Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра интеллектуальных информационных технологий

Тема: «Приемы разработки оконных клиент-серверных приложений на Java с использованием сокетов»

Отчёт лабораторной работы №6

по дисциплине «Современные платформы программирования»
за II семестр

Выполнил:
студент 3-его курса
VI-го семестр
факультета ЭИС
группы ПО-4(1)
зачётная книжка №190333
Галанин П. И.
«__» _____ 2022 г.

Проверил:
ст. преподаватель
кафедры ИИТ
Крощенко А. А.
«__» _____ 2022 г.

Брест 2022

Отчёт лабораторной работы №6

Тема: «Приемы разработки оконных клиент-серверных приложений на Java с использованием сокетов»

Цель: освоить приемы разработки оконных клиент-серверных приложений на Java с использованием сокетов.

Что нужно сделать:

Разработать клиент-серверное оконное приложение на Java с использованием сокетов и JavaFX.

Можно сделать одну программу с сочетанием функций клиента и сервера либо две отдельных (клиентская часть и серверная часть). Продемонстрировать работу разработанной программы в сети, либо локально (127.0.0.1). Лабораторную работу разрешается выполнять в команде из 2-х человек.

Вариант 8

Игра «Крестики-нолики». Классическая игра для двух игроков на поле 3x3.

Исходный код:

Листинг: src/com/.../TicTacToe.java

```
1  package com.mrwayfarout.tictactoe;
2
3  import java.awt.BasicStroke;
4  import java.awt.Color;
5  import java.awt.Dimension;
6  import java.awt.Font;
7  import java.awt.Graphics;
8  import java.awt.Graphics2D;
9  import java.awt.RenderingHints;
10 import java.awt.Toolkit;
11 import java.awt.event.MouseEvent;
12 import java.awt.event.MouseListener;
13 import java.awt.image.BufferedImage;
14 import java.io.DataInputStream;
15 import java.io.DataOutputStream;
16 import java.io.IOException;
17 import java.net.InetAddress;
18 import java.net.ServerSocket;
19 import java.net.Socket;
20 import java.util.Scanner;
21
22 import javax.imageio.ImageIO;
23 import javax.swing.JFrame;
24 import javax.swing.JPanel;
25
26 public class TicTacToe implements Runnable {
27
28     private String ip = "localhost";
29     private int port = 22222;
30     private Scanner scanner = new Scanner(System.in);
31     private JFrame frame;
```

```
32      private final int WIDTH = 506;
33      private final int HEIGHT = 527;
34      private Thread thread;
35
36      private Painter painter;
37      private Socket socket;
38      private DataOutputStream dos;
39      private DataInputStream dis;
40
41      private ServerSocket serverSocket;
42
43      private BufferedImage board;
44      private BufferedImage redX;
45      private BufferedImage blueX;
46      private BufferedImage redCircle;
47      private BufferedImage blueCircle;
48
49      private String[] spaces = new String[9];
50
51      private boolean yourTurn = false;
52      private boolean circle = true;
53      private boolean accepted = false;
54      private boolean unableToCommunicateWithOpponent = false;
55      private boolean won = false;
56      private boolean enemyWon = false;
57      private boolean tie = false;
58
59      private int lengthOfSpace = 160;
60      private int errors = 0;
61      private int firstSpot = -1;
62      private int secondSpot = -1;
63
64      private Font font = new Font("Verdana", Font.BOLD, 32);
65      private Font smallerFont = new Font("Verdana", Font.BOLD, 20);
66      private Font largerFont = new Font("Verdana", Font.BOLD, 50);
67
68      private String waitingString = "Waiting for another player";
69      private String unableToCommunicateWithOpponentString = "Unable to communicate with opponent.";
70      private String wonString = "You won!";
71      private String enemyWonString = "Opponent won!";
72      private String tieString = "Game ended in a tie.";
73
74      private int[][] wins = new int[][] { { 0, 1, 2 }, { 3, 4, 5 }, { 6, 7, 8 }, { 0, 3, 6 }, { 1,
        4, 7 }, { 2, 5, 8 }, { 0, 4, 8 }, { 2, 4, 6 } };
75
76      /**
77       * <pre>
78       * 0, 1, 2
79       * 3, 4, 5
80       * 6, 7, 8
81       * </pre>
82       */
83
84      public TicTacToe() {
85          System.out.println("Please input the IP: ");
86          ip = scanner.nextLine();
87          System.out.println("Please input the port: ");
88          port = scanner.nextInt();
89          while (port < 1 || port > 65535) {
90              System.out.println("The port you entered was invalid, please input another port: ");
91              port = scanner.nextInt();
92          }
93
```

```
94          loadImages();
95
96          painter = new Painter();
97          painter.setPreferredSize(new Dimension(WIDTH, HEIGHT));
98
99          if (!connect()) initializeServer();
100
101         frame = new JFrame();
102         frame.setTitle("Tic-Tac-Toe");
103         frame.setContentPane(painter);
104         frame.setSize(WIDTH, HEIGHT);
105         frame.setLocationRelativeTo(null);
106         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
107         frame.setResizable(false);
108         frame.setVisible(true);
109
110         thread = new Thread(this, "TicTacToe");
111         thread.start();
112     }
113
114     public void run() {
115         while (true) {
116             tick();
117             painter.repaint();
118
119             if (!circle && !accepted) {
120                 listenForServerRequest();
121             }
122
123         }
124     }
125
126     private void render(Graphics g) {
127         g.drawImage(board, 0, 0, null);
128         if (unableToCommunicateWithOpponent) {
129             g.setColor(Color.RED);
130             g.setFont(smallerFont);
131             Graphics2D g2 = (Graphics2D) g;
132             g2.setRenderingHint(RenderingHints.KEY_TEXT_ANTIALIASING,
    RenderingHints.VALUE_TEXT_ANTIALIAS_ON);
133             int stringWidth =
    g2.getFontMetrics().stringWidth(unableToCommunicateWithOpponentString);
134             g.drawString(unableToCommunicateWithOpponentString, WIDTH / 2 - stringWidth / 2,
    HEIGHT / 2);
135             return;
136         }
137
138         if (accepted) {
139             for (int i = 0; i < spaces.length; i++) {
140                 if (spaces[i] != null) {
141                     if (spaces[i].equals("X")) {
142                         if (circle) {
143                             g.drawImage(redX, (i % 3) * lengthOfSpace + 10 * (i % 3), (int) (i /
    3) * lengthOfSpace + 10 * (int) (i / 3), null);
144                         } else {
145                             g.drawImage(blueX, (i % 3) * lengthOfSpace + 10 * (i % 3), (int) (i /
    3) * lengthOfSpace + 10 * (int) (i / 3), null);
146                         }
147                     } else if (spaces[i].equals("O")) {
148                         if (circle) {
149                             g.drawImage(blueCircle, (i % 3) * lengthOfSpace + 10 * (i % 3), (int)
    (i / 3) * lengthOfSpace + 10 * (int) (i / 3), null);
150                         } else {
```

```
                            g.drawImage(redCircle, (i % 3) * lengthOfSpace + 10 * (i % 3), (int)
(i / 3) * lengthOfSpace + 10 * (int) (i / 3), null);
                        }
                    }
                }
            }
            if (won || enemyWon) {
                Graphics2D g2 = (Graphics2D) g;
                g2.setStroke(new BasicStroke(10));
                g.setColor(Color.BLACK);
                g.drawLine(firstSpot % 3 * lengthOfSpace + 10 * firstSpot % 3 + lengthOfSpace / 2,
(int) (firstSpot / 3) * lengthOfSpace + 10 * (int) (firstSpot / 3) + lengthOfSpace / 2,
secondSpot % 3 * lengthOfSpace + 10 * secondSpot % 3 + lengthOfSpace / 2, (int) (secondSpot /
3) * lengthOfSpace + 10 * (int) (secondSpot / 3) + lengthOfSpace / 2);

                g.setColor(Color.RED);
                g.setFont(largerFont);
                if (won) {
                    int stringWidth = g2.getFontMetrics().stringWidth(wonString);
                    g.drawString(wonString, WIDTH / 2 - stringWidth / 2, HEIGHT / 2);
                } else if (enemyWon) {
                    int stringWidth = g2.getFontMetrics().stringWidth(enemyWonString);
                    g.drawString(enemyWonString, WIDTH / 2 - stringWidth / 2, HEIGHT / 2);
                }
            }
            if (tie) {
                Graphics2D g2 = (Graphics2D) g;
                g.setColor(Color.BLACK);
                g.setFont(largerFont);
                int stringWidth = g2.getFontMetrics().stringWidth(tieString);
                g.drawString(tieString, WIDTH / 2 - stringWidth / 2, HEIGHT / 2);
            }
        } else {
            g.setColor(Color.RED);
            g.setFont(font);
            Graphics2D g2 = (Graphics2D) g;
            g2.setRenderingHint(RenderingHints.KEY_TEXT_ANTIALIASING,
RenderingHints.VALUE_TEXT_ANTIALIAS_ON);
            int stringWidth = g2.getFontMetrics().stringWidth(waitingString);
            g.drawString(waitingString, WIDTH / 2 - stringWidth / 2, HEIGHT / 2);
        }

    }

    private void tick() {
        if (errors >= 10) unableToCommunicateWithOpponent = true;

        if (!yourTurn && !unableToCommunicateWithOpponent) {
            try {
                int space = dis.readInt();
                if (circle) spaces[space] = "X";
                else spaces[space] = "O";
                checkForEnemyWin();
                checkForTie();
                yourTurn = true;
            } catch (IOException e) {
                e.printStackTrace();
                errors++;
            }
        }
    }

    private void checkForWin() {
```

```
209         for (int i = 0; i < wins.length; i++) {
210             if (circle) {
211                 if (spaces[wins[i][0]] == "O" && spaces[wins[i][1]] == "O" && spaces[wins[i][2]]
    == "O") {
212                     firstSpot = wins[i][0];
213                     secondSpot = wins[i][2];
214                     won = true;
215                 }
216             } else {
217                 if (spaces[wins[i][0]] == "X" && spaces[wins[i][1]] == "X" && spaces[wins[i][2]]
    == "X") {
218                     firstSpot = wins[i][0];
219                     secondSpot = wins[i][2];
220                     won = true;
221                 }
222             }
223         }
224     }
225 
226     private void checkForEnemyWin() {
227         for (int i = 0; i < wins.length; i++) {
228             if (circle) {
229                 if (spaces[wins[i][0]] == "X" && spaces[wins[i][1]] == "X" && spaces[wins[i][2]]
    == "X") {
230                     firstSpot = wins[i][0];
231                     secondSpot = wins[i][2];
232                     enemyWon = true;
233                 }
234             } else {
235                 if (spaces[wins[i][0]] == "O" && spaces[wins[i][1]] == "O" && spaces[wins[i][2]]
    == "O") {
236                     firstSpot = wins[i][0];
237                     secondSpot = wins[i][2];
238                     enemyWon = true;
239                 }
240             }
241         }
242     }
243 
244     private void checkForTie() {
245         for (int i = 0; i < spaces.length; i++) {
246             if (spaces[i] == null) {
247                 return;
248             }
249         }
250         tie = true;
251     }
252 
253     private void listenForServerRequest() {
254         Socket socket = null;
255         try {
256             socket = serverSocket.accept();
257             dos = new DataOutputStream(socket.getOutputStream());
258             dis = new DataInputStream(socket.getInputStream());
259             accepted = true;
260             System.out.println("CLIENT HAS REQUESTED TO JOIN, AND WE HAVE ACCEPTED");
261         } catch (IOException e) {
262             e.printStackTrace();
263         }
264     }
265 
266     private boolean connect() {
267         try {
```

```
268             socket = new Socket(ip, port);
269             dos = new DataOutputStream(socket.getOutputStream());
270             dis = new DataInputStream(socket.getInputStream());
271             accepted = true;
272         } catch (IOException e) {
273             System.out.println("Unable to connect to the address: " + ip + ":" + port + " |
    Starting a server");
274             return false;
275         }
276         System.out.println("Successfully connected to the server.");
277         return true;
278     }
279 
280     private void initializeServer() {
281         try {
282             serverSocket = new ServerSocket(port, 8, InetAddress.getByName(ip));
283         } catch (Exception e) {
284             e.printStackTrace();
285         }
286         yourTurn = true;
287         circle = false;
288     }
289 
290     private void loadImages() {
291         try {
292             board = ImageIO.read(getClass().getResourceAsStream("/board.png"));
293             redX = ImageIO.read(getClass().getResourceAsStream("/redX.png"));
294             redCircle = ImageIO.read(getClass().getResourceAsStream("/redCircle.png"));
295             blueX = ImageIO.read(getClass().getResourceAsStream("/blueX.png"));
296             blueCircle = ImageIO.read(getClass().getResourceAsStream("/blueCircle.png"));
297         } catch (IOException e) {
298             e.printStackTrace();
299         }
300     }
301 
302     @SuppressWarnings("unused")
303     public static void main(String[] args) {
304         TicTacToe ticTacToe = new TicTacToe();
305     }
306 
307     private class Painter extends JPanel implements MouseListener {
308         private static final long serialVersionUID = 1L;
309 
310         public Painter() {
311             setFocusable(true);
312             requestFocus();
313             setBackground(Color.WHITE);
314             addMouseListener(this);
315         }
316 
317         @Override
318         public void paintComponent(Graphics g) {
319             super.paintComponent(g);
320             render(g);
321         }
322 
323         @Override
324         public void mouseClicked(MouseEvent e) {
325             if (accepted) {
326                 if (yourTurn && !unableToCommunicateWithOpponent && !won && !enemyWon) {
327                     int x = e.getX() / lengthOfSpace;
328                     int y = e.getY() / lengthOfSpace;
329                     y *= 3;
```

```
                    int position = x + y;

                    if (spaces[position] == null) {
                        if (!circle) spaces[position] = "X";
                        else spaces[position] = "O";
                        yourTurn = false;
                        repaint();
                        Toolkit.getDefaultToolkit().sync();

                        try {
                            dos.writeInt(position);
                            dos.flush();
                        } catch (IOException e1) {
                            errors++;
                            e1.printStackTrace();
                        }

                        System.out.println("DATA WAS SENT");
                        checkForWin();
                        checkForTie();

                    }
                }
            }
        }

        @Override
        public void mousePressed(MouseEvent e) {

        }

        @Override
        public void mouseReleased(MouseEvent e) {

        }

        @Override
        public void mouseEntered(MouseEvent e) {

        }

        @Override
        public void mouseExited(MouseEvent e) {

        }

    }

}
```

Список использованных источников:

1. Learn Socket Programming in Java with Example in Hindi - YouTube – [Электронный ресурс] - Режим доступа: https://www.youtube.com/watch?v=wndMuud3AT8. Дата доступа: 01.05.2022.
2. Классы Socket и ServerSocket в Java – [Электронный ресурс] - Режим доступа: https://javarush.ru/groups/posts/654-klassih-socket-i-serversocket-ili-allo-server-tih-menja-slihshishjh. Дата доступа: 01.05.2022.
3. Как создать исполняемый jar файл в IntelliJ IDEA - YouTube – [Электронный ресурс] - Режим доступа: https://www.youtube.com/watch?v=tA8rEz_xFrQ. Дата доступа: 01.05.2022.
4. Setup IntelliJ IDEA (2021) for JavaFX & SceneBuilder and Create Your First JavaFX Application - YouTube – [Электронный ресурс] - Режим доступа: https://www.youtube.com/watch?v=ZfaPMLdgJxQ. Дата доступа: 01.05.2022.
5. JavaFX - Opening an FXML file in New Window - YouTube – [Электронный ресурс] - Режим доступа: https://www.youtube.com/watch?v=ZzwvQ6pa_tk. Дата доступа: 01.05.2022.
6. How To Fix JavaFX runtime components are missing and are required to run this application - YouTube – [Электронный ресурс] - Режим доступа: https://www.youtube.com/watch?v=sdkW_cUH3hw. Дата доступа: 01.05.2022.
7. Export JavaFX 11, 15 or 17 projects into an executable jar file with IntelliJ [2022] - YouTube – [Электронный ресурс] - Режим доступа: https://www.youtube.com/watch?v=F8ahBtXkQzU. Дата доступа: 01.05.2022.
8. How to make Tic Tac Toe game using JavaFX | Java Game Development - YouTube – [Электронный ресурс] - Режим доступа: https://www.youtube.com/watch?v=tZlZ04Sy3uc. Дата доступа: 21.05.2022.