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 №1

по дисциплине «Теория принятия решений»

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

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Скворцов А. А./

Киров 2017

1. Задание на лабораторную работу

Целью лабораторной работы является освоение технологии и методики построения экспертных систем на примере разработки учебной экспертной системы. Студент выступает в роли одновременно эксперта и инженера по знаниям.

1. Выполнение лабораторной работы

Гипотезы:

* С++
* Java
* Python
* Pascal
* C#
* Java Script
* Ruby
* PHP

Таблица параметров представлена в таблице 1. Таблица сопоставления гипотез и параметров представлена в таблице 2 и таблице 3.

Таблица 1

|  |  |
| --- | --- |
| Параметр | Возможные значения |
| Скорость выполнения | Высокая, средняя, низкая |
| Скорость разработки | Высокая, средняя, низкая |
| Тип | Компиляция, интерпретация |
| Среда выполнения | Кроссплатформенная, не кроссплатформенная |
| Сложность изучения | Высокая, низкая |
| Востребованность | Высокая, средняя, низкая |
| Уровень | Высокий, средний низкий |

Таблица 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Гипотеза | Скорость Выполнения | Скор. Разработки | Тип | Среда выполнения |
| С++ | Высокая | Низкая | Компилируемый | не кроссплатформенный |
| Java | Средняя | Средняя | Интерпретируемый | Кроссплатформенный |
| Python | Низкая | Высокая | Интерпретируемый | Кроссплатформенный |
| Pascal | Высокая | Низкая | Компилируемый | не кроссплатформенный |
| C# | Средняя | Средняя | Компилируемый | Не кроссплатформенный |
| Java Script | Средняя | Низкая | Интерпретируемый | Кроссплатформенный |
| Ruby | Средняя | Средняя | Интерпретируемый | Кроссплатформенный |
| PHP | Средняя | Средняя | Интерпретируемый | Кроссплатформенный |

Таблица 3

|  |  |  |  |
| --- | --- | --- | --- |
| Гипотеза | Сложность | Востребованность | уровень |
| С++ | Высокая | Высокая | Низкий |
| Java | Средняя | Высокая | Средний |
| Python | Низкая | Высокая | Высокий |
| Pascal | Низкая | Низкая | Низкий |
| C# | Средняя | Средняя | Средняя |
| Java Script | Средняя | Высокая | Низкий |
| Ruby | Низкая | Средняя | Высокий |
| PHP | Низкая | Высокая | Средний |

1. Экранные формы

Экранные формы представлены на рисунке 1

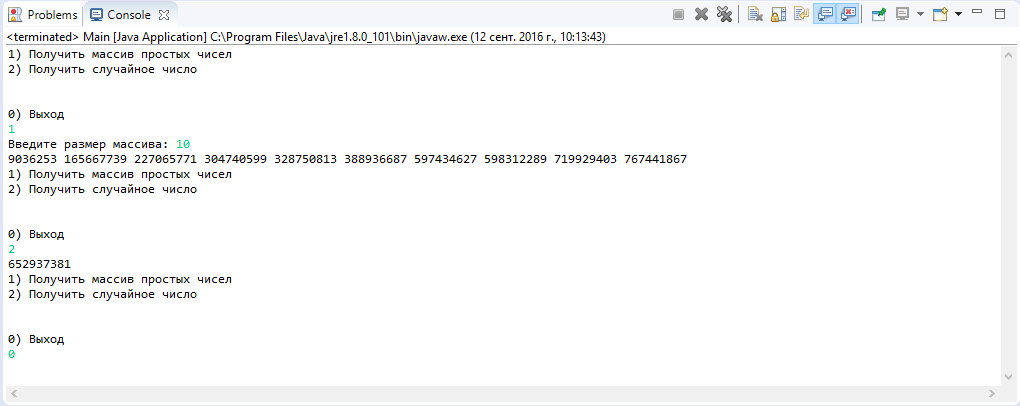


Рисунок 1 – Демонстрация работы программы

1. Исходный код программы

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

|  |
| --- |
| import java.util.\*;  public class Main {    public static void printMenu() {  System.out.println("1) Получить массив простых чисел");  System.out.println("2) Получить случайное число");  System.out.println("\n\n0) Выход");  }  public static void main(String[] args) {  Scanner s = new Scanner(System.in);    int key;    do {  printMenu();  key = s.nextInt();    switch (key) {  case 1:  System.out.print("Введите размер массива: ");  int len = s.nextInt();  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

|  |
| --- |
| import java.util.Arrays;  import java.util.Random;  public class Prime {    private static int getNext(int prime) {  int i = prime + 1;    for(; !isPrime(i); i++) {}    return i;  }    public static int getRandomPrime() {  Random r = new Random();  return getNext(r.nextInt(999999999));  }    private static boolean isPrime(int n) {  for (int j = 2; j < Math.sqrt(n); j++) {  if (n % j == 0) {  return false;  }  }    return true;  }    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 объектно-ориентированное программирование и на основе полученных знаний был разработан класс позволяющий генерировать случайные простые числа и упорядоченные массивы из них. Данные знания являются фундаментальными и необходимы для дальнейшего продолжения изучения языка Java.