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 №6

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

В соответствии с выбранным вариантом разработать модуль в функциональном стиле на языке программирования Python.

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

* getRandomPrime() – возвращает случайное простое число из диапазона [2, 109];
* getRandomArray(int length) – возвращает упорядоченный по возрастанию массив простых чисел размерности length;
* getNext(int prime) – возвращает следующее после prime простое число в диапазоне [2, 109]
* isPrime(int arg0) – определяет является ли заданный аргумент простым числом

Продемонстрировать работу модуля

1. Экранные формы

Экранные формы представлены на рисунке 1

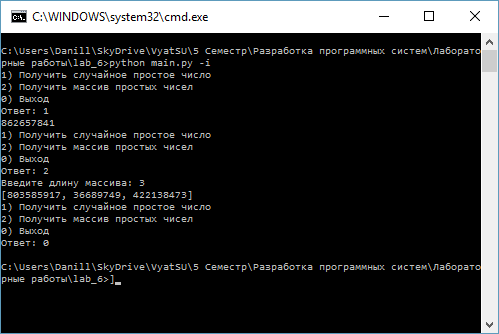


Рисунок 1 – Демонстрация работы программы

1. Исходный код программы

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

|  |
| --- |
| import prime  import optparse  def interact():  k = 1  while k != 0:  print("1) Получить случайное простое число")  print("2) Получить массив простых чисел")  print("0) Выход")  s = input("Ответ: ")  if s.isdigit():  k = int(s)  else:  print("Неверный ввод")  continue  if k == 1:  print(prime.getRandomPrime())  elif k == 2:  s = input("Введите длину массива: ")  if s.isdigit():  length = int(s)  else:  print("Неверный ввод")  continue  if length > 0:  print(prime.getRandomArray(length))  else:  print("Неверный диапазон")  opt = optparse.OptionParser('Lанная программа позволяет получить простые числа. Для справки используйте -h')  opt.add\_option("-n", "--number", dest="count", type="int", help="количество чисел")  opt.add\_option("-c", "--check", dest="check", type="int", help="проверить число на простоту")  opt.add\_option("-a", "--after", dest="after", type="int", help="вернуть простое число, которое следует после")  opt.add\_option("-i", "--interactive", action="store\_true", dest="interact", help="войти в интерактивный режим")  opt.set\_defaults(interact = False)  (options, args) = opt.parse\_args()  if (not options):  opt.print\_usage()  exit(0)  if (options.count != None) and (0 < options.count <= 100):  print(", ".join(map(str, prime.getRandomArray(options.count))))  if (options.check != None) and (options.check > 1):  print(prime.isPrime(options.check))  if (options.after != None) and (options.after > 0):  print(prime.getNext(options.after))  if (options.interact):  interact() |

Рисунок 2 – Класс Main

|  |
| --- |
| import random  getNext = lambda i: getNext(i + 1) if not isPrime(i + 1) else i + 1  isPrime = lambda i: len(list(filter(lambda x: i % x == 0, range(2, int(i\*\*0.5) + 1)))) == 0  getRandomPrime = lambda: getNext(random.randint(2, 10\*\*9))  getRandomArray = lambda length: [getRandomPrime() for i in range(length)] |

Рисунок 3 – модуль Prime

1. Вывод

В ходе выполнения лабораторной работы были полученные необходимые знания базового синтаксиса языка программирования Python. Было изучено функциональное программирование и на основе полученных знаний был разработан модуль позволяющий генерировать случайные простые числа и массивы из них. Данные знания являются фундаментальными и необходимы для дальнейшего продолжения изучения языка Python.