Министерство образования и науки Российской Федерации Федеральное агентство по образованию

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

**Знакомство с языком программирования Python**

Лабораторная работа №6

по курсу «Разработка программных систем»

Вариант 1

Выполнил студент группы ИВТб-31 /Категов А. Д./ Проверил преподаватель /Чистяков Г.А./

Киров 2024

1. Цель

Целью работы является получение навыков работы с интерпретатором языка программирования Python, а также получение навыков создания элементарных программ.

1. Задание

В соответствии с выбранным вариантом разработать набор методов на языке программирования Python. Для выполнения лабораторной работы необходимо решить следующие задачи.

* Установить на рабочую станцию интерпретатор языка Python
* Установить на рабочую станцию любую подходящую среду разработки
* Настроить окружение
* Реализовать в отдельном модуле группу методов
* Продемонстрировать работу модуля на подготовленном сценарии

Вариант 1.

Разработать модуль с набором методов для работы с простыми числами.

В модуле реализованы функции для работы с простыми числами:

* Проверку числа на простоту
* Разложение на простые множители
* Получение следующего простого числа
* Получение случайного простого числа
* Получение массива простых чисел

1. Листинг программы

Листинг программной реализации приведен в приложении А.

**Вывод**: в ходе выполнения лабораторной работы был установлен интерпретатор Python, были изучены основы языка программирования Python и получены навыки написания элементарных программ. В процессе работы был написан в модуль для работы с простыми числами. На основании всего сказанного выше можно сделать вывод о том, что цель достигнута, необходимые знания получены, задание выполнено верно, что доказывается полученными результатами.

Приложение А

(обязательное)

Листинг программы

**prime.py**

import random

MAX\_PRIME = 999999997

def is\_prime(n): # optimized trial division with 6k optimization

    if n <= 1: # numbers less than or equal to 1 are not prime

        return False

    if n <= 3:                   # 2 and 3 are prime

        return True

    if n % 2 == 0 or n % 3 == 0: # exclude even numbers and numbers divisible by 3

        return False

    i = 5 # check the divisors from 5 to sqrt(n) in increments of 6

    while i \* i <= n:

        if n % i == 0 or n % (i + 2) == 0:

            return False

        i += 6

    return True

prime\_factorization = lambda num, k = 2: [num] if is\_prime(num) else [k] + prime\_factorization(num // k, k) if num % k == 0 else prime\_factorization(num, k + 1)

next\_prime = lambda i: next\_prime(i + 1) if not is\_prime(i + 1) else i + 1 # find the next prime number

get\_random\_prime = lambda: next\_prime(random.randint(2, MAX\_PRIME - 1))

def generate\_primes(size): # generating an array of prime numbers

    primes = []

    while len(primes) < size:

        prime = get\_random\_prime()

        primes.append(prime)

    return primes

**main.py**

from prime import \*

def inputInt():

    try:

        value = int(input())

        if value < 0:

            return None

        return value

    except ValueError:

        return None

if \_\_name\_\_ == '\_\_main\_\_':

    while True:

        print()

        print("Выберите действие:")

        print("1. Генерация случайного простого числа")

        print("2. Генерация массива из простых чисел")

        print("3. Проверка числа на простоту")

        print("4. Разложение на простые множители")

        print("5. Выход")

        command = inputInt()

        if (command == None) or (5 < command < 1):

            continue

        if command == 1:

            print(get\_random\_prime())

        elif command == 2:

            while True:

                print("Введите размер массива")

                value = inputInt()

                if value == None:

                    continue

                if (value <= 0) or (value > 50):

                    print("Недопустимый размер массива")

                else:

                    result = generate\_primes(value)

                    print(result)

                break

        elif command == 3:

            while True:

                print("Введите неотрицательное целое число")

                value = inputInt()

                if value == None:

                    continue

                if value == 1 or value == 0:

                    print("Число не является ни простым ни составным")

                else:

                    print("Число простое" if is\_prime(value) else "Число составное")

                break

        elif command == 4:

            while True:

                print("Введите неотрицательное целое число")

                value = inputInt()

                if value == None:

                    continue

                if value == 1 or value == 0:

                    print("Число не является ни простым, ни составным")

                else:

                    result = prime\_factorization(value)

                    print(f"Простые множители: {\*result,}")

                break

        elif command == 5:

            break