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 №2

по дисциплине «Разработка программных систем»

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

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Долженкова М. Л./

Киров 2016

1. Задание на лабораторную работу

* Провести ряд преобразований программного кода, полученного в ходе выполнения предыдущей работы, с использованием встроенных средств рефакторинга
* Сопроводить код комментариями с использование JavaDoc
* Сгенерировать документацию к разработанным классам

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. Рефакторинг

В ходе выполнения лабораторной работы были использованы средства рефакторинга. Например переименование переменной.

1. Экранные формы JavaDoc

Экранные формы представлены на рисунке 1

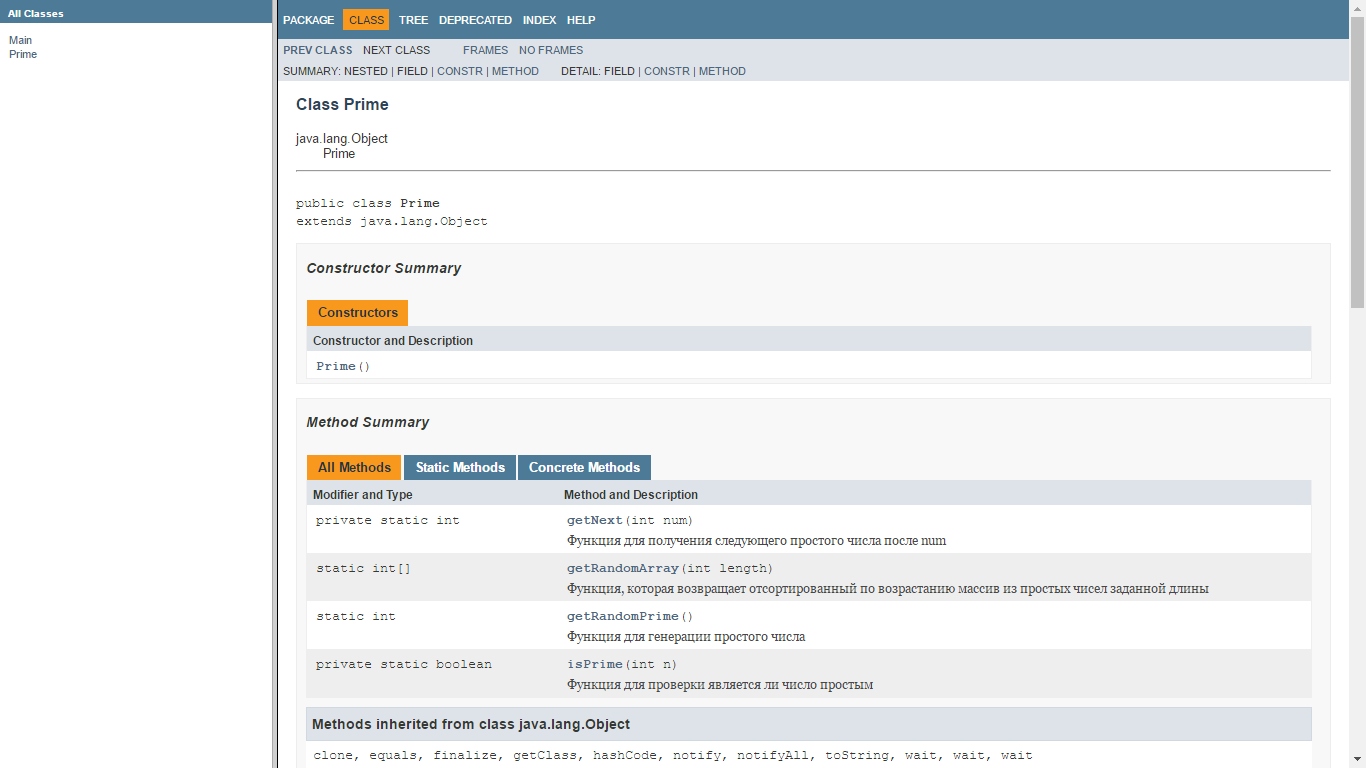


Рисунок 1 – Экранная форма JavaDoc

1. Исходный код программы

Исходный код класса Main представлен на рисунке 2. Исходный код класса Prime представлен на рисунке 3.

|  |
| --- |
| /\*\*  \* Created by Danill on 22.09.2016.  \*/  import java.util.\*;  public class Main {  /\*\*  \* Процедура, которая организует вывод текстового меню на экран  \*/  public static void printMenu() {  System.out.println("1) Выдать массив случайных чисел");  System.out.println("2) Выдать случайное число");  System.out.println("\n\n0) Выход");  }  /\*\*  \* Функция считывания целого числа и вывода сообщений при ошибочных данных  \* @return первое правильное прочитанное целочисленное число  \*/  public static int readInt() {  Scanner s = new Scanner(System.in);  while(!s.hasNextInt()) {  System.out.println("Неправильный ввод! Попробуйте снова");  if (s.hasNextLine()) s.nextLine();  }  return s.nextInt();  }  public static void main(String[] args) {  Scanner s = new Scanner(System.in);  int key;  do {  printMenu();  key = readInt();  switch (key) {  case 1:  System.out.print("Введите размер массива: ");  int len = readInt();  int[] arr = Prime.getRandomArray(len);  for(int i = 0; i < len; i++) System.out.print(arr[i] + " ");  System.out.print("\n");  break;  case 2:  System.out.print(Prime.getRandomPrime() + "\n");  break;  default:  break;  }  } while (key != 0);  s.close();  }  } |

Рисунок 2 – Класс Main

|  |
| --- |
| /\*\*  \* 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 3 – Класс Prime

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.