**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп’ютерних наук та кібернетики

Кафедра теорії та технології програмування

**Звіт**

до лабораторної роботи на тему

**Розв'язування рівнянь чисельними методами**

Виконала студент 4-го курсу

Сушко Діана Валентинівна

**Київ – 2024**

# **Опис завдання**

Метою роботи є формування базових знань та умінь розробляти для мобільного пристрою програми обчислювального характеру, використовувати програмні засоби роботи з файлами та візуалізувати дані.

Загальні вимоги:

* Має бути документ (PDF) з описом вимог до програмного продукту (ТЗ)
* Програма повинна реалізовувати запис/читання даних з файлу або локальної БД
* Програма повинна мати пункт з інформацією про розробника та його/її фото
* Повинні бути застосовані базові принципи UX-дизайну та ергономіки для мобільних пристроїв

Для розробки мобільного додатку було взято 1 варіант (Розв'язування рівнянь чисельними методами).

Умови:

1. **Введення даних**: Обирається кредитна програма, вводяться суми стартового запозичення і розміри періодичного погашення.
2. **Виведення даних**: Сума заборгованості на потрібний період для різних кредитних програм.
3. **Збереження даних**: Зберегти у файлі результати табуляції функцій на введеному проміжку.
4. **Графічна візуалізація**: Графік функції на введеному проміжку на основі даних, записаних у файл.

# 

# **Використані технології**

* Android Studio (IDE): - обрано головною IDE для розробки додатку на мові Java/Kotlin.
* Graphlib: використано graphlib для візуалізації графіків функцій
* Programing language: використано мову програмування Java, яка є однією з основних мов для платформи Android.

# **Опис алгоритмів, які застосовуються для обчислень**

case "Standard":  
 remainingAmount = initialAmount - (monthlyPayment \* 12);  
 break;  
case "Premium":  
 for (int i = 0; i < 12; i++) {  
 remainingAmount -= (monthlyPayment + (monthlyPayment \* 0.05 \* i));  
 }

1. **Алгоритм для "Standard"**

У випадку обраної кредитної програми "Standard" алгоритм розраховує залишкову суму після року платежів за простою формулою:

Формула: remainingAmount = initialAmount - (monthlyPayment \* 12).

Опис: Від початкової суми кредиту (initialAmount) віднімається річна сума платежів (monthlyPayment \* 12). Річна сума платежів – це помісячний платіж, помножений на 12 місяців.

Особливість: У цьому випадку залишкова сума зменшується рівномірно, без урахування процентів чи змінних умов, тобто щомісяця платіж однаковий.

Приклад:

Початкова сума: 12000

Щомісячний платіж: 1000

Після 12 місяців залишкова сума становитиме: 12000 - (1000 \* 12) = 0 (якщо всі платежі зроблені в повному обсязі).

2. **Алгоритм для "Premium"**

У випадку програми "Premium" алгоритм розраховує залишкову суму за допомогою циклу, який враховує збільшення щомісячного платежу на 5% кожного місяця. Це означає, що кожен наступний платіж є на 5% більшим за попередній:

Процес: За допомогою циклу for (int i = 0; i < 12; i++) виконується розрахунок залишкової суми на кожен місяць, змінюючи щомісячний платіж залежно від поточного місяця:

monthlyPayment + (monthlyPayment \* 0.05 \* i), де i – номер місяця (0 – перший місяць, 1 – другий і так далі).

Опис: Кожного місяця платіж зростає на 5% порівняно з початковою сумою. Алгоритм віднімає збільшений щомісячний платіж від залишкової суми на кожному кроці, аж до 12 місяця.

Приклад:

Початкова сума: 12000

Щомісячний платіж: 1000

Перший місяць: 1000, другий місяць: 1000 + 5%, третій місяць: 1000 + 10%, і так далі.

# **Опис роботи з програмою**

Цей Android застосунок призначений для обчислення коренів многочленів за допомогою різних чисельних методів, таких як метод ділення навпіл, метод Ньютона та його модифікація. Додатково, застосунок може будувати графік функції і працювати з файлом для збереження результатів обчислень. Ось детальніше про роботу цього застосунку:

### Основна логіка роботи застосунку:

#### 1. **Головний екран — MainActivity**

На головному екрані користувач може ввести дані, такі як коефіцієнти многочлена, точність, межі для пошуку коренів, а також вибрати метод, яким він хоче знаходити корінь. Головний інтерфейс складається з таких компонентів:Spinner (випадаючий список) з ID @+id/creditProgramSpinner – розміщений у верхній частині інтерфейсу, дозволяє користувачеві вибрати тип кредитної програми. Його ширина задана як match\_parent, що означає, що він займе всю доступну ширину контейнера, а висота встановлена на wrap\_content, щоб підлаштовуватися під вміст.

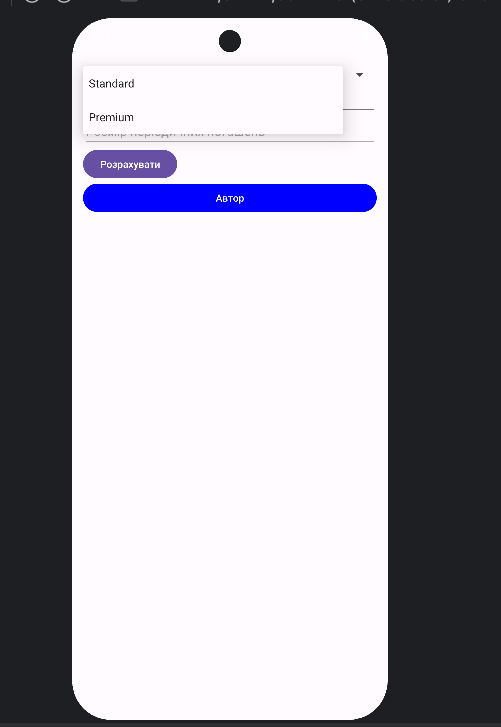
EditText для введення початкової суми кредиту з ID @+id/initialAmountEditText – текстове поле, що має підказку ("Початкова сума запозичення") та налаштоване для введення десяткових чисел (numberDecimal). Поле дозволяє користувачеві ввести початкову суму, з якої починається розрахунок кредиту.

EditText для введення розміру щомісячного платежу з ID @+id/monthlyPaymentEditText – текстове поле, яке також підтримує введення десяткових чисел і має підказку ("Розмір періодичних погашень"). Це поле призначене для введення суми щомісячного платежу, яку користувач планує сплачувати для погашення кредиту.

Button для запуску розрахунку з ID @+id/calculateButton – це кнопка з текстом "Розрахувати". Коли користувач натискає на неї, виконується логіка для обчислення платежів чи інших фінансових параметрів на основі введених даних.

#### Button для переходу до екрану автора з ID @+id/Author – ця кнопка з синім фоном (колір заданий параметром backgroundTint="#0000FF") і текстом "Автор". Вона використовує атрибут android:onClick="openAuthorActivity", що означає, що при натисканні викликається метод openAuthorActivity, який відкриває новий екран, ймовірно, з інформацією про автора програми.

# Робота програми



Так виглядає початковий екран.



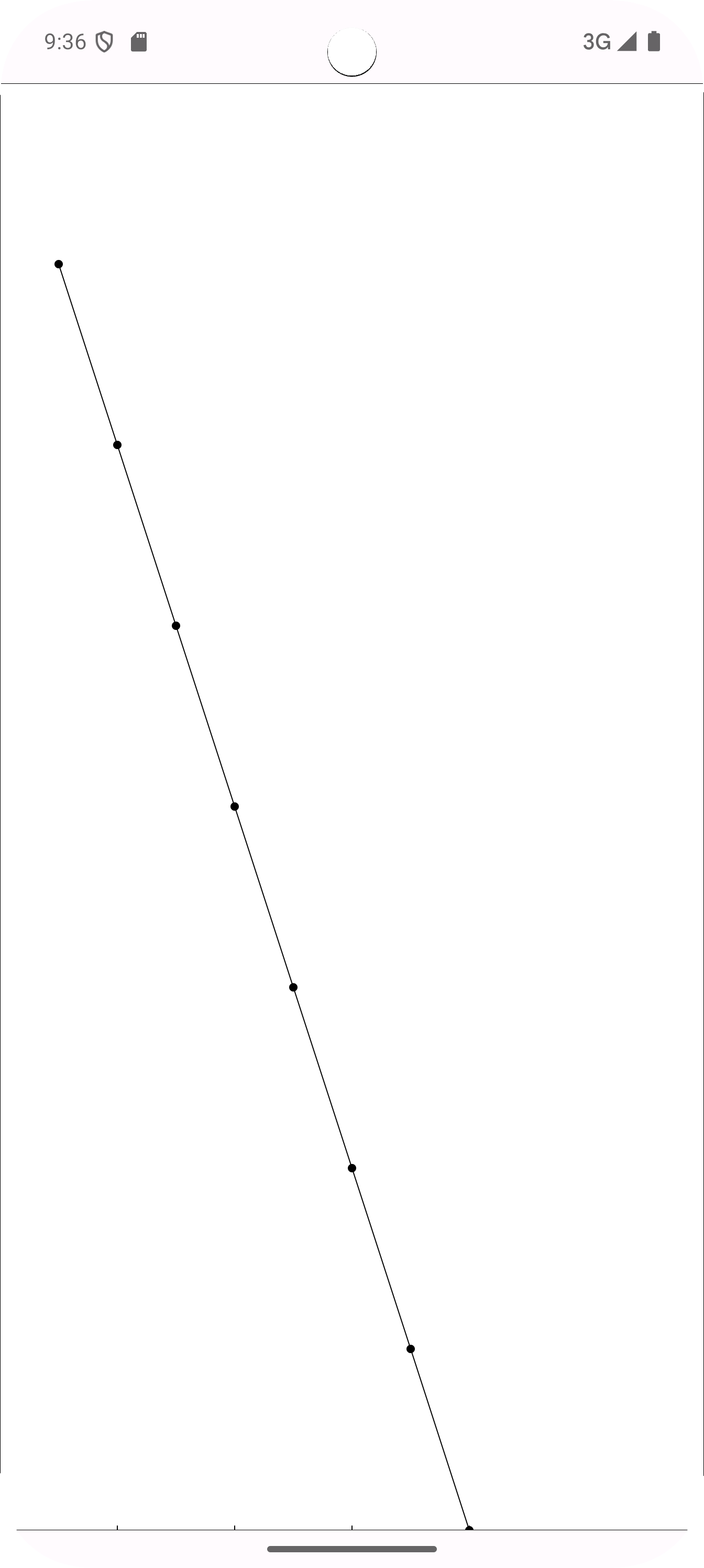
Так виглядає сторінка автора



Сторінка після розрахунку кредиткої заборгованості



При натисканні кнопки зберегти, зберігає значення до файлу.



При натисканні кнопки, відобразити, відображає графік функції з даних, що збережені у файлі.

# **Список використаних джерел**

1. Голубєва К.М., Денисов С.В., Кашпур О.Ф., Клюшин Д.А., Риженко А.І. Чисельні методи інтегрування. – Київ: Київський національний університет імені Тараса Шевченка, 2019 – 55 с.
2. Документація Android Studio

URL: <https://developer.android.com/develop>

# **Додаток A**

package com.mobileapp.mobilelaba1;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
  
public class AuthorActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_author*);  
 }  
  
 public void openMainPage(View view) {  
 Intent intent = new Intent(this, MainActivity.class);  
 startActivity(intent);  
 }  
}

# **Додаток B**

package com.mobileapp.mobilelaba1;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
public class CalculationActivity extends AppCompatActivity {  
  
 private TextView resultTextView;  
 private Button saveButton, graphButton;  
 private final String fileName = "credit\_calculations.txt";  
  
 private double initialAmount, monthlyPayment;  
 private String program;  
 private double remainingAmount;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_calculation*);  
  
 resultTextView = findViewById(R.id.*resultTextView*);  
 saveButton = findViewById(R.id.*saveButton*);  
 graphButton = findViewById(R.id.*graphButton*);  
  
 program = getIntent().getStringExtra("program");  
 initialAmount = getIntent().getDoubleExtra("initialAmount", 0);  
 monthlyPayment = getIntent().getDoubleExtra("monthlyPayment", 0);  
  
 remainingAmount = calculateRemainingAmount(initialAmount, monthlyPayment, program);  
  
 String resultText = "Програма: " + program + "\n" +  
 "Початкова сума: " + initialAmount + "\n" +  
 "Періодичний платіж: " + monthlyPayment + "\n" +  
 "Залишок заборгованості: " + remainingAmount;  
  
 resultTextView.setText(resultText);  
  
 // Збереження даних у файл  
 saveButton.setOnClickListener(v -> saveDataToFile(resultText));  
  
 // Перехід до відображення графіку  
 graphButton.setOnClickListener(v -> {  
 Intent intent = new Intent(CalculationActivity.this, GraphActivity.class);  
 intent.putExtra("initialAmount", initialAmount);  
 intent.putExtra("monthlyPayment", monthlyPayment);  
 intent.putExtra("program", program);  
 startActivity(intent);  
 });  
 }  
  
 private double calculateRemainingAmount(double initialAmount, double monthlyPayment, String program) {  
 double remainingAmount = initialAmount;  
  
 switch (program) {  
 case "Standard":  
 remainingAmount = initialAmount - (monthlyPayment \* 12);  
 break;  
 case "Premium":  
 for (int i = 0; i < 12; i++) {  
 remainingAmount -= (monthlyPayment + (monthlyPayment \* 0.05 \* i));  
 }  
 break;  
 default:  
 remainingAmount = initialAmount;  
 }  
  
 return remainingAmount;  
 }  
  
 private void saveDataToFile(String data) {  
 try {  
 FileOutputStream fos = openFileOutput(fileName, Context.*MODE\_PRIVATE*);  
 fos.write(data.getBytes());  
 fos.close();  
 Toast.*makeText*(this, "Дані збережено у файл: " + fileName, Toast.*LENGTH\_LONG*).show();  
 } catch (IOException e) {  
 Toast.*makeText*(this, "Помилка збереження даних у файл", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

# **Додаток С**

package com.mobileapp.mobilelaba1;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
  
import com.softmoore.android.graphlib.Graph;  
import com.softmoore.android.graphlib.GraphView;  
import com.softmoore.android.graphlib.Point;  
  
public class GraphActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_graph*);  
  
 // Отримання даних з Intent  
 double initialAmount = getIntent().getDoubleExtra("initialAmount", 0);  
 double monthlyPayment = getIntent().getDoubleExtra("monthlyPayment", 0);  
 String program = getIntent().getStringExtra("program");  
  
 // Підготовка точок для графіку  
 Point[] points = createGraphPoints(initialAmount, monthlyPayment, program);  
  
 // Створення графіку  
 Graph graph = new Graph.Builder()  
 .setWorldCoordinates(0, 12, 0, initialAmount) // Встановлення координат (місяці, залишок)  
 .addLineGraph(points) // Додавання лінійного графіку  
 .build();  
  
 // Підключення GraphView до макета  
 GraphView graphView = findViewById(R.id.*graphView*);  
 graphView.setGraph(graph);  
 }  
  
 // Метод для підготовки точок для графіку  
 private Point[] createGraphPoints(double initialAmount, double monthlyPayment, String program) {  
 Point[] points = new Point[12];  
 double remainingAmount = initialAmount;  
  
 switch (program) {  
 case "Standard":  
 for (int i = 0; i < 12; i++) {  
 remainingAmount -= monthlyPayment;  
 points[i] = new Point(i + 1, remainingAmount);  
 }  
 break;  
  
 case "Premium":  
 for (int i = 0; i < 12; i++) {  
 remainingAmount -= (monthlyPayment + (monthlyPayment \* 0.05 \* i));  
 points[i] = new Point(i + 1, remainingAmount);  
 }  
 break;  
  
 case "Business":  
 for (int i = 0; i < 12; i++) {  
 if (i > 2) remainingAmount -= monthlyPayment;  
 points[i] = new Point(i + 1, remainingAmount);  
 }  
 break;  
  
 case "Flexible":  
 double monthlyInterestRate = 0.02;  
 for (int i = 0; i < 12; i++) {  
 double flexiblePayment = remainingAmount \* monthlyInterestRate;  
 remainingAmount -= flexiblePayment;  
 points[i] = new Point(i + 1, remainingAmount);  
 }  
 break;  
  
 case "Annuitet":  
 double annualRate = 0.12;  
 double annuitetPayment = (initialAmount \* (annualRate / 12)) / (1 - Math.*pow*(1 + (annualRate / 12), -12));  
 for (int i = 0; i < 12; i++) {  
 remainingAmount -= annuitetPayment;  
 points[i] = new Point(i + 1, remainingAmount);  
 }  
 break;  
  
 default:  
 for (int i = 0; i < 12; i++) {  
 points[i] = new Point(i + 1, initialAmount);  
 }  
 }  
  
 return points;  
 }  
}

**Додаток D**

package com.mobileapp.mobilelaba1;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Spinner;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Spinner creditProgramSpinner;  
 private EditText initialAmountEditText, monthlyPaymentEditText;  
 private Button calculateButton;  
  
 private String selectedProgram;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 creditProgramSpinner = findViewById(R.id.*creditProgramSpinner*);  
 initialAmountEditText = findViewById(R.id.*initialAmountEditText*);  
 monthlyPaymentEditText = findViewById(R.id.*monthlyPaymentEditText*);  
 calculateButton = findViewById(R.id.*calculateButton*);  
  
 // Заповнення Spinner з варіантами програм  
 ArrayAdapter<CharSequence> adapter = ArrayAdapter.*createFromResource*(this,  
 R.array.*credit\_programs*, android.R.layout.*simple\_spinner\_item*);  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 creditProgramSpinner.setAdapter(adapter);  
  
 creditProgramSpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
 selectedProgram = parent.getItemAtPosition(position).toString();  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
 selectedProgram = "Standard";  
 }  
 });  
  
 calculateButton.setOnClickListener(v -> {  
 String initialAmount = initialAmountEditText.getText().toString();  
 String monthlyPayment = monthlyPaymentEditText.getText().toString();  
  
 if (initialAmount.isEmpty() || monthlyPayment.isEmpty()) {  
 Toast.*makeText*(MainActivity.this, "Будь ласка, введіть всі дані", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 Intent intent = new Intent(MainActivity.this, CalculationActivity.class);  
 intent.putExtra("program", selectedProgram);  
 intent.putExtra("initialAmount", Double.*parseDouble*(initialAmount));  
 intent.putExtra("monthlyPayment", Double.*parseDouble*(monthlyPayment));  
 startActivity(intent);  
 });  
 }  
 public void openAuthorActivity(View view) {  
 Intent intent = new Intent(this, AuthorActivity.class);  
 startActivity(intent);  
 }  
}