

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«Вятский государственный университет»**  
Факультет автоматики и вычислительной техники  
Кафедра электронных вычислительных машин

Лабораторная работа №3 по курсу  
«Разработка программных систем»

Выполнил студент группы ИВТ-32 \_\_\_\_\_ /Рзаев А. Э./  
Проверил доцент кафедры ЭВМ \_\_\_\_\_ /Чистяков Г. А./

Киров 2017

## 1 Задание

В ходе выполнения лабораторной работы необходимо решить следующие задачи:

- Выбрать и согласовать с преподавателем задачу, для решения которой может быть использована программа, разработанная в ходе предыдущей лабораторной работы.
- Разработать программу для решения выбранной задачи (взаимодействие с пользователем должно осуществляться с применением пользовательского интерфейса).

## 2 Листинг программы

Листинг программы приведен в приложении А.

## 3 Экранные формы

Экранные формы приведены в приложении Б.

## 4 Вывод

В ходе выполнения лабораторной работы были изучены основы технологии Swing, ее основные компоненты. Написано приложение с графическим пользовательским интерфейсом, предназначенное для создания списков случайных чисел, представляемых в форматах JSON и XML (на выбор пользователя).

Приложение А  
(обязательное)  
Листинг программы

**Lab3.java**

```
package rpslabs;

import javax.swing.*;

public class Lab3 {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new PrimeNumberGenerator());
    }
}
```

**PrimeNumberGenerator.java**

```
package rpslabs;

import java.awt.Dimension;
import javax.swing.*;

public class PrimeNumberGenerator implements Runnable {
    private static final String MAIN_FRAME_TITLE = "Prime random number generator";
    private static final String NUMBER_PANEL_TITLE = "Random number";
    private static final String ARRAY_PANEL_TITLE = "Array of random numbers";
    private JFrame mainFrame;

    @Override
    public void run() {
        mainFrame = new JFrame(MAIN_FRAME_TITLE);
        mainFrame.setPreferredSize(new Dimension(640, 480));

        JPanel primeRandomNumberPanel = new PrimeRandomNumberPanel();
        JPanel primeRandomArrayPanel = new PrimeRandomArrayPanel();

        JTabbedPane mainPane = new JTabbedPane();
        mainPane.addTab(NUMBER_PANEL_TITLE, primeRandomNumberPanel);
        mainPane.addTab(ARRAY_PANEL_TITLE, primeRandomArrayPanel);

        mainFrame.getContentPane().add(mainPane);
        mainFrame.pack();
        mainFrame.setVisible(true);
    }
}
```

**PrimeRandomArrayPanel.java**

```
package rpslabs;

import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.util.Arrays;
import java.util.stream.Collectors;
import javax.swing.*;
import javax.swing.border.EmptyBorder;

public class PrimeRandomArrayPanel extends JPanel {
    private static final long serialVersionUID = 1L;
    private static final String ARRAY_LENGTH_LABEL = "Length of array: ";
```

```

private static final String PRIME_ARRAY_BUTTON_LABEL = "Get array of ranodm
    numbers";
private static final String FORMAT_LABEL = "Data format: ";
private static final String JSON_FORMAT = "JSON";
private static final String XML_FORMAT = "XML";
private static final int MAX_ARRAY_LENGTH = 10000000;
private static final String INVALID_ARRAY_LENGTH = String.format(
    "Length of array must be in (0; %s]",
    MAX_ARRAY_LENGTH
);

private JButton primeArrayButton;
private JTextArea primeArrayTextArea;
private JTextField arrayLengthField;
private JLabel arrayLengthLabel;
private String dumpFormat = JSON_FORMAT;
private PrimeRandom random = new PrimeRandom();

public PrimeRandomArrayPanel() {
    super();

    this.setLayout(new BorderLayout(this, BorderLayout.Y_AXIS));
    this.setBorder(new EmptyBorder(10, 10, 10, 10));

    primeArrayButton = new JButton(PRIME_ARRAY_BUTTON_LABEL);
    primeArrayButton.addActionListener(this::onPrimeArrayButtonClicked);

    primeArrayTextArea = new JTextArea();
    primeArrayTextArea.setEditable(false);

    arrayLengthField = new JTextField();
    arrayLengthField.setMaximumSize(new Dimension(200, 30));
    arrayLengthLabel = new JLabel(ARRAY_LENGTH_LABEL);

    JPanel lengthPanel = new JPanel();
    lengthPanel.setLayout(new BorderLayout(lengthPanel, BorderLayout.X_AXIS));
    lengthPanel.add(arrayLengthLabel);
    lengthPanel.add(arrayLengthField);
    lengthPanel.add(primeArrayButton);

    JRadioButton selectJSON = new JRadioButton(JSON_FORMAT);
    selectJSON.setActionCommand(JSON_FORMAT);
    selectJSON.setSelected(true);
    selectJSON.addActionListener(e -> dumpFormat = e.getActionCommand());

    JRadioButton selectXML = new JRadioButton(XML_FORMAT);
    selectXML.setActionCommand(XML_FORMAT);
    selectXML.addActionListener(e -> dumpFormat = e.getActionCommand());

    ButtonGroup formatGroup = new ButtonGroup();
    formatGroup.add(selectJSON);
    formatGroup.add(selectXML);

    JPanel formatPanel = new JPanel();
    formatPanel.setLayout(new BorderLayout(formatPanel, BorderLayout.X_AXIS));
    formatPanel.add(new JLabel(FORMAT_LABEL));
    formatPanel.add(selectJSON);
    formatPanel.add(selectXML);

    this.add(lengthPanel);
    this.add(Box.createVerticalStrut(10));

```

```

        this.add(formatPanel);
        this.add(Box.createVerticalStrut(10));
        this.add(new JScrollPane(primeArrayTextArea));
    }

    private String dumpArrayToJSON(int[] array) {
        StringBuilder serialized = new StringBuilder();
        serialized.append("[\n");
        serialized.append(
            Arrays.stream(array)
                .mapToObj(Integer::toString)
                .map(v -> "    " + v + "\n")
                .collect(Collectors.joining(""))
        );
        serialized.append("]");
        return serialized.toString();
    }

    private String dumpArrayToXML(int[] array) {
        StringBuilder serialized = new StringBuilder();
        serialized.append("<?xml version=\"1.0\" ?>\n");
        serialized.append("<array>\n");
        serialized.append(
            Arrays.stream(array)
                .mapToObj(Integer::toString)
                .map(v -> "    <item>" + v + "</item>\n")
                .collect(Collectors.joining(""))
        );
        serialized.append("</array>");
        return serialized.toString();
    }

    private void onPrimeArrayButtonClicked(ActionEvent event) {
        try {
            int length = Integer.parseInt(arrayLengthField.getText());

            if (length <= 0 || length > MAX_ARRAY_LENGTH) {
                JOptionPane.showMessageDialog(this, INVALID_ARRAY_LENGTH);
            } else {
                int[] array = random.getRandomArray(length);

                if (dumpFormat.equals(JSON_FORMAT)) {
                    primeArrayTextArea.setText(dumpArrayToJSON(array));
                } else {
                    primeArrayTextArea.setText(dumpArrayToXML(array));
                }
            }
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, INVALID_ARRAY_LENGTH);
        }
    }
}

```

### PrimeRandomNumberPanel.java

```

package rpslabs;

import java.awt.Dimension;
import java.awt.event.ActionEvent;

```

```

import javax.swing.*;
import javax.swing.border.EmptyBorder;

public class PrimeRandomNumberPanel extends JPanel {
    private static final long serialVersionUID = 1L;
    private static final String PRIME_NUMBER_BUTTON_LABEL = "Get prime random
        number";
    private PrimeRandom random = new PrimeRandom();
    private JTextField primeNumberField;
    private JButton primeNumberButton;

    public PrimeRandomNumberPanel() {
        super();

        this.setLayout(new BoxLayout(this, BoxLayout.X_AXIS));
        this.setBorder(new EmptyBorder(10, 10, 10, 10));

        primeNumberField = new JTextField();
        primeNumberField.setMaximumSize(new Dimension(200, 30));
        primeNumberField.setEditable(false);

        primeNumberButton = new JButton(PRIME_NUMBER_BUTTON_LABEL);
        primeNumberButton.addActionListener(this::onPrimeNumberButtonClicked);

        this.add(primeNumberButton);
        this.add(Box.createHorizontalStrut(10));
        this.add(primeNumberField);
    }

    private void onPrimeNumberButtonClicked(ActionEvent event) {
        primeNumberField.setText(Integer.toString(random.getRandomNumber()));
    }
}

```

Приложение Б  
(обязательное)  
Экранные формы

The screenshot shows a web application window titled "Генератор простых случайных чисел". It has two tabs: "Случайное число" (selected) and "Список случайных чисел". In the "Случайное число" tab, there is a button labeled "Выдать простое случайное число" and a text input field containing the number "288559663".

Рисунок 1 – Вкладка для генерации единственного случайного числа.

The screenshot shows the same application window, but with the "Список случайных чисел" tab selected. It features a text input field for "Длина массива:" with the value "100", and a button "Выдать список случайных чисел". Below these is a radio button group for "Формат данных:" with "JSON" and "XML" options; "XML" is selected. The main area displays an XML array of 20 random numbers.

```
<?xml version="1.0" ?>
<array>
  <item>8071001</item>
  <item>19836923</item>
  <item>34779059</item>
  <item>35171737</item>
  <item>37614869</item>
  <item>45655513</item>
  <item>46777559</item>
  <item>51123497</item>
  <item>77627113</item>
  <item>78792269</item>
  <item>83343731</item>
  <item>85253483</item>
  <item>98645419</item>
  <item>117308027</item>
  <item>131718569</item>
  <item>135689479</item>
  <item>138014827</item>
  <item>157110581</item>
  <item>161550959</item>
  <item>171945689</item>

```

Рисунок 2 – Вкладка для генерации списка случайных чисел.