**МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное**

**образовательное учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Лабораторная работа 4**

Условные

операторы и циклы в языке Python3

|  |
| --- |
| Выполнил студент группы ИВТ-б-о-20-1 |
| Симанский М.Ю « »\_\_\_\_\_\_\_\_\_20\_\_г. |
| Подпись студента\_\_\_\_\_\_\_\_\_\_\_ |
| Работа защищена « »\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г. |
| Проверил Воронкин Р.А. \_\_\_\_\_\_\_\_  (подпись) |

**Цель работы:** приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоить операторы языка Python версии 3 if , while , for , break и continue , позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

Ссылка на репозиторий: https://github.com/MaksimSimanskiy/lab4.git

**Задание 1**

5. С клавиатуры вводится цифра (от 1 до 4). Вывести на экран названия месяцев, соответствующих времени года с номером (считать зиму временем года 1).

**Код**

m = int(input('Введите число от 1 до 4 '))

if m == 4:

print('Сентябрь,Октябрь,Ноябрь')

elif m == 3:

print('Июнь,Июль,Август')

elif m == 2:

print('Март,Апрель,Май')

elif m == 1:

print('Январь,Декабрь,Февраль')

else:

print('Number not include in sequence 1-4')

**UML-диаграмма**

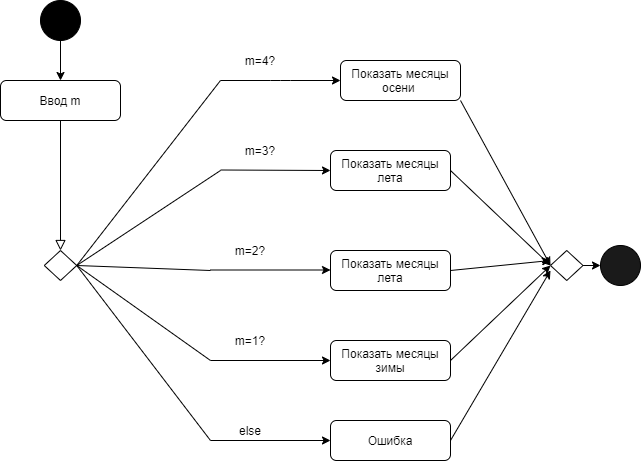


Рисунок – UML-диаграмма

**Результат**

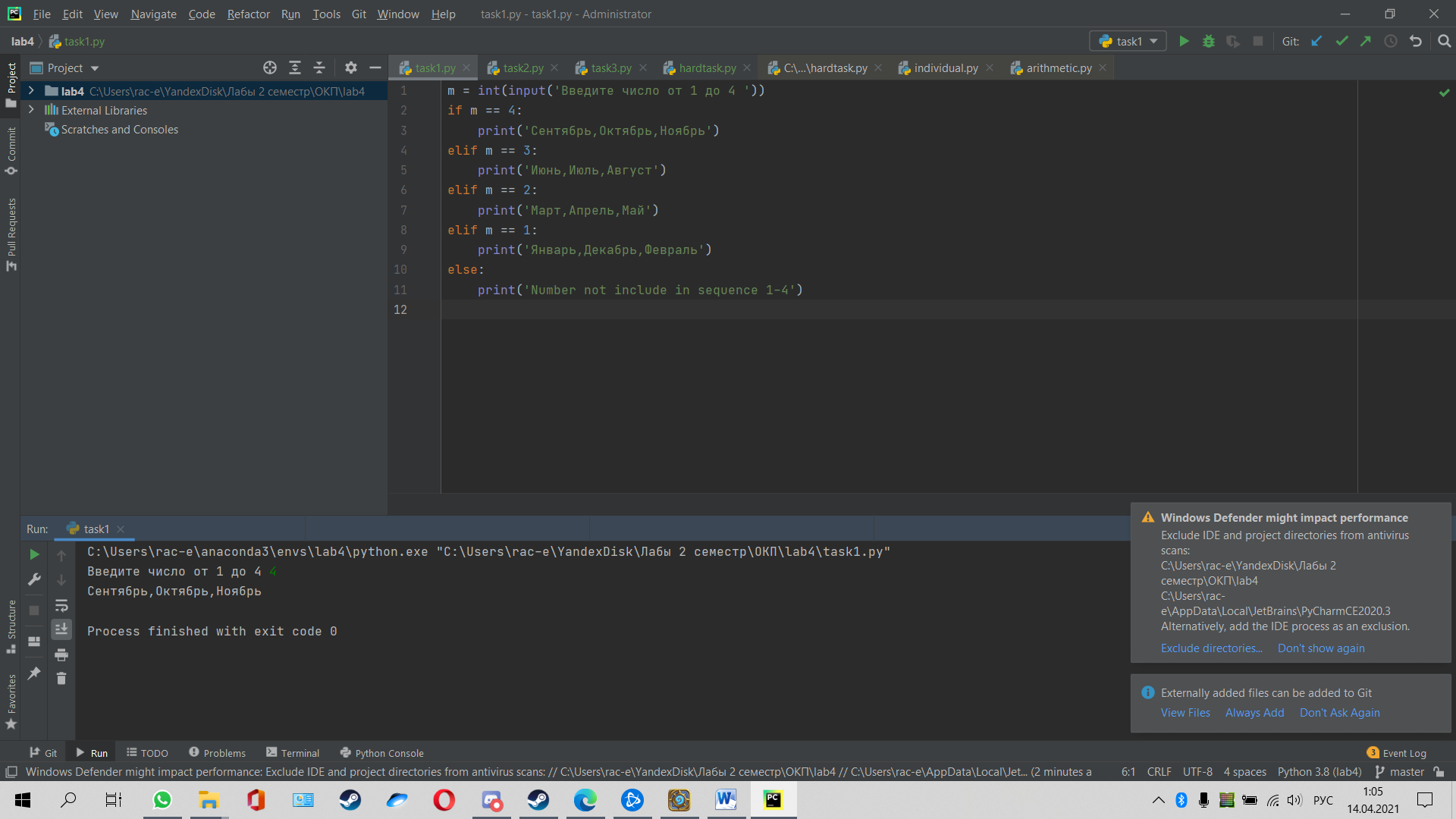


Рисунок – Результат работы

**Задание 2**

Даны произвольные действительные числа , и с . Вывести на экран сообщения:

треугольник с данными длинами сторон построить можно (указать равнобедренный, равносторонний или разносторонний получится треугольник), либо треугольник с данными длинами сторон построить нельзя.

**Код**

a = int(input('Enter first side '))

b = int(input('Enter second side '))

c = int(input('Enter third side '))

if a + b <= c or a + c <= b or b + c <= a:

print("Треугольник не существет")

elif a != b and a != c and b != c:

print("Разносторонний")

elif a == b == c:

print("Равносторонний")

else:

print("Равнобедренный")

**UML-диаграмма**



Рисунок – UML-диаграмма

**Результат**

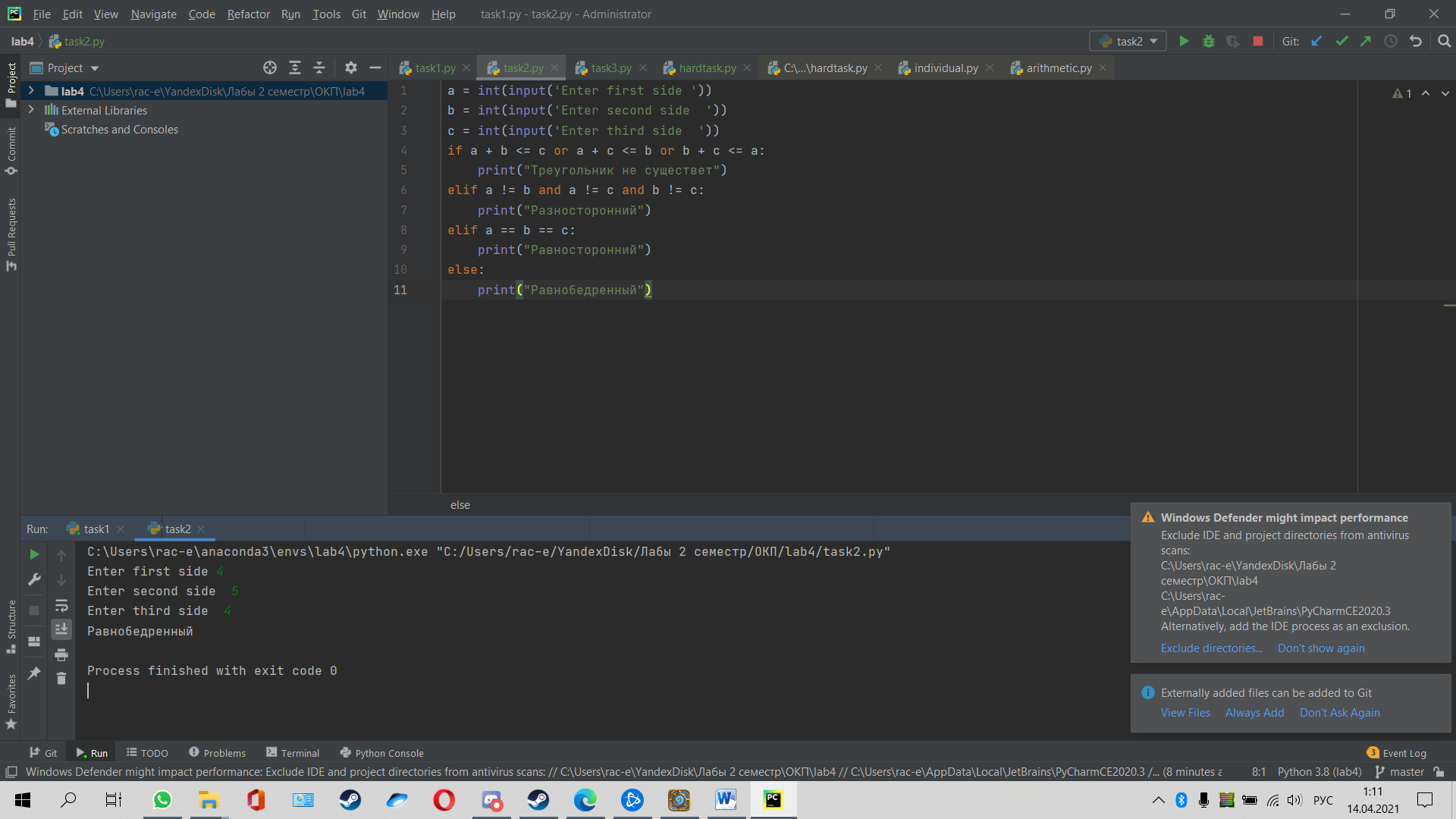


Рисунок – Результат работы

**Задание 3**

18. Составить программу, выдающую 1, если заданное число - простое и 0 - в противном случае. Число называется простым, если он делится только на 1 и на само себя. Делители числа лежат в интервале от 2 до корня из , где - заданное число.

**Код**

k = int(input("Введите число: "))

a = 0

for i in range(2, a // 2):

if a % i == 0:

print("0")

break

else:

print("1")

**UML-диаграмма**

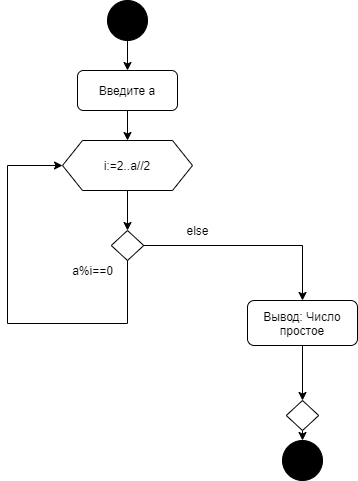


Рисунок – UML- диаграмма

**Результат**

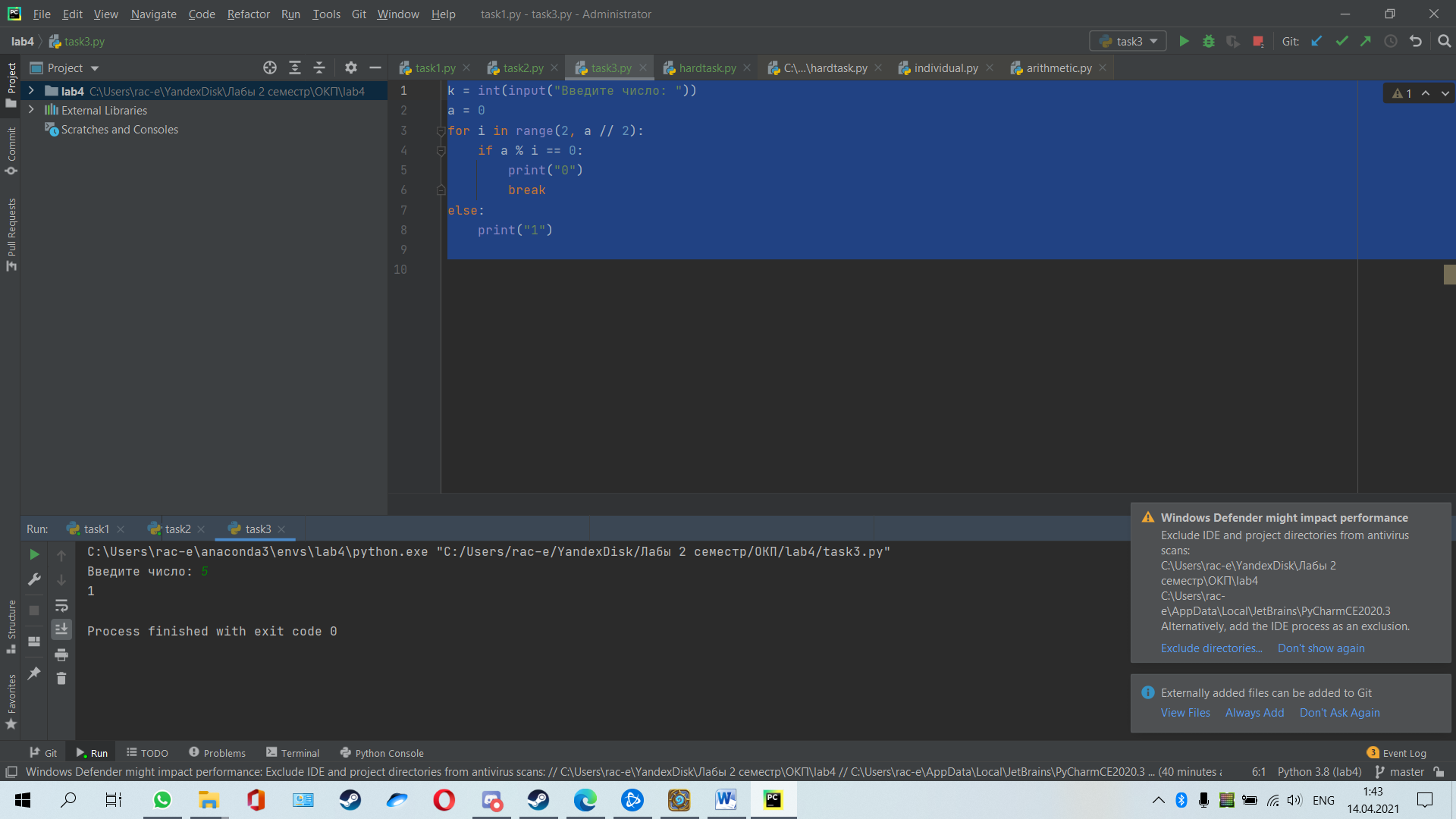
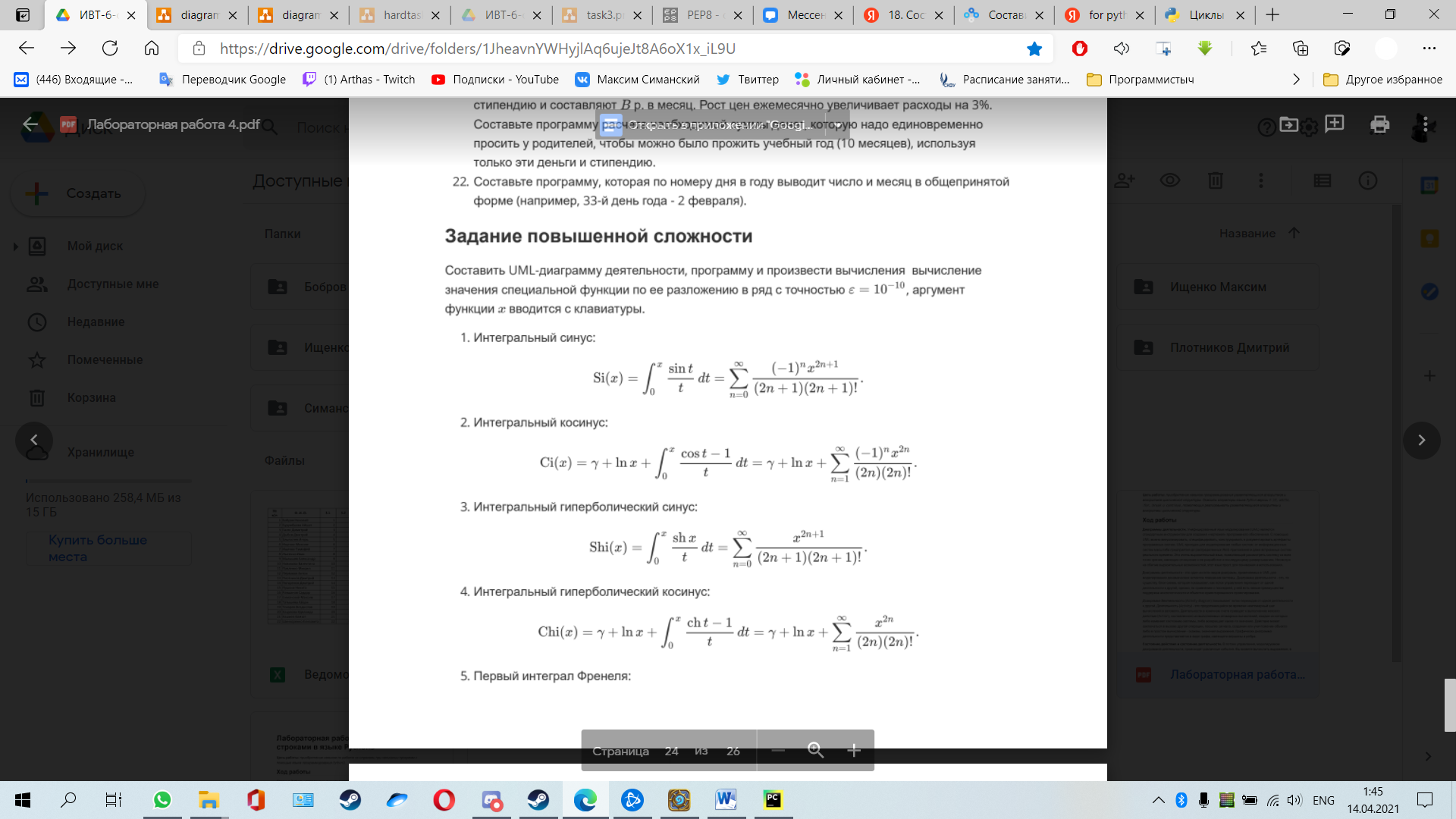


Рисунок – Результат работы

**Задание повышенной сложности**



**Код**

from decimal import Decimal

import math

EPS = 1e-10

x = float(input('Введите значение x='))

s = x

n = 0

a = 0

while math.fabs(s) > EPS:

s \*= (((-1) \*\* n) \* (x \*\* (2 \* n + 1)))/(math.factorial((2 \* n + 1) \* (2 \* n + 1)))

a += s

n += 1

print(f"Si({x}) = {a}")

**UML-диаграмма**

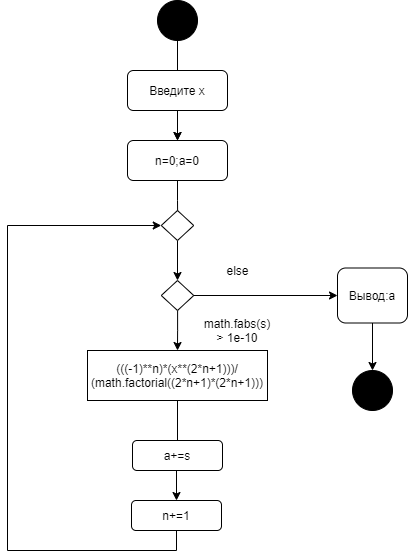


Рисунок – UML-диаграмма

**Результат**

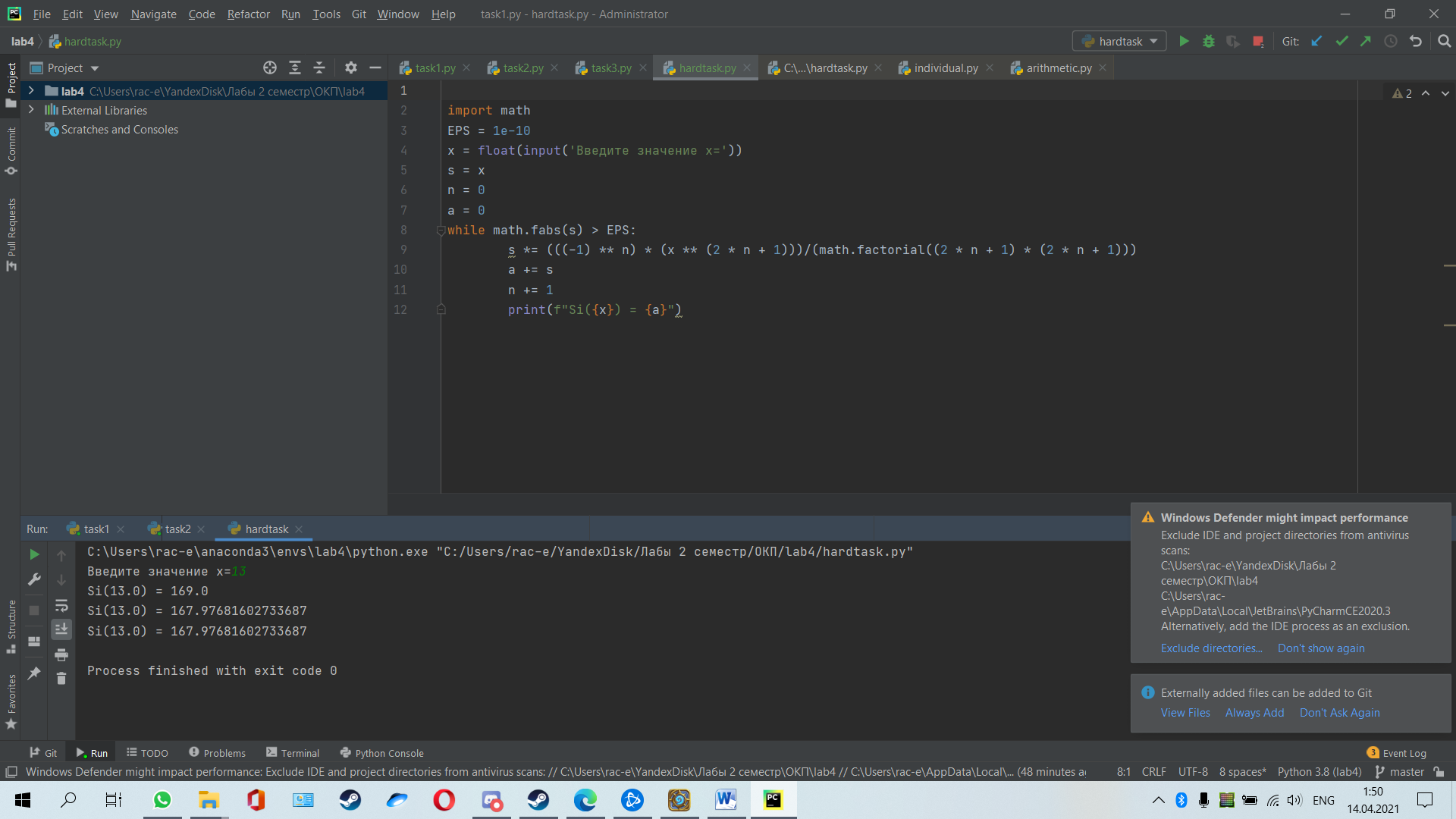


Рисунок – Результат работы

**Ответы на контрольные вопросы**

1. С помощью UML можно визуализировать, специфицировать, конструировать и документировать артефакты программных систем. UML пригоден для моделирования любых систем: от информационных систем масштаба предприятия до распределенных Web-приложений и даже встроенных систем реального времени
2. Состояние действия и состояние деятельности. В потоке управления, моделируемом диаграммой деятельности, происходят различные события. Вы можете вычислить выражение, в результате чего изменяется значение некоторого атрибута или возвращается некоторое значение.
3. Переходы. Когда действие или деятельность в некотором состоянии завершается, поток управления сразу переходит в следующее состояние действия или деятельности. Для описания этого потока используются переходы, показывающие путь из одного состояния действия или деятельности в другое. В UML переход представляется простой линией со стрелкой
4. Алгоритм разветвляющейся структуры - это алгоритм, в котором вычислительный процесс осуществляется либо по одной, либо по другой ветви, в зависимости от выполнения некоторого условия.
5. Линейный идет лишь в одном направлении , а в разветвляющемся возможны разные исходы.
6. Условные операторы:if и while.Условный оператор позволяет выполнять действия в зависимости от булева значения условия.
7. Операторы сравнения: ==,!=, <>,>, <=
8. If a>4: print(‘Hello’)
9. Для реализации выбора из нескольких альтернатив можно использовать конструкцию if – elif –else.
10. В сложных условия можно использовать операторы: &&, ||
11. Да операторы ветвления могу иметь внутри другие ветвления в виде множественного ветвления
12. Оператор for выполняет указанный набор инструкций заданное количество раз, которое определяется количеством элементов в наборе.
13. Существует 2 вида циклов: while и for
14. Функция range возвращает неизменяемую последовательность чисел в виде объекта range.

>>> range(5)

range(0, 5)

>>> list(range(5))

[0, 1, 2, 3, 4]

1. >>> list(range(15, 0, 2))
2. Python позволяет вкладывать циклы друг друга. Вложенный цикл – это цикл, который встречается внутри другого цикла.
3. Бесконечный цикл while — это [цикл](https://pythonru.com/tag/cikl-for), в котором условие никогда не становится ложным. Это значит, что тело исполняется снова и снова, а цикл никогда не заканчивается. Прервать его можно с помощью break и continue
4. Оператор break предназначен для досрочного прерывания работы цикла while.
5. Оператор continue запускает цикл заново, при этом код, расположенный после данного оператора, не выполняется.
6. В операционной системе по умолчанию присутствуют стандартных потока вывода на консоль: буферизованный поток stdout для вывода данных и информационных сообщений, а также небуферизованный поток stderr для вывода сообщений об ошибках. По умолчанию функция print использует поток stdout.
7. Для того, чтобы использовать поток stderr необходимо передать его в параметре file функции print.
8. В Python завершить программу и передать операционной системе заданный код возврата можно посредством функции exit.