

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
КАФЕДРА ИИТ

Лабораторная работа №6  
по дисциплине «Современные платформы программирования»

Выполнила:  
Андросюк М.М.  
Группа: ПО-5  
Проверил:  
Крощенко А.А.

Брест 2022

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

**Задание:** Игра «Быки и коровы» Первый игрок задумывает четырехзначное число, так чтобы все цифры числа были разные. Цель второго игрока – отгадать это число. Каждый ход, отгадывающий называет число, тоже четырехзначное и с разными цифрами. Если цифра из называемого числа есть в отгадываемом числе, то эта ситуация называется корова. Если цифра из называемого числа есть в отгадываемом числе и стоит на том же месте, то эта ситуация называется бык. Информация о количестве быков и коров открытая. Дается фиксированное количество попыток. Например, для загаданного числа 0475 называемое число 0251 содержит 1 быка и 1 корову.

## Server:

### Main

```
public class Main {  
  
    public static void main(String[] args) throws IOException {  
        ServerSocket serverSocket = new ServerSocket(8080);  
        new GameHandler(new PlayerThread(serverSocket.accept()), new  
PlayerThread(serverSocket.accept())).start();  
    }  
}
```

### PlayerThread

```
public class PlayerThread extends Thread {  
  
    private final BufferedReader reader;  
    private final BufferedWriter writer;  
    private MessageListener  
messageListener;  
  
    public PlayerThread(Socket socket) throws IOException {  
        reader = new BufferedReader(new InputStreamReader(socket.getInputStream()));  
        writer = new BufferedWriter(new  
OutputStreamWriter(socket.getOutputStream()));  
    }  
  
    public void setMessageListener(MessageListener  
messageListener) { this.messageListener = messageListener;  
    }  
  
    @Override  
    public void run() {
```

```

    }
    try {
        while (true) {
            messageListener.processMessage(reader.readLine(
            ));
        }
    } catch (IOException ignored) {}
}

public void sendMessage(String message)
{ try {
    writer.write(message + "\n");
    writer.flush();
} catch (IOException ignored) {}
}
}

```

### **MessageListener**

```

public interface MessageListener {
    void processMessage(String message);
}

```

### **GameHandler**

```

public class GameHandler implements MessageListener {

    private PlayerThread numberGuesser;
    private PlayerThread numberPicker;
    private static final int maxGuessCount
    = 10; private short guessCount;
    private String number;

    public GameHandler(PlayerThread firstPlayer, PlayerThread
    secondPlayer) { firstPlayer.setMessageListener(this);
    secondPlayer.setMessageListener(this);
    boolean isFirstPlayerGuesser = new Random().nextBoolean();
    numberGuesser = isFirstPlayerGuesser ? firstPlayer :
    secondPlayer; numberPicker = isFirstPlayerGuesser ?
    secondPlayer : firstPlayer;
    }

    public void start() {
        numberGuesser.start();
        numberPicker.start();
        numberPicker.sendMessage("Выберите число");
    }

    private void
    restartGame() {
        guessCount = 0;

        number = null;
        PlayerThread numberGuesserTmp = numberGuesser;
        numberGuesser = numberPicker;
        numberPicker = numberGuesserTmp;
        numberPicker.sendMessage("Выберите число");
    }
}

```

@Override

```

    }
    public void processMessage(String
        message) { if (number == null) {
            number = message;
            numberGuesser.sendMessage("Угадайте число");
            return;
        }

        if (++guessCount == maxGuessCount && !message.equals(number))
            { numberGuesser.sendMessage("Проигрыш");
              numberPicker.sendMessage("Противник проиграл");
              restartGame();
              return;
            }

        if (message.equals(number)) {
            numberGuesser.sendMessage("Победа");
            numberPicker.sendMessage("Противник победил");
            restartGame();
            return;
        }

        int bullCount =
        0; int cowCount
        = 0;

        for (int i = 0; i < message.length(); i++) {
            if (number.charAt(i) == message.charAt(i)) {
                bullCount++;
            }
            else if (number.contains(Character.toString(message.charAt(i)))) {
                cowCount++;
            }
        }

        String messageToSend = "Попытка " + guessCount + ". Быки: " + bullCount + ". Коровы:
" + cowCount;
        numberGuesser.sendMessage(messageToSend);
        numberPicker.sendMessage(messageToSend);
    }
}

```

#### **Client:**

##### **GameController**

```

public class GameController
{
    @FXML
    private Label result;
    @FXML
    private Label command;
    @FXML
    private TextField
    numberInput; @FXML
    private Button enterButton;

    private BufferedWriter
    writer; private boolean
    isGuesser;
}

```

```

}
public GameController() {
    new Thread(() -> {
        try {
            Socket socket = new Socket("localhost", 8080);
            BufferedReader reader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            writer = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()));

            while (true) {
                String str = reader.readLine();

                if (str.equals("Угадайте
число")) { isGuesser = true;
            }

            if (str.equals("Выберите число") || str.equals("Угадайте
число")) { Platform.runLater(() -> command.setText(str));
                Platform.runLater(() -> enterButton.setDisable(false));
                continue;
            }

            Platform.runLater(() ->

result.setText(str)); if

(!str.startsWith("Попытка")) {
                Thread.sleep(5000);
                isGuesser = false;
                Platform.runLater(() -> result.setText(""));
            }
        }
    });
}

```

```

    }
    } catch (IOException | InterruptedException ignored) { }
  }).start();
}

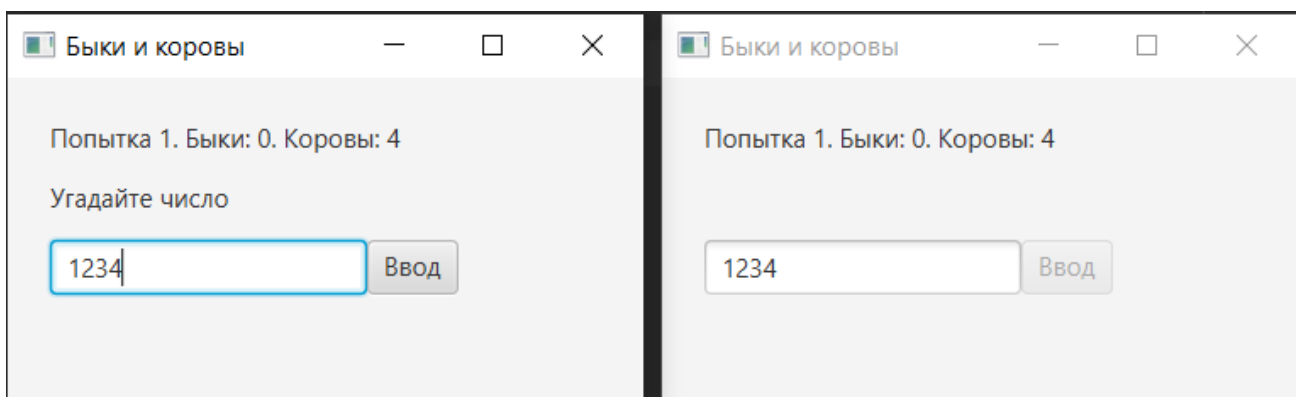
@FXML
protected void onClick() throws IOException {
    List<Character> characters = numberInput.getText().chars().mapToObj(character ->
(char) character).toList();
    if (characters.size() != 4 || new HashSet<>(characters).size() != 4 ||
!characters.stream().allMatch(Character::isDigit)) {
        command.setText("Неверный ввод. Попробуйте еще раз");
        return;
    }

    writer.write(numberInput.getText() +
"\n"); writer.flush();

    if (!isGuesser) {
        command.setText("");
        enterButton.setDisable(true);
        return;
    }

    command.setText("Угадайте число");
}
}

```



**Вывод:** Разработано клиент-серверное оконное приложение на Java с использованием сокетов и JavaFX.