Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №6\_1**

**Дисциплина: Платформо-независимое программирование**

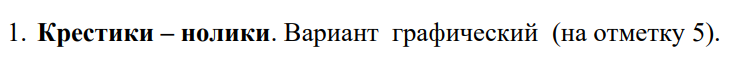
Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.С. Агаджанян

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

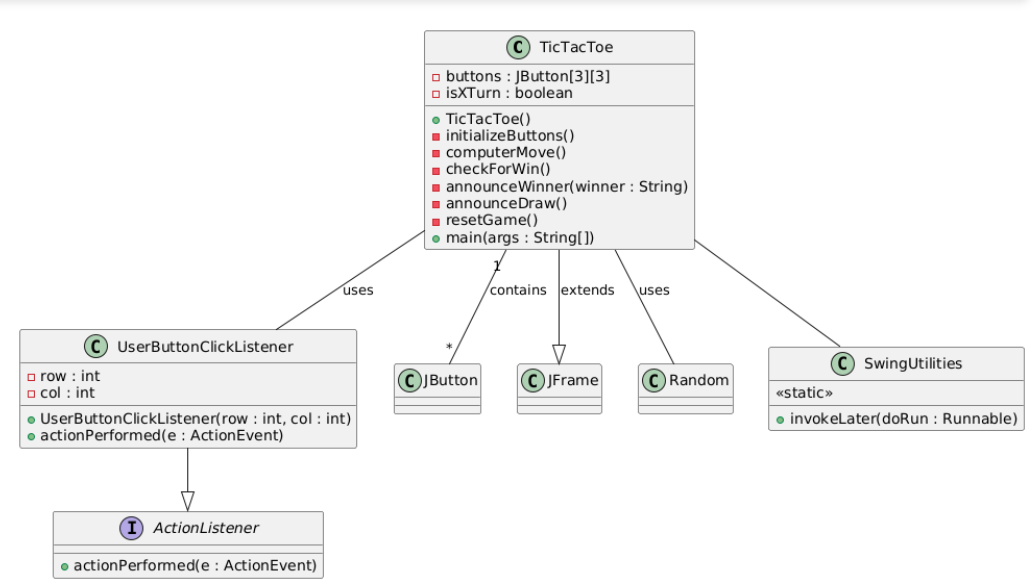
Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В. И. Шиян

**Тема**: Многопоточность

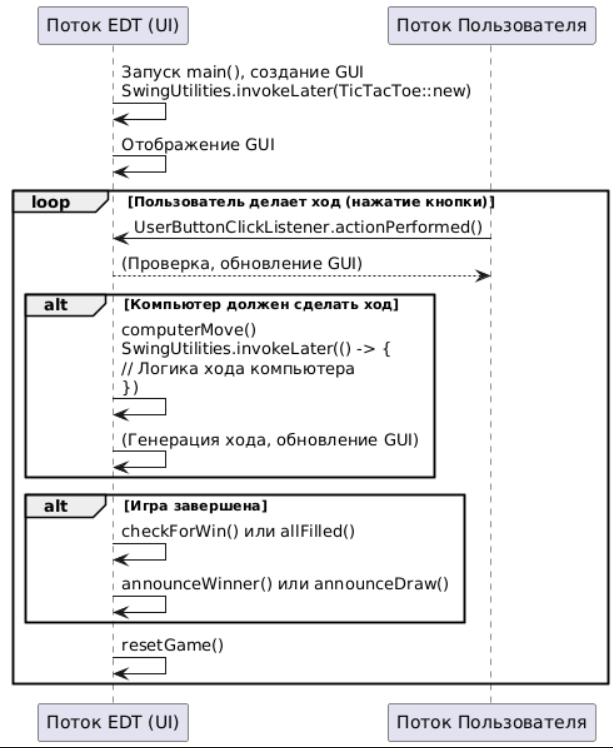
**Задание**:



UML-диаграмма:



Временная диаграмма работы потоков:



**Листинг:**

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Random;

import javax.swing.\*;

public class TicTacToe extends JFrame {

    private JButton[][] buttons = new JButton[3][3];

    private boolean isXTurn = true; // true если ход X, false если O (Компьютер)

    public TicTacToe() {

        setTitle("Крестики-нолики");

        setSize(300, 300);

        setDefaultCloseOperation(EXIT\_ON\_CLOSE);

        setLayout(new GridLayout(3, 3));

        initializeButtons();

        setVisible(true);

    }

    private void initializeButtons() {

        for (int i = 0; i < 3; i++) {

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

                buttons[i][j] = new JButton("");

                buttons[i][j].setFont(new Font("Arial", Font.PLAIN, 60));

                buttons[i][j].setFocusPainted(false);

                buttons[i][j].addActionListener(new UserButtonClickListener(i, j));

                add(buttons[i][j]);

            }

        }

    }

    private class UserButtonClickListener implements ActionListener {

        private int row;

        private int col;

        public UserButtonClickListener(int row, int col) {

            this.row = row;

            this.col = col;

        }

        @Override

        public void actionPerformed(ActionEvent e) {

            if (isXTurn && buttons[row][col].getText().equals("")) {

                buttons[row][col].setText("X");

                isXTurn = false;

                checkForWin();

                if (!isXTurn) {

                    computerMove();

                }

            }

        }

    }

    private void computerMove() {

        SwingUtilities.invokeLater(() -> {

            Random rand = new Random();

            int row, col;

            do {

                row = rand.nextInt(3);

                col = rand.nextInt(3);

            } while (!buttons[row][col].getText().equals(""));

            buttons[row][col].setText("O");

            isXTurn = true;

            checkForWin();

        });

    }

    private void checkForWin() {

        for (int i = 0; i < 3; i++) {

            if (buttons[i][0].getText().equals(buttons[i][1].getText()) &&

                buttons[i][0].getText().equals(buttons[i][2].getText()) &&

                !buttons[i][0].getText().equals("")) {

                announceWinner(buttons[i][0].getText());

            }

            if (buttons[0][i].getText().equals(buttons[1][i].getText()) &&

                buttons[0][i].getText().equals(buttons[2][i].getText()) &&

                !buttons[0][i].getText().equals("")) {

                announceWinner(buttons[0][i].getText());

            }

        }

        if (buttons[0][0].getText().equals(buttons[1][1].getText()) &&

            buttons[0][0].getText().equals(buttons[2][2].getText()) &&

            !buttons[0][0].getText().equals("")) {

            announceWinner(buttons[0][0].getText());

        }

        if (buttons[0][2].getText().equals(buttons[1][1].getText()) &&

            buttons[0][2].getText().equals(buttons[2][0].getText()) &&

            !buttons[0][2].getText().equals("")) {

            announceWinner(buttons[0][2].getText());

        }

        // Проверка на ничью

        boolean allFilled = true;

        for (int i = 0; i < 3; i++) {

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

                if (buttons[i][j].getText().equals("")) {

                    allFilled = false;

                }

            }

        }

        if (allFilled) {

            announceDraw();

        }

    }

    private void announceWinner(String winner) {

        JOptionPane.showMessageDialog(this, "Победитель: " + winner);

        resetGame();

    }

    private void announceDraw() {

        JOptionPane.showMessageDialog(this, "Ничья!");

        resetGame();

    }

    private void resetGame() {

        for (int i = 0; i < 3; i++) {

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

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

            }

        }

        isXTurn = true;

    }

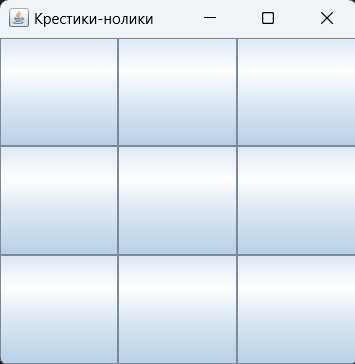
    public static void main(String[] args) {

        SwingUtilities.invokeLater(TicTacToe::new);

    }

}

Начальное окно:



Играем:

Изображение выглядит как снимок экрана, текст, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Конец игры:

