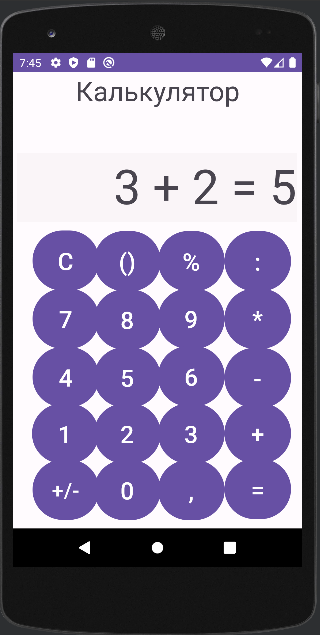
Лабораторная работа №2

Создание пользовательских интерфейсов и использование элементов управления в приложениях под Android

Цель работы: изучить интерфейс IDE Android Studio и получить навыки составления и отладки простого Android приложения с использованием базовых элементов графического интерфейса пользователя.

Ход работы:

В результате работы, был создан интерфейс калькулятора:



Контрольные вопросы

1. Иерархия классов View - ViewGroup:

- View: Основной класс для всех UI компонентов.

- ViewGroup: Абстрактный класс для контейнеров, которые содержат другие View, например, LinearLayout, RelativeLayout, FrameLayout и др.

2. Способы разметки - структуры расположения элементов в окне:

- XML-разметка: Создание макета интерфейса пользователя с использованием XML-разметки.

- Программное создание: Создание макета интерфейса пользователя программным способом с использованием методов Java.

3. Назначение XML-атрибутов в файле разметки:

- Описание структуры интерфейса: Определяются параметры отображения и расположения элементов интерфейса, например, ширина, высота, цвет, отступы и другие свойства виджетов.

4. Стандартные типы разметок:

- LinearLayout: Для вертикального или горизонтального расположения элементов.

- RelativeLayout: Позволяет размещать элементы относительно друг друга или относительно родительского контейнера.

- FrameLayout: Располагает дочерние элементы в области экрана, используя стек.

5. Назначение разметки LinearLayout:

- LinearLayout: Используется для размещения элементов в одной строке (горизонтально) или в одном столбце (вертикально).

6. Основные (базовые) виджеты:

- Button: Кнопка для выполнения действий при нажатии.

- TextView: Для отображения текста.

- EditText: Позволяет пользователю вводить текст.

7. Классы, представляющие текстовые поля:

- EditText: Редактируемое текстовое поле, куда пользователь может вводить текст.

- TextView: Поле для отображения текста (обычно не редактируется).

8. Виджет для отображения графики:

- ImageView: Для отображения изображений.

9. Классы, используемые для кнопок и флажков:

- Button: Стандартная кнопка для выполнения действий.

- CheckBox: Флажок, который может быть установлен или снят.

10. Компоненты для отображения времени:

- TimePicker: Позволяет пользователю выбрать время.

- DatePicker: Позволяет пользователю выбрать дату.

11. Классы для текстовых полей с автозаполнением и возможностью редактирования текста:

- AutoCompleteTextView: Поле ввода с функцией автозаполнения и возможностью редактирования.

12. Элемент для отображения вертикального списка с прокруткой:

- ListView: Используется для отображения большого списка данных с возможностью прокрутки.

13. Назначение элемента Spinner:

- Spinner: Позволяет пользователю выбирать один из нескольких вариантов в выпадающем списке.

14. Возможности элемента GridView:

- GridView: Организует данные в виде таблицы с ячейками одинакового размера.

15. Возможности виджета Gallery:

- Gallery: использовался для отображения изображений в виде горизонтальной прокручиваемой галереи. В настоящее время считается устаревшим.

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

/\* Задание 1

double angle60 = Math.cos(Math.PI / 3);

double angle45 = Math.cos(Math.PI / 4);

double angle40 = Math.cos(Math.toRadians(40));

System.out.println("Косинус угла 60 градусов: " + angle60);

System.out.println("Косинус угла 45 градусов: " + angle45);

System.out.println("Косинус угла 40 градусов: " + angle40);

\*/

/\* Задание 2

Scanner scanner = new Scanner(System.in);

System.out.print("Введите длину первого катета: ");

double a = scanner.nextDouble();

System.out.print("Введите длину второго катета: ");

double b = scanner.nextDouble();

double c = Math.sqrt(a \* a + b \* b); //гипотенуза

double perimeter = a + b + c; //периметр

double area = 0.5 \* a \* b; //площадь

System.out.println("Периметр треугольника: " + perimeter);

System.out.println("Площадь треугольника: " + area);

\*/

/\* Задание 3

Scanner scanner = new Scanner(System.in);

System.out.print("Введите натуральное число: ");

int n = scanner.nextInt();

int count = 0;

while (n > 0) {

count++;

n /= 1;

}

System.out.println("Количество цифр в числе: " + count);

\*/

/\* Задание 4

Scanner scanner = new Scanner(System.in);

System.out.print("Введите число: ");

double n = scanner.nextDouble();

int signum = 0;

if (n < 0) {

signum = -1;

} else if (n > 0) {

signum = 1;

}

System.out.println("Значение функции sgn от числа " + n + " равно " + signum);

\*/

}

}