкомовних повідомлень є активним тільки під час виконання цього методу. Процес, який в даний час виконує Broadcast receiver, є пріоритетним процесом і буде збереженим, окрім випадків гострої нестачі пам’яті в системі.

Коли програма повертається з *onReceive*(), приймач стає неактивним і система вважає, що робота об’єкта Broadcast receiver закінчена. Процес з активним широкомовним одержувачем захищений від знищення системою. Однак процес, що містить неактивні компоненти, може бути знищений системою в будь-який момент, якщо пам’ять, яку він використовує, буде необхідна для інших процесів.

**4.1.3 Content providers**

Content providers надають доступ до даних (читання, додавання, оновлення). Content provider може надавати доступ до даних не тільки свого додатку, а й інших. Дані можуть розміщуватися в файловій системі чи в базі даних.