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 №1 по курсу «Разработка программных систем»

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

Киров 2017

1 Задание

Разработать класс для генерации массива простых чисел. Класс должен иметь два публичных метода:

- int getRandomNumber() - возвращает простое число из диапазона $[2, 10^9]$;
- int[] getRandomArray(int length) - возвращает упорядоченный по возрастанию массив простых чисел размерности length.

и два внутренних метода:

- int getNext(int prime) - возвращает следующее после prime простое число из диапазона $[2, 10^9]$;
- boolean isPrime(int number) - определяет является ли заданный аргумент простым числом.

Продемонстрировать работу класса.

2 Листинг программы

Листинг программы приведен в приложении А.

3 Вывод

В ходе выполнения лабораторной работы были изучены основные конструкции языка программирования Java, структура программы, стандартные средства ввода/вывода; изучен основной функционал интегрированной среды разработки Eclipse; написана программа для генерации простых чисел.

Приложение А
(обязательное)
Листинг программы

Lab1.java

```
package rzaevali;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Lab1 {

    public static void main(String[] args) throws IOException {
        Rnd rnd = new Rnd();
        BufferedReader reader = new BufferedReader(new
           InputStreamReader(System.in));

        while (true) {
            System.out.print(
                    ”What do you want to do\n” +
                    ”1. Get random prime number\n” +
                    ”2. Get array of prime numbers\n” +
                    ”3. Quit\n”);
            String line = reader.readLine();
            int c;
            try {
                c = Integer.parseInt(line);
            } catch (NumberFormatException ignore) {
                System.out.println(”Unknown command”);
                continue;
            }

            if (c == 1) {
                System.out.println(rnd.getRandomNumber());
            } else if (c == 2) {
                System.out.println(”Enter the length of array:”);
                int length;
                while (true) {
                    line = reader.readLine();
                    try {
                        length = Integer.parseInt(line);
                    } catch (NumberFormatException ignore) {
                        System.out.println(”The length of the array
                           must be positive integer number”);
                        continue;
                    }

                    if (length <= 0) {
                        System.out.println(”The length of the array
                           must be positive integer number”);
                        continue;
```

```
                    } else {
                        break;
                    }
                }

                int[] array = rnd.getRandomArray(length);
                for (int v : array) {
                    System.out.println(v);
                }
            } else if (c == 3) {
                break;
            } else {
                System.out.println("Unknown command");
            }
        }
    }

}
```

Rnd.java

```
package rzaevali;

import java.util.Random;
import java.util.Arrays;

public class Rnd {

    private Random _random;

    private static final int LAST_PRIME_NUMBER = 999999937;

    public Rnd() {
        _random = new Random();
    }

    public int getRandomNumber() {
        int n = 2 + _random.nextInt(LAST_PRIME_NUMBER - 1);
        return isPrime(n) ? n : getNext(n);
    }

    public int[] getRandomArray(int length) {
        int[] array = new int[length];

        for (int i = 0; i < length; ++i) {
            array[i] = getRandomNumber();
        }
        Arrays.sort(array);

        return array;
    }

    private int getNext(int number) {
```

```
    for (int i = number + 1; i < 1000000000; ++i) {
        if (isPrime(i)) {
            return i;
        }
    }
    return -1;
}

private boolean isPrime(int number) {
    if (number == 2) {
        return true;
    } else {
        return pollardP1Test(number) == 1 && fermatTest(number)
           == 1;
    }
}

private int fermaTest(int number) {
    int x = (int) Math.sqrt(number), y = 0;
    int r = x * x - y * y - number;
    for (;;) {
        if (r == 0) {
            return x != y ? x - y : x + y;
        } else {
            if (r > 0) {
                r -= y + y + 1;
                ++y;
            } else {
                r += x + x + 1;
                ++x;
            }
        }
    }
}

private int gcd(int a, int b) {
    if (a == 0) {
        return b;
    } else {
        return gcd(b % a, a);
    }
}

private int mulmod(long a, long b, long m) {
    return (int) ((a * b) % m);
}

private long powmod(long a, long n, long m) {
    long res = 1;
    for (int i = 0; i < n; ++i) {
        res = (res * a) % m;
    }
```

```
        return res;
    }

    private int pollardP1Test(int number) {
        final int b = 13;
        final int[] q = { 2, 3, 5, 7, 11, 13 };

        int a = 5 % number;
        for (int j = 0; j < 10; ++j) {
            while (gcd(a, number) != 1) {
                a = mulmod(a, a, number);
                a += 3;
                a %= number;
            }

            for (int i = 0; i < q.length; ++i) {
                int qq = q[i];
                int e = (int) Math.floor(Math.log((double) b) / Math.
                   log((double) qq));
                int aa = (int) powmod(a, powmod(qq, e, number),
                   number);
                if (aa == 0) {
                    continue;
                }

                int g = gcd(aa - 1, number);
                if (1 < g && g < number) {
                    return g;
                }
            }
        }

        return 1;
    }
}
```