МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО –

КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП`ЮТЕРНИХ НАУК

**Курсова робота**

з дисципліни "Прикладне програмування Java" на тему: розробка програмного додатку з розв'язку квадратного рівняння

Роботу виконав здобувач вищої освіти,

студент групи ІСД – 12

Котасов О.С

Роботу перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата перевірки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ, 2023р.

**Зміст**

|  |  |
| --- | --- |
| **РОЗДІЛ 1**. ОПИС ЗАВДАННЯ ТА МАТЕМАТИЧНА МОДЕЛЬ   * 1. РОЗРАХУНОК КОРЕНІВ КВАДРАТНОГО РІВНЯННЯ.   2. Математична модель   3. РОЗРАХУНОК ГЕОМЕТРИЧНОЇ ПРОГРЕСІЇ   4. Математична модель   ……………………………………………………………………………….. | 3  4  5 |
| РОЗДІЛ 2. ПРОЦЕС РОЗРОБКИ ДОДАТКІВ | 6 |
| ………………………………………………………………………………... |
| РОЗДІЛ 3. ВІДОБРАЖЕННЯ ФУНКЦІОНАЛУ ДОДАТКУ  …………………………………………………………………………........... | 7 |
| РОЗДІЛ 4. РОЗРОБКА TIC-TAC-TOE  ………………………………………………………………………………... | 8 |
| ВИСНОВОК  …………………………………………………………………………........... | 9 |
| ЛІТЕРАТУРА  …………………………………………………………………………........... | 10 |
| ЛІСТИНГ №1. КВАДРАТНЕ РІВНЯННЯ | 11 |
| ЛІСТИНГ №2. ГЕОМЕТРИЧНА ПРОГРЕСІЯ  ЛІСТИНГ №3 Tic\_Tac\_Toe | 14  17 |
|  |
|  |
|  |

**РОЗДІЛ 1**

ОПИС ЗАВДАННЯ ТА МАТЕМАТИЧНА МОДЕЛЬ

* 1. РОЗРАХУНОК КОРЕНІВ КВАДРАТНОГО РІВНЯННЯ.

Потрібно розробити додаток на Java Swing, який дозволяє розв'язувати квадратні рівняння. За допомогою Swing, додаток повинен мати графічний інтерфейс та реалізовані основні функції, такі як обчислення дискримінанта, знаходження коренів, включаючи комплексні корені, та відображення результатів.

Графічний інтерфейс повинен містити поля для введення коефіцієнтів a, b та c, кнопку для обчислення результатів і поле для відображення результатів. Крім того, має бути ще дві кнопки: одна для очищення всіх полів і друга для виходу з додатку. Після натискання на кнопку Обчислити, додаток має розраховувати дискримінант і залежно від його значення, вивести корені рівняння.

* 1. **Математична модель**

Для розвязання квадратного рівняння потрібно використовувати один з методів(через дискріминант або теорема Вієта), Мій додаток знаходить корені через дискримінант.



Рисунок 1. Знаходження Дискримінанта

Потім, додаток перевіряє значення дискримінанта , та в залежності від значення відбувається розгалуження:

* Якщо дискримінант більше за нуль, то рівняння має два дійсних розв'язка.
* Якщо дискримінант дорівнює нулю, то рівняння має один дійсний розв'язок.
* Якщо воно менше за нуль, то рівняння має комплексні розв'язки. Додаток перетворює дискримінант в додатнє число і додає комплексні розв'язки)

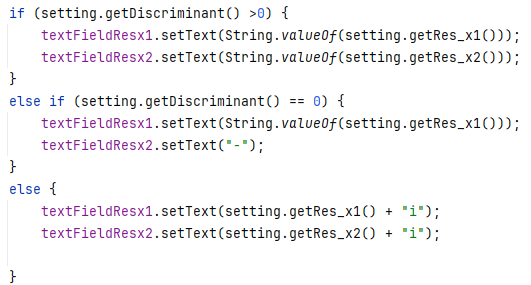
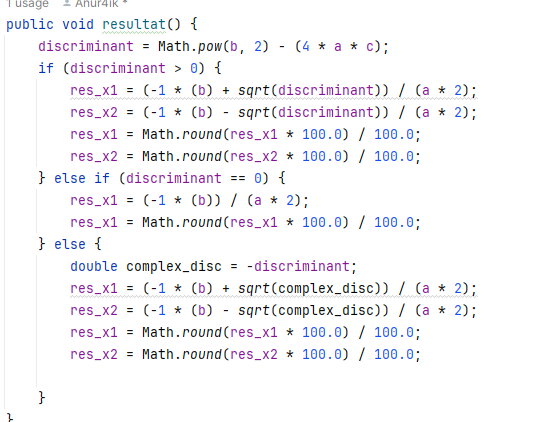


Рисунок 2. Знаходження коренів та вивід їх

* 1. **РОЗРАХУНОК ГЕОМЕТРИЧНОЇ ПРОГРЕСІЇ**

Додаток на Java Swing, який дозволяє знаходити геометричну прогресію.

Графічний інтерфейс повинен містити поля для введення першого числа прогресії (b₁), знаменника геометричної прогресії (q), кількості членів прогресії (bₙ) ,кнопку для обчислення результатів та поле для відображення результатів. Крім того, має бути ще дві кнопки: одна для очищення всіх полів і друга для виходу з додатку. Після натискання на кнопку 'Обчислити', додаток повинен знайти та вивести в таблицю значення кожної геометричної прогресії, які були задані.

* 1. **Математична модель**

Додаток приймає 3 параметри це перше число прогресії(b₁),знаменник прогресії(q),та кількість членів прогресії. Потім за допомогою формули знаходить кожний член геометричної прогресії bn= ⋅

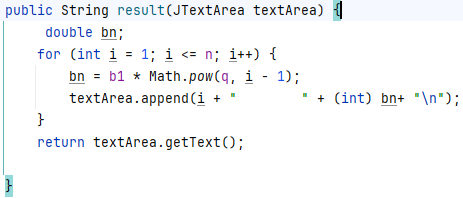


Рисунок 3. Обчислення геометричної прогресії

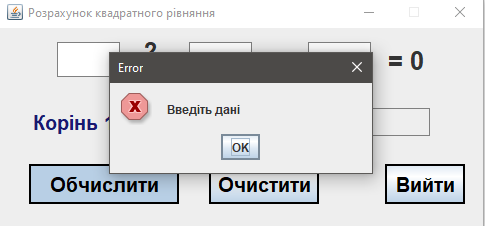
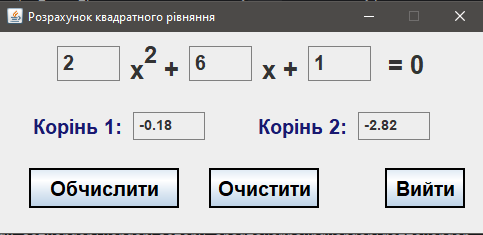
**РОЗДІЛ 2**

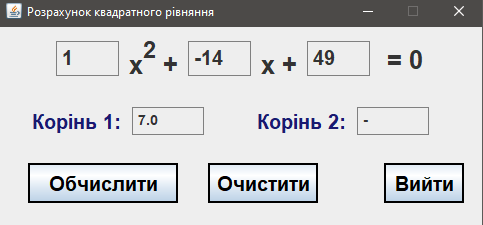
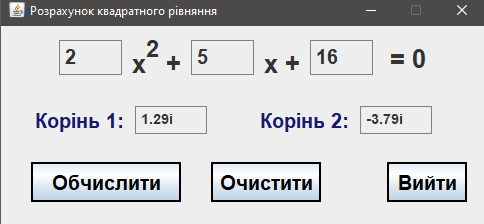
ПРОЦЕС РОЗРОБКИ ДОДАТКІВ

Спочатку я почав з розробки додатка який буде знаходити корені квадратного рівняння. Для реалізації графічного інтерфейсу мені потрібно було використати бібліотеку Swing. Після підключення Swing до проекту, я почав розробку дизайну інтерфейсу. В процесі розробки був реалізований зрозумілий інтерфейс для введення коефіцієнтів квадратного рівняння та відображення результатів обчислень. Використовувалися базові компоненти Swing, такі як JLabel, JTextField та JButton. Я швидко пройшов цей етап і почав думати над логікою програми. Для початку мені треба було зробити перевірку введених користувачем даних. Це дозволяє уникнути помилок, пов'язаних з неправильним введенням коефіцієнтів. Мені потрібно було реалізувати перевірку дискримінанта.Тому програма обробляє три можливі випадки значення дискримінанта: позитивний, нульовий та від'ємний. В залежності від значення дискримінанта, програма відповідно обчислює та відображає дійсні або комплексні корені рівняння.

Після завершення розробки додатка для знаходження коренів квадратного рівняння, я перейшов до розробки додатка, який обчислює елементи геометричної прогресії. Для реалізації графічного інтерфейсу також використовується бібліотека Swing.. Для створення інтерфейсу я використав компоненти Swing, такі як JLabel, JTextField ,JButton та JTextArea. Інтерфейс дозволяє користувачеві вводити початковий член прогресії (b1), знаменник прогресії (q) та кількість елементів (n). Відображення результатів обчислень здійснюється в окремому полі. Після розробки інтерфейсу я перейшов до реалізації логіки обчислення елементів геометричної прогресії. Потрібно реалізувати перевірку, чи введені значення коректні .Використовуючи формулу геометричної прогресії, обчислити всі елементи до заданої кількості. bn= ⋅

**РОЗДІЛ 3**

ВІДОБРАЖЕННЯ ФУНКЦІОНАЛУ ДОДАТКУ

 Рисунок 4. Перевірка полів Рисунок 5. Вивід коренів при D>0

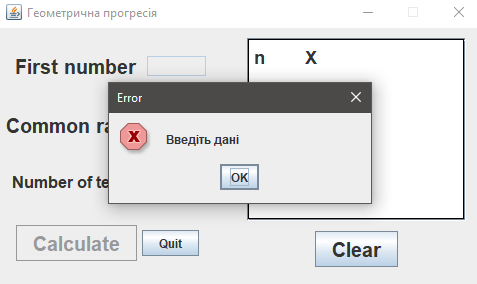
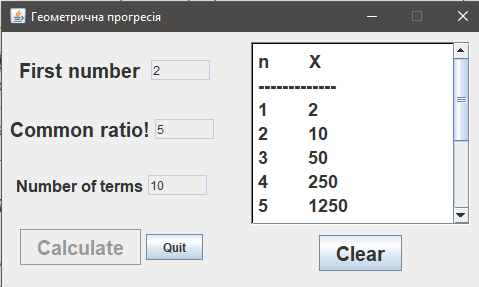
Рисунок 6. Вивід коренів при D=0 Рисунок 7. Вивід коренів при D<0

Рисунок 8. Перевірка полів Рисунок 9. Вивід прогресії

**РОЗДІЛ 4**

**Розробка Tic-Tac-Toe**

Для засвоєння знань з Java та бібліотекою Swing я вирішив розробити гру "Хрестики-Нулики". Я використав JPanel та застосував GridLayout, що дозволило мені реалізувати поле 3 на 3. Далі я розташував кнопки на всі клітинки поля. Я додав логіку, щоб при натисканні на кнопку відображався хрестик або нулик. Використавши знання про матриці в Java, я реалізував їх так, щоб при натисканні на кнопку можна було змінювати значення в матриці. Основною задачею залишалась перевірка рядків на наявність виграшної комбінації. Завдяки кільком сайтам і GPT я зміг реалізувати ефективну перевірку. Після якої відбувається оголошення переможця та перезавантаження гри.

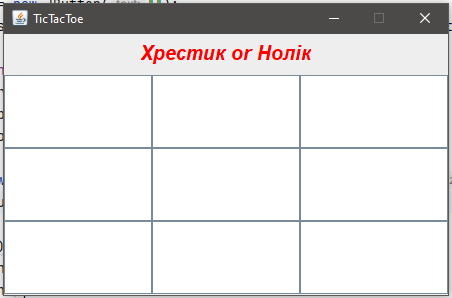
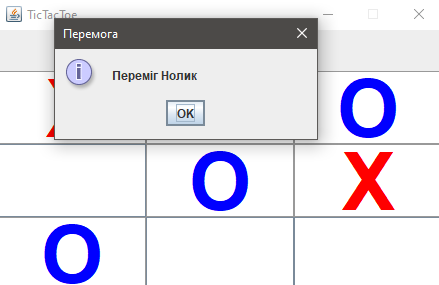


Рисунок 10. Вигляд Tic-Tac-Toe Рисунок 11. Переміг Нолик

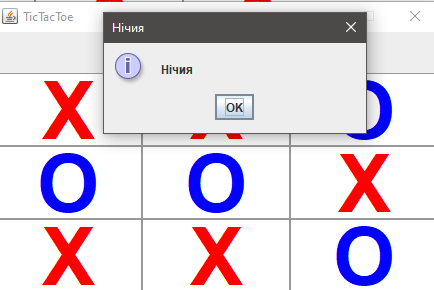
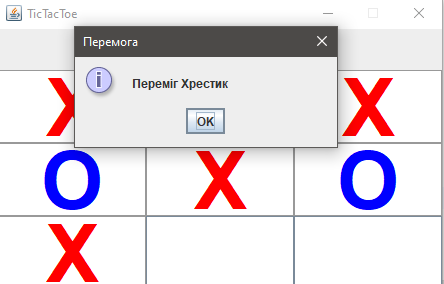


Рисунок 11. Переміг Хрестик Рисунок 11. Нічия

**Висновок**

Результатом виконання роботи є повнофункціональний програмні додатки, які можуть бути використані для навчальних цілей, а також в якості інструменту для розв'язання практичних задач з алгебри. Робота над проектом дозволила закріпити теоретичні знання з програмування на мові Java, розвинути навички проектування програмного забезпечення та роботи з графічними інтерфейсами Swing.

**Використані Джерела**

1 Introduction to Java Programming, Comprehensive Version Edition: 9th Edition

Hardcover: 1190 pages;

2. GPT-4o;

3. ВІКІПІДРУЧНИК:

https://uk.wikibooks.org/wiki/Освоюємо\_Java/Графічні\_компоненти\_Swing;

4. Java Swing Tutorial https://www.javatpoint.com/java-swing;

5. Java 8 API https://docs.oracle.com/javase/8/docs/api;

6.Java Rush,Матриці https://javarush.com/ua/quests/lectures/ua.questsyntaxpro.level05.lecture04

**Лістинг №1(Квадратне рівняння)**

**SwingUI**

1 package quadratic;

3 import javax.swing.\*;

4 import javax.swing.border.Border;

5 import java.awt.\*;

6 import java.awt.event.ActionEvent;

7 import java.awt.event.ActionListener;

9 public class SwingUI extends JFrame {

10 public void MyFrame() {

11 JPanel panel1 = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));

12 JPanel panel2 = new JPanel(new FlowLayout(FlowLayout.LEFT, 5, 20));

13 JPanel panel3 = new JPanel(new FlowLayout(FlowLayout.LEFT, 30, 5));

14 JButton button1 = new JButton("Обчислити");

15 JButton button2 = new JButton("Очистити");

16 JButton button3 = new JButton("Вийти");

17 JTextField textFieldx2 = new JTextField(3);

18 JTextField textFieldx = new JTextField(3);

19 JTextField textField = new JTextField(3);

20 JTextField textFieldResx1 = new JTextField(5);

21 JTextField textFieldResx2 = new JTextField(5);

22 textFieldResx1.setEditable(false);

23 textFieldResx2.setEditable(false);

24 JLabel labelx2 = new JLabel("<html>x<sup>2</sup> + </html>");

25 JLabel labelx = new JLabel("x +");

26 JLabel label = new JLabel(" = 0");

27 JLabel labelResultx1 = new JLabel("Корінь 1: ");

28 JLabel labelResultx2 = new JLabel(" Корінь 2: ");

29 JLabel margin\_left\_panel2 = new JLabel(" ");

30 JLabel margin\_left\_panel3 = new JLabel(" ");

31 labelx2.setBorder(BorderFactory.createEmptyBorder(-5, 0, 0, 0));

32 labelx.setBorder(BorderFactory.createEmptyBorder(10, 0, 0, 0));

33 Border matteBorder = BorderFactory.createMatteBorder(1, 1, 1, 1, Color.gray);

34 Border emptyBorder = BorderFactory.createEmptyBorder(2, 5, 5, 5);

35 textFieldx2.setBorder(BorderFactory.createCompoundBorder(matteBorder, emptyBorder));

36 textFieldx.setBorder(BorderFactory.createCompoundBorder(matteBorder, emptyBorder));

37 textField.setBorder(BorderFactory.createCompoundBorder(matteBorder, emptyBorder));

38 textFieldResx1.setBorder(BorderFactory.createCompoundBorder(matteBorder, emptyBorder));

39 textFieldResx2.setBorder(BorderFactory.createCompoundBorder(matteBorder, emptyBorder));

40 labelx2.setFont(new Font("Comic Sans", Font.BOLD, 26));

41 labelx.setFont(new Font("Comic Sans", Font.BOLD, 26));

42 label.setFont(new Font("Comic Sans", Font.BOLD, 26));

43 panel1.add(textFieldx2);

44 panel1.add(labelx2);

45 panel1.add(textFieldx);

46 panel1.add(labelx);

47 panel1.add(textField);

48 panel1.add(label);

49 labelResultx1.setFont(new Font("Comic Sans", Font.BOLD, 20));

50 labelResultx1.setForeground(Color.decode("#191970"));

51 labelResultx2.setFont(new Font("Comic Sans", Font.BOLD, 20));

52 labelResultx2.setForeground(Color.decode("#191970"));

53 textFieldx2.setFont(new Font("Comic Sans", Font.BOLD, 20));

54 textFieldx.setFont(new Font("Comic Sans", Font.BOLD, 20));

55 textField.setFont(new Font("Comic Sans", Font.BOLD, 20));

56 textFieldResx1.setFont(new Font("Comic Sans", Font.BOLD, 14));

57 textFieldResx2.setFont(new Font("Comic Sans", Font.BOLD, 14));

58 panel2.add(margin\_left\_panel2);

59 panel2.add(labelResultx1);

60 panel2.add(textFieldResx1);

61 panel2.add(labelResultx2);

62 panel2.add(textFieldResx2);

63 button1.setPreferredSize(new Dimension(150, 40));

64 button1.setFont(new Font("Comic Sans", Font.BOLD, 20));

65 button1.setFocusable(false);

66 button1.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2));

67 button1.setForeground(Color.BLACK);

68 button2.setPreferredSize(new Dimension(110, 40));

69 button2.setFont(new Font("Comic Sans", Font.BOLD, 20));

70 button2.setFocusable(false);

71 button2.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2));

72 button2.setForeground(Color.BLACK);

73 button3.setPreferredSize(new Dimension(80, 40));

74 button3.setFont(new Font("Comic Sans", Font.BOLD, 20));

75 button3.setFocusable(false);

76 button3.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2));

77 button3.setForeground(Color.BLACK);

78 panel3.add(button1);

79 panel3.add(button2);

80 panel3.add(margin\_left\_panel3);

81 panel3.add(button3);

82 button1.addActionListener(new ActionListener() {

83 public void actionPerformed(ActionEvent e) {

84 try {

85 double a = Double.parseDouble(textFieldx2.getText());

86 double b = Double.parseDouble(textFieldx.getText());

87 double c = Double.parseDouble(textField.getText());

88 textFieldx2.setEditable(false);

89 textFieldx.setEditable(false);

90 textField.setEditable(false);

91 OperationController setting = new OperationController(a, b, c);

92 setting.resultat();

93 if (setting.getDiscriminant() > 0) {

94 textFieldResx1.setText(String.valueOf(setting.getRes\_x1()));

95 textFieldResx2.setText(String.valueOf(setting.getRes\_x2()));

96 } else if (setting.getDiscriminant() == 0) {

97 textFieldResx1.setText(String.valueOf(setting.getRes\_x1()));

98 textFieldResx2.setText("-");

99 } else {

100 textFieldResx1.setText(setting.getRes\_x1() + "i");

101 textFieldResx2.setText(setting.getRes\_x2() + "i");

102 }

103 } catch (NumberFormatException ex) {

104 textFieldx2.setText("");

105 textFieldx.setText("");

106 textField.setText("");

107 JOptionPane.showMessageDialog(SwingUI.this, "Введіть дані", "Error", JOptionPane.ERROR\_MESSAGE);

108 }

109 }

110 });

111 button2.addActionListener(new ActionListener() {

112 public void actionPerformed(ActionEvent e) {

113 textFieldx2.setText("");

114 textFieldx.setText("");

115 textField.setText("");

116 textFieldResx1.setText("");

117 textFieldResx2.setText("");

118 textFieldx2.setEditable(true);

119 textFieldx.setEditable(true);

120 textField.setEditable(true);

121 }

122 });

123 button3.addActionListener(new ActionListener() {

124 public void actionPerformed(ActionEvent e) {

125 System.exit(0);

126 }

127 });

128 panel1.setPreferredSize(new Dimension(250, 60));

129 panel2.setPreferredSize(new Dimension(250, 60));

130 panel3.setPreferredSize(new Dimension(250, 70));

131 this.setTitle("Розрахунок квадратного рівняння");

132 this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

133 this.setSize(500, 240);

134 this.setResizable(false);

135 this.setLocationRelativeTo(null);

136 this.add(panel1, BorderLayout.NORTH);

137 this.add(panel2, BorderLayout.CENTER);

138 this.add(panel3, BorderLayout.SOUTH);

139 this.setVisible(true);

140 }

141}

**OperationController**

1 package quadratic;

2 import static java.lang.Math.sqrt;

3 public class OperationController {

4 private double a, b, c;

5 private double res\_x1, res\_x2;

6 private double discriminant;

7 public double getDiscriminant() {

8 return discriminant;

9 }

10 public OperationController(double a, double b, double c) {

11 this.a = a;

12 this.b = b;

13 this.c = c;

14 }

15 public double getRes\_x1() {

16 return res\_x1;

17 }

18 public double getRes\_x2() {

19 return res\_x2;

20 }

21 public void resultat() {

22 discriminant = Math.pow(b, 2) - (4 \* a \* c);

23 if (discriminant > 0) {

24 res\_x1 = (-1 \* (b) + sqrt(discriminant)) / (a \* 2);

25 res\_x2 = (-1 \* (b) - sqrt(discriminant)) / (a \* 2);

26 res\_x1 = Math.round(res\_x1 \* 100.0) / 100.0;

27 res\_x2 = Math.round(res\_x2 \* 100.0) / 100.0;

28 } else if (discriminant == 0) {

29 res\_x1 = (-1 \* (b)) / (a \* 2);

30 res\_x1 = Math.round(res\_x1 \* 100.0) / 100.0;

31 } else {

32 double complex\_disc = -discriminant;

33 res\_x1 = (-1 \* (b) + sqrt(complex\_disc)) / (a \* 2);

34 res\_x2 = (-1 \* (b) - sqrt(complex\_disc)) / (a \* 2);

35 res\_x1 = Math.round(res\_x1 \* 100.0) / 100.0;

36 res\_x2 = Math.round(res\_x2 \* 100.0) / 100.0;

37 }

38 }

39}

**MainApp**

1package quadratic;

2public class MainApp {

3 public static void main(String[] args) {

4 new SwingUI().MyFrame();

5 }

6}

**Лістинг №2(Геометрична прогресія)**

**SwingUI**

1 package degreesOfNumber;

2 import javax.swing.\*;

3 import javax.swing.border.Border;

4 import javax.swing.border.EmptyBorder;

5 import javax.swing.table.DefaultTableModel;

6 import java.awt.\*;

7 import java.awt.event.ActionEvent;

8 import java.awt.event.ActionListener;

9 public class SwingUI extends JFrame {

10 public void MyFrame(){

11 JPanel panel1 = new JPanel();

12 JPanel panel2 = new JPanel(new FlowLayout(FlowLayout.CENTER, 0, 10));

13 panel1.setLayout(new BoxLayout(panel1, BoxLayout.Y\_AXIS));

14 panel1.add(Box.createVerticalStrut(20));

15 JPanel in\_panel1 = new JPanel();

16 JPanel in\_panel2 = new JPanel();

17 JPanel in\_panel3 = new JPanel();

18 JPanel in\_panel4 = new JPanel();

19 JLabel l\_firstNumber= new JLabel(" First number ");

20 JLabel l\_qNumber = new JLabel("Common ratio!");

21 JLabel l\_kNumber = new JLabel("Number of terms");

22 JButton button1 = new JButton("Calculate");

23 JTextField t\_firstNumber = new JTextField(5);

24 JTextField t\_qNumber = new JTextField(5);

25 JTextField t\_kNumber = new JTextField(5);

26 JTextArea textArea = new JTextArea(7, 12);

27 JScrollPane scrollPane = new JScrollPane(textArea);

28 JButton button2 = new JButton("Quit");

29 JButton button3 = new JButton("Clear");

30 button1.setFont(new Font("Comic Sans", Font.BOLD, 20));

31 button3.setFont(new Font("Comic Sans", Font.BOLD, 20));

32 button1.setFocusable(false);

33 button2.setFocusable(false);

34 button3.setFocusable(false);

35 l\_firstNumber.setFont(new Font("Comic Sans",Font.BOLD,20));

36 l\_qNumber.setFont(new Font("Comic Sans",Font.BOLD,20));

37 l\_kNumber.setFont(new Font("Comic Sans",Font.BOLD,16));

38 textArea.setEditable(false);

39 Border matteBorder = BorderFactory.createMatteBorder(1, 1, 1, 1, Color.black);

40 Border emptyBorder = BorderFactory.createEmptyBorder(5, 5, 5, 5);

41 textArea.setBorder(BorderFactory.createCompoundBorder(matteBorder, emptyBorder));

42 textArea.setText("n X"+"\n");

43 textArea.append("-------------"+"\n");

44 scrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL\_SCROLLBAR\_NEVER);

45 textArea.setFont(new Font("Comic Sans",Font.BOLD,18));

46 in\_panel1.add(l\_firstNumber);

47 in\_panel1.add(t\_firstNumber);

48 panel1.add(in\_panel1);

49 in\_panel2.add(l\_qNumber);

50 in\_panel2.add(t\_qNumber);

51 panel1.add(in\_panel2);

52 in\_panel3.add(l\_kNumber);

53 in\_panel3.add(t\_kNumber);

54 panel1.add(in\_panel3);

55 in\_panel4.add(button1);

56 in\_panel4.add(button2);

57 panel1.add(in\_panel4);

58 panel2.add(scrollPane );

59 panel2.add(button3);

60 button1.addActionListener(new ActionListener() {

61 public void actionPerformed(ActionEvent e) {

62 t\_firstNumber.setEditable(false);

63 t\_qNumber.setEditable(false);

64 t\_kNumber.setEditable(false);

65 button1.setEnabled(false);

66 try {

67 int b = Integer.parseInt(t\_firstNumber.getText());

68 int q = Integer.parseInt(t\_qNumber.getText());

69 int n = Integer.parseInt(t\_kNumber.getText());

70 OperationController setting = new OperationController(b, q, n);

71 textArea.setText(String.valueOf(setting.result(textArea)));

72 }

73 catch (NumberFormatException ex){

74 JOptionPane.showMessageDialog(SwingUI.this, "Введіть дані", "Error", JOptionPane.ERROR\_MESSAGE);

75 }

76 }

77 });

78 button2.addActionListener(new ActionListener() {

79 public void actionPerformed(ActionEvent e) {

80

81 System.exit(0);

82 }

83 });

84 button3.addActionListener(new ActionListener() {

85 public void actionPerformed(ActionEvent e) {

86 t\_firstNumber.setEditable(true);

87 t\_qNumber.setEditable(true);

88 button1.setEnabled(true);

89 t\_kNumber.setEditable(true);

90 t\_firstNumber.setText("");

91 t\_qNumber.setText("");

92 t\_kNumber.setText("");

93 textArea.setText("n X"+"\n");

94 textArea.append("-------------"+"\n");

95 }

96 });

97 panel1.setPreferredSize(new Dimension(220, 60));

98 panel2.setPreferredSize(new Dimension(250, 60));

99 this.setTitle("Геометрична прогресія");

100 this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

101 this.setSize(500, 300);

102 this.setResizable(false);

103 this.setLocationRelativeTo(null);

104 this.add(panel1, BorderLayout.WEST);

105 this.add(panel2, BorderLayout.EAST);

106 this.setVisible(true);

107 }

108}

**OperationController**

1 package degreesOfNumber;

2 import javax.swing.\*;

3 public class OperationController {

4 private int b1, q, n;

5 public OperationController(int b1, int q, int n) {

6 this.b1 = b1;

7 this.q = q;

8 this.n = n;

9 }

10 public String result(JTextArea textArea) {

11 double bn;

12 for (int i = 1; i <= n; i++) {

13 bn = b1 \* Math.pow(q, i - 1);

14 textArea.append(i + " " + (int) bn+ "\n");

15 }

16 return textArea.getText();

17 }

18}

**MainApp**

1package degreesOfNumber;

2public class MainApp {

3 public static void main(String[] args) {

4 new SwingUI().MyFrame();

5 }

6}

**Лістинг №3(Tic\_Tac\_Toe)**

**SwingUI**

1 package tic\_tac;

2 import javax.swing.\*;

3 import java.awt.\*;

4 import java.awt.event.ActionEvent;

5 import java.awt.event.ActionListener;

6 public class SwingUI extends JFrame {

7 public int count = 0;

8 public void MyFrame() {

9 JPanel panel1 = new JPanel();

10 JPanel panel2 = new JPanel(new GridLayout(3, 3));

11 int[][] matrix = {

12 {2, 2, 2},

13 {2, 2, 2},

14 {2, 2, 2}

15 };

16 JLabel label = new JLabel("Хрестик or Нолік");

17 JButton button1 = new JButton("");

18 JButton button2 = new JButton("");

19 JButton button3 = new JButton("");

20 JButton button4 = new JButton("");

21 JButton button5 = new JButton("");

22 JButton button6 = new JButton("");

23 JButton button7 = new JButton("");

24 JButton button8 = new JButton("");

25 JButton button9 = new JButton("");

26 JButton[] buttons = {button1, button2, button3, button4, button5, button6, button7, button8, button9};

27 for (int i = 0; i < buttons.length; i++) {

28 buttons[i].setPreferredSize(new Dimension(150, 40));

29 buttons[i].setFocusable(false);

30 buttons[i].setBackground(Color.white);

31 }

32 label.setFont(new Font("Lobster", Font.BOLD + Font.ITALIC, 20));

33 label.setForeground(Color.red);

34 panel1.add(label);

35 panel2.add(button1);

36 panel2.add(button2);

37 panel2.add(button3);

38 panel2.add(button4);

39 panel2.add(button5);

40 panel2.add(button6);

41 panel2.add(button7);

42 panel2.add(button8);

43 panel2.add(button9);

44 button1.addActionListener(new ActionListener() {

45 public void actionPerformed(ActionEvent e) {

46 matrix[0][0] = tic\_or\_toe(button1);

47 check(matrix, buttons);

48 yes\_draw(matrix, buttons);

49

50 }

51 });

52 button2.addActionListener(new ActionListener() {

53 public void actionPerformed(ActionEvent e) {

54 matrix[0][1] = tic\_or\_toe(button2);

55 check(matrix, buttons);

56 yes\_draw(matrix, buttons);

57 }

58 });

59 button3.addActionListener(new ActionListener() {

60 public void actionPerformed(ActionEvent e) {

61 matrix[0][2] = tic\_or\_toe(button3);

62 check(matrix, buttons);

63 yes\_draw(matrix, buttons);

64 }

65 });

66 button4.addActionListener(new ActionListener() {

67 public void actionPerformed(ActionEvent e) {

68 matrix[1][0] = tic\_or\_toe(button4);

69 check(matrix, buttons);

70 yes\_draw(matrix, buttons);

71 }

72 });

73 button5.addActionListener(new ActionListener() {

74 public void actionPerformed(ActionEvent e) {

75 matrix[1][1] = tic\_or\_toe(button5);

76 check(matrix, buttons);

77 yes\_draw(matrix, buttons);

78 }

79 });

80 button6.addActionListener(new ActionListener() {

81 public void actionPerformed(ActionEvent e) {

82 matrix[1][2] = tic\_or\_toe(button6);

83 check(matrix, buttons);

84 yes\_draw(matrix, buttons);

85 }

86 });

87 button7.addActionListener(new ActionListener() {

88 public void actionPerformed(ActionEvent e) {

89 matrix[2][0] = tic\_or\_toe(button7);

90 check(matrix, buttons);

91 yes\_draw(matrix, buttons);

92 }

93 });

94 button8.addActionListener(new ActionListener() {

95 public void actionPerformed(ActionEvent e) {

96 matrix[2][1] = tic\_or\_toe(button8);

97 check(matrix, buttons);

98 yes\_draw(matrix, buttons);

99 }

100 });

101 button9.addActionListener(new ActionListener() {

102 public void actionPerformed(ActionEvent e) {

103 matrix[2][2] = tic\_or\_toe(button9);

104 check(matrix, buttons);

105 yes\_draw(matrix, buttons);

106 }

107 });

108 panel2.setPreferredSize(new Dimension(50, 220));

109 this.setTitle("TicTacToe");

110 this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

111 this.setSize(460, 300);

112 this.setResizable(false);

113 this.setLocationRelativeTo(null);

114 this.add(panel1, BorderLayout.NORTH);

115 this.add(panel2, BorderLayout.SOUTH);

116 this.setVisible(true);

117 }

118 public int tic\_or\_toe(JButton button) {

119 if (count % 2 == 0) {

120 button.setText("X");

121 button.setFont(new Font("Comic Sans", Font.BOLD, 80));

122 button.setEnabled(false); // Блокируем кнопку

123 UIManager.getDefaults().put("Button.disabledText", Color.RED);

124 count++;

125 return 1;

126 } else {

127 button.setText("O");

128 button.setFont(new Font("Comic Sans", Font.BOLD, 80));

129 button.setForeground(Color.blue);

130 button.setEnabled(false);

131 UIManager.getDefaults().put("Button.disabledText", Color.BLUE);

132 count++;

133 return 0;

134 }

135 }

136 void yes\_draw(int[][] matrix, JButton[] buttons) {

137 int max = 0;

138 for (int i = 0; i < 3; i++) {

139 for (int j = 0; j < 3; j++) {

140 if (matrix[i][j] != 2) {

141 max++;

142 }

143 }

144 }

145 if (max == 9) {

146 JOptionPane.showMessageDialog(null, "Нічия", "Нічия", JOptionPane.INFORMATION\_MESSAGE);

147 restart\_game(buttons, matrix);

148 }

149 }

150 public void check(int[][] matrix, JButton[] buttons) {

151 if (count % 2 == 0) {

152 if (checkWinner(matrix, 0) == true) {

153 JOptionPane.showMessageDialog(SwingUI.this, "Переміг Нолик", "Перемога", JOptionPane.INFORMATION\_MESSAGE);

154 restart\_game(buttons, matrix);

155 }

156 } else {

157 if (checkWinner(matrix, 1) == true) {

158 JOptionPane.showMessageDialog(SwingUI.this, "Переміг Хрестик", "Перемога", JOptionPane.INFORMATION\_MESSAGE);

159 restart\_game(buttons, matrix);

160 }

161 }

162 }

163 public boolean checkWinner(int[][] matrix, int player) {

164 for (int i = 0; i < 3; i++) {

165 if (matrix[i][0] == player && matrix[i][1] == player && matrix[i][2] == player) {

166 return true;

167 }

168 }

169 for (int i = 0; i < 3; i++) {

170 if (matrix[0][i] == player && matrix[1][i] == player && matrix[2][i] == player) {

171 return true;

172 }

173 }

174 if (matrix[0][0] == player && matrix[1][1] == player && matrix[2][2] == player) {

175 return true;

176 }

177 if (matrix[0][2] == player && matrix[1][1] == player && matrix[2][0] == player) {

178 return true;

179 }

180 return false;

181 }

182 public void restart\_game(JButton[] buttons, int[][] matrix) {

183 for (int i = 0; i < buttons.length; i++) {

184 buttons[i].setText("");

185 buttons[i].setEnabled(true);

186 }

187 for (int i = 0; i < matrix.length; i++) {

188 for (int j = 0; j < matrix[i].length; j++) {

189 matrix[i][j] = 2;

190 }

191 }

192 count = 0;

193 }

194}

**Tic\_Tac\_Toe**

1package tic\_tac;

2public class Tic\_Tac\_Toe{

3 public static void main(String[] args) {

4 new SwingUI().MyFrame();

5 }

6}