МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(«ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №3

по дисциплине «Разработка программных систем»

Выполнил студент группы ИВТ-32 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щесняк Д. С./

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г. А./

Киров 2016

1. Задание на лабораторную работу

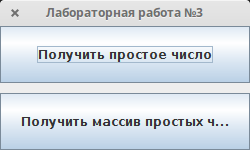
Разработать программу для решения выбранной задачи (взаимодействие с пользователем должно осуществляться с применением пользовательского интерфейса).

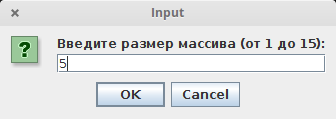
1. Структуры классов

|  |  |  |
| --- | --- | --- |
| **Класс Prime** |  | **Класс Main** |
| Public:   * Int getRandomPrime() * Int[] getRandomArray(int length) |  | Public:   * Void main(String[] args) * Void printMenu() |
| Private:   * Int getNext(int prime) * Boolean isPrime(int arg0) |  | Private: |

1. Экранные формы

Экранные формы представлены на рисунках 1-3

Рисунок 1 – Экранная форма главного экрана

Рисунок 2 – Ввод размера массива

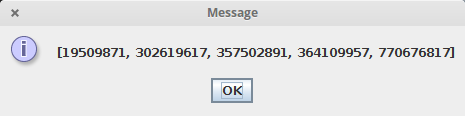


Рисунок 3 – Вывод массива простых чисел

1. Исходный код программы

Исходный код класса App представлен на рисунке 4. Исходный код класса Prime представлен на рисунке 5. Исходный код класса Gui представлен на рисунке 6.

|  |
| --- |
| public class App {  public static void main(String[] args) {  Gui app = new Gui();  app.setVisible(true);  }  } |

Рисунок 4 – Класс App

|  |
| --- |
| /\*\*  \* Created by Danill on 22.09.2016.  \*/  import java.util.Arrays;  import java.util.Random;  public class Prime {  /\*\*  \* Функция для получения следующего простого числа после num  \* @param num любое натуральное число  \* @return Ближайшее простое число, которое следует за num  \*/  private static int getNext(int num) {  int i = num + 1;  for(; !isPrime(i); i++) {}  return i;  }  /\*\*  \* Функция для генерации простого числа  \* @return Случайное простое число, которое находится между 0 и 99999999  \*/  public static int getRandomPrime() {  Random r = new Random();  return getNext(r.nextInt(999999999));  }  /\*\*  \* Функция для проверки является ли число простым  \* @param n число, которое требудется проверить  \* @return true если число является простым и false если нет  \*/  private static boolean isPrime(int n) {  for (int j = 2; j < Math.sqrt(n); j++) {  if (n % j == 0) {  return false;  }  }  return true;  }  /\*\*  \* Функция, которая возвращает отсортированный по возрастанию массив из простых чисел заданной длины  \* @param length Длина массива  \* @return отсортированный по возрастанию массив заполненный случайными чисоами  \*/  public static int[] getRandomArray(int length) {  int[] arr = new int[length];  for(int i = 0; i < length; i++) {  arr[i] = getRandomPrime();  }  Arrays.sort(arr);  return arr;  }  } |

Рисунок 5 – Класс Prime

|  |
| --- |
| import javax.swing.\*;  import java.awt.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.util.Arrays;  /\*\*  \* Created by Daniil on 06.10.2016.  \*/  public class Gui extends JFrame {  private JButton randPrimeBut = new JButton("Получить простое число");  private JButton randPrimeArrayBut = new JButton("Получить массив простых чисел");  public Gui() {  super("Лабораторная работа №3");  this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  this.setSize(250, 150);  this.setResizable(false);  Container content = this.getContentPane();  content.setLayout(new GridLayout(0, 1, 10, 10));  content.add(randPrimeBut);  content.add(randPrimeArrayBut);  randPrimeArrayBut.addActionListener(new OnRandPrimeArrayButClick());  randPrimeBut.addActionListener(new OnRandPrimeButClick());  }  /\*\*  \* Класс-обработчик, который обрабатывает нажатие  \* на кнопку "Получить массив простых чисел"  \*/  class OnRandPrimeArrayButClick implements ActionListener {  @Override  public void actionPerformed(ActionEvent e) {  String v = JOptionPane.showInputDialog("Введите размер массива (от 1 до 15):").trim();  try {  int length = Integer.valueOf(v);  if ( (length < 1) || (length > 15) ) {  throw new NumberFormatException();  }  JOptionPane.showMessageDialog(null, Arrays.toString(Prime.getRandomArray(length)));  } catch (NumberFormatException except) {  JOptionPane.showMessageDialog(null, "Введен неверный параметр");  }  }  }  /\*\*  \* Класс-обработчик, который обрабатывает клик  \* по кнопке "Вернуть простое число"  \*/  class OnRandPrimeButClick implements ActionListener {  @Override  public void actionPerformed(ActionEvent e) {  JOptionPane.showMessageDialog(null, String.valueOf(Prime.getRandomPrime()));  }  }  } |

Рисунок 6 – Класс Gui

1. Вывод

В ходе выполнения лабораторной работы были полученные необходимые знания базового синтаксиса языка программирования Java. Было изучено создание документации средствами JavaDoc и методы рефакторинга кода. На основе полученных знаний была разработана документация для класса Prime. Данные знания являются фундаментальными и необходимы для дальнейшего продолжения изучения языка Java.