

Лабораторна робота 6

РОЗМІТКИ В ANDROID STUDIO

Мета роботи: познайомитися з розмітками в Android Studio, навчитися застосовувати розмітки LinearLayout, FrameLayout, RelativeLayout і ConstraintLayout.

Теоретичні відомості

Розмітка або *макет (Layout)* використовується для розташування дочірніх компонентів на екрані пристрою. Використовуючи вкладені розмітки, можна створювати інтерфейси користувача будь-якої складності.

В Android існують такі основні розмітки:

- RelativeLayout – визначає положення одних View відносно інших.
- LinearLayout – визначає розташування всіх дочірніх елементів по вертикалі або по горизонталі один за одним.
- TableLayout – розглядає дочірні елементи у вигляді рядків і стовпців.
- GridView – елементи представлені в двовимірній сітці з прокруткою. Елементи в GridView потрапляють з призначеного примірника ListAdapter.
- FrameLayout - заповнює собою екран для відображення єдиного View.
- ConstraintLayout - подібно RelativeLayout прив'язує елементи інтерфейсу до контейнера(батьківського елемента), в якому вони знаходяться або до інших елементів цього контейнера і має розширені можливості таких прив'язок.
- CoordinatorLayout - координує взаємне розташування View-елементів, які знаходяться всередині цього View-group
- AppBarLayout - нагадує вертикальний LinearLayout, елементи якого можуть вести себе по-різному (в залежності від прапорів) при прокручуванні вмісту.
- CollapsingToolbarLayout - це ViewGroup, який надає багато візуальних характеристик та взаємодій для згортання панелей інструментів. Використовується разом з AppBarLayout та CoordinatorLayout.

Розмітка *LinearLayout* дозволяє компоувати різні елементи графічного інтерфейсу один за одним вертикально або горизонтально. LinearLayout може бути кореневим елементом макета або бути вкладеним сам в себе стільки разів, скільки потрібно. Також можна взаємодіяти з іншими розмітками, наприклад, RelativeLayout або FrameLayout, де LinearLayout може бути як вкладеним макетом, так і тим, в який вкладається.

LinearLayout дозволяє задавати орієнтацію контенту (за замовчуванням використовується горизонтальна орієнтація), висоту і ширину елементів, встановлювати тип вирівнювання самого елемента або його вмісту, зовнішні та внутрішні відступи з усіх боків і таке інше.

Розмітка *FrameLayout* призначена для виведення на екран одного або кількох візуальних елементів, які по замовчуванню прикріплюються до верхнього лівого кута екрану. Елементи будуть накладатися один на одного, частково або повністю затінюючи

попередніх, якщо зверху буде знаходитися непрозорий елемент.

У компонованнях *RelativeLayout* та *ConstraintLayout* елементи інтерфейсу упорядковуються відносно контейнера (батьківського елемента), в якому вони знаходяться або відносно інших елементів цього контейнера.

ConstraintLayout (Android Studio за замовчуванням пропонує використовувати саме її) має додаткові можливості, особливо корисні при роботі у графічному режимі конструювання (вкладка Design). В Android Studio для цього контейнера є зручний візуальний редактор. Крім того *ConstraintLayout*, в більшості випадків, дає змогу уникнути зайвої вкладеності контейнерів, що позитивно впливає на продуктивність та читабельність коду. Цією розміткою можна, в певній мірі, замінити деякі розмітки, які з'явилися раніше.

У компонованні *ConstraintLayout* існують різні типи прив'язок: по горизонталі, по вертикалі та під певним кутом.

ConstraintLayout можна використовувати разом з *ConstraintSet* для одночасної анімації декількох елементів. *ConstraintSet* можна створити в коді або додати прямо в xml-файл.

До недоліків *ConstraintLayout* можна віднести:

- при великій кількості елементів і взаємозв'язків між ними складно зрозуміти, як вони всі взаємодіють, ніж у випадку ієрархічної вкладеності контейнерів;
- у конкретних випадках ця розмітка за продуктивністю може поступатися попереднім розміткам з вкладеними контейнерами, тому що додатково потрібен час на побудову дерева залежностей.

Завдання

1. Розташувати компоненти графічного інтерфейсу, застосовуючи вкладення розміток *LinearLayout*. Доповнити попереднє розташування додатковими компонентами з використанням розмітки *FrameLayout*.
2. Використовуючи розмітку *RelativeLayout* в XML-форматі, розташувати компоненти графічного інтерфейсу з вирівнюванням відносно батьківського елемента та інших елементів графічного інтерфейсу.
3. Використовуючи розмітку попереднього завдання за допомогою Java-коду видалити один з компонентів і вставити його в іншому місці екрану.
4. Використовуючи розмітку *ConstraintLayout* в режимі Design та XML-форматі, виконати завдання 1. Порівняти можливості *RelativeLayout* та *ConstraintLayout*.
5. За допомогою *ConstraintSet* створити прив'язки, ланцюжки, бар'єри, групи та використати *constraintCircle*. Використати *ConstraintSet* в коді або в xml-файлі.