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 №6 по курсу «Разработка программных систем»

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

Киров 2017

1 Задание

В ходе выполнения лабораторной работы необходимо решить следующие задачи:

- Установить на компьютер интерпретатор языка Python.
- Установить на компьютер подходящую среду разработки.
- Настроить рабочее окружение.
- Реализовать в отдельном модуле группу методов.
- Продемонстрировать работу модуля на подготовленном сценарии.

Модуль должен иметь четыре метода:

а) `def get_random_prime()` - возвращает случайное простое число из диапазона $[2; 10^9]$.

б) `def get_random_array(length)` - возвращает упорядоченный по возрастанию массив простых чисел размерности length.

в) `def get_next(prime)` - возвращает следующее после prime простое число.

г) `def is_prime(n)` - определяет является ли заданный аргумент простым числом.

2 Листинг программы

Листинг программы приведен в приложении А.

3 Вывод

В ходе выполнения лабораторной работы был установлен интерпретатор CPython, среда разработки PyCharm; изучены основные конструкции языка Python, написан в функциональном стиле модуль для генерации массива простых чисел, а также программа для демонстрации его работы; изучен формат строк документации Python.

Приложение А
(обязательное)
Листинг программы

primerandom.py

```
from typing import List

_LAST_PRIME_NUMBER = 999999937

def _pollard_p1_test(number: int) -> bool:
    """
    This method determines whether given number is is a probable prime
    using Pollard's p - 1 algorithm

    :param number: Testing number
    :return: True if number is probable prime or False otherwise
    """
    from math import gcd, log, floor

    b = 13
    q = (2, 3, 5, 7, 11, 13)

    a = 5 % number

    e = lambda b, v: int(floor(log(b, v)))

    aa = lambda a, v: int(pow(a, pow(v, e(b, v), number), number))

    red = lambda a: a if gcd(a, number) == 1 else ((a ** 2) % number + 3) % number

    test = lambda a: len([gcd(aa(a, v), number) for v in q if aa(a, v) != 0 and 1
       < gcd(aa(a, v), number) < number]) > 0

    gen = lambda a, n: [red(a)] + gen(red(a), n - 1) if n > 0 else []

    return len([i for i in gen(a, 10) if test(i)]) == 0

def is_prime(number: int) -> bool:
    """
    This method determines whether number is prime or not

    :param number: Testing number
    :return: True if number is prime or False otherwise
    """
    return number == 2 or _pollard_p1_test(number)

def get_next(number: int) -> int:
    """
    This method finds prime number following given argument

    :param number: Current number
    :return: Prime number or -1 if there is no one
    """
    lrec = lambda n: -1 if n == 10 ** 9 else n if is_prime(n) else lrec(n + 1)

    return lrec(number + 1)

def get_random_number() -> int:
    """
```

```
    Random prime number generator

    :return: Random prime number between 2 and 1 000 000 000
    """
    from random import randint

    n = randint(2, _LAST_PRIME_NUMBER)

    return n if is_prime(n) else get_next(n)

def get_random_array(length: int) -> List[int]:
    """
    Generate sorted array of random prime numbers

    :param length: The length of the array
    :return: Array of random prime numbers
    """
    return sorted(get_random_number() for _ in range(length))
```

demo.py

```
#!/usr/bin/python3
from typing import Optional
from primerandom import get_random_number, get_random_array

def read_int(min_value: int, max_value: int) -> Optional[int]:
    try:
        number = int(input())
        if min_value <= number <= max_value:
            return number
        else:
            return None
    except ValueError:
        return None

if __name__ == '__main__':
    while True:
        print('What do you want to do:')
        print('1. Get random prime number')
        print('2. Get array of prime numbers')
        print('3. Quit')

        op = read_int(1, 3)
        if op is None:
            print('Unknown command')
            continue
        if op == 1:
            print(get_random_number())
        elif op == 2:
            print(Enter the length of array:')
            while True:
                ln = read_int(1, 1000000)
                if ln is None:
                    print('Length of array must be in (0; 1000000]')
                else:
                    print('\n'.join(str(v) for v in get_random_array(ln)))
                    break
        elif op == 3:
            break
```