

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5**  
**дисциплины «Программирование на Python»**  
**Вариант**

Выполнил:

Говоров Егор Юрьевич

2 курс, группа ИВТ-б-о-22-1,

09.03.02 «Информатика и

вычислительная техника»,

направленность (профиль)

«Программное обеспечение средств

вычислительной техники и

автоматизированных систем», очная

форма обучения

---

(подпись)

Руководитель практики:

Воронкин.Р.А, канд. технических наук,

доцент кафедры инфокоммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

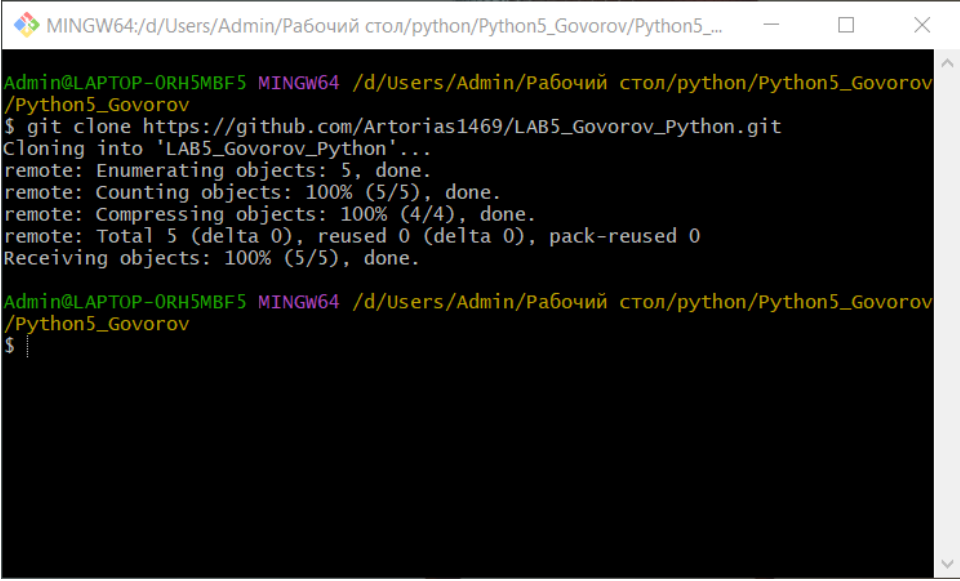
Ставрополь, 2023 г.

Тема: Условные операторы и циклы в языке Python

Цель: Приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоить операторы языка Python версии 3.x if, while, for, break и continue

Ход работы:

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.



```
Admin@LAPTOP-0RH5MBF5 MINGW64 /d/Users/Admin/Рабочий стол/python/Python5_Govorov/Python5_Govorov
$ git clone https://github.com/Artorias1469/LAB5_Govorov_Python.git
Cloning into 'LAB5_Govorov_Python'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 1. Клонирование репозитория

2. Дополнил файл .gitignore необходимыми правилами



```
*.gitignore - Блокнот
Файл Правка Формат Вид Справка
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

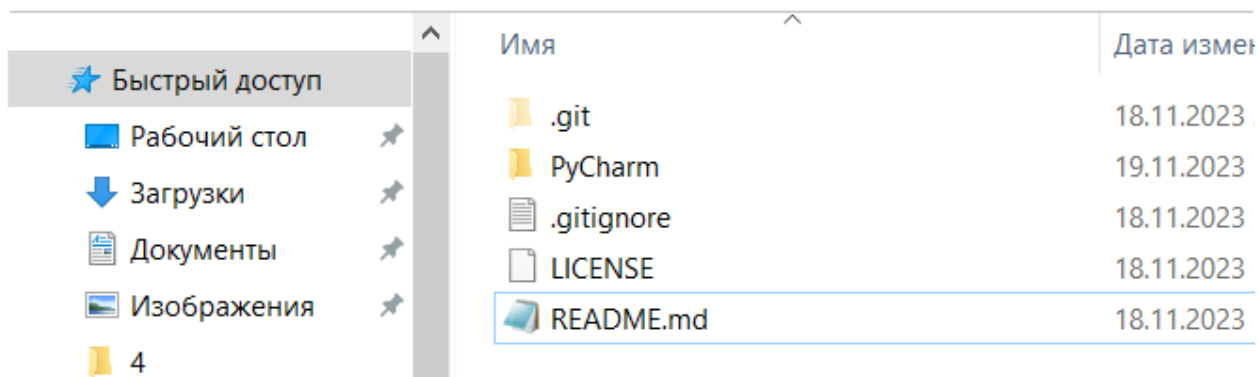
# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST

# PyInstaller
<
```

Рис 2. Внес изменения в .gitignore

### 3. Создал папку PyCharm для выполненных проектов



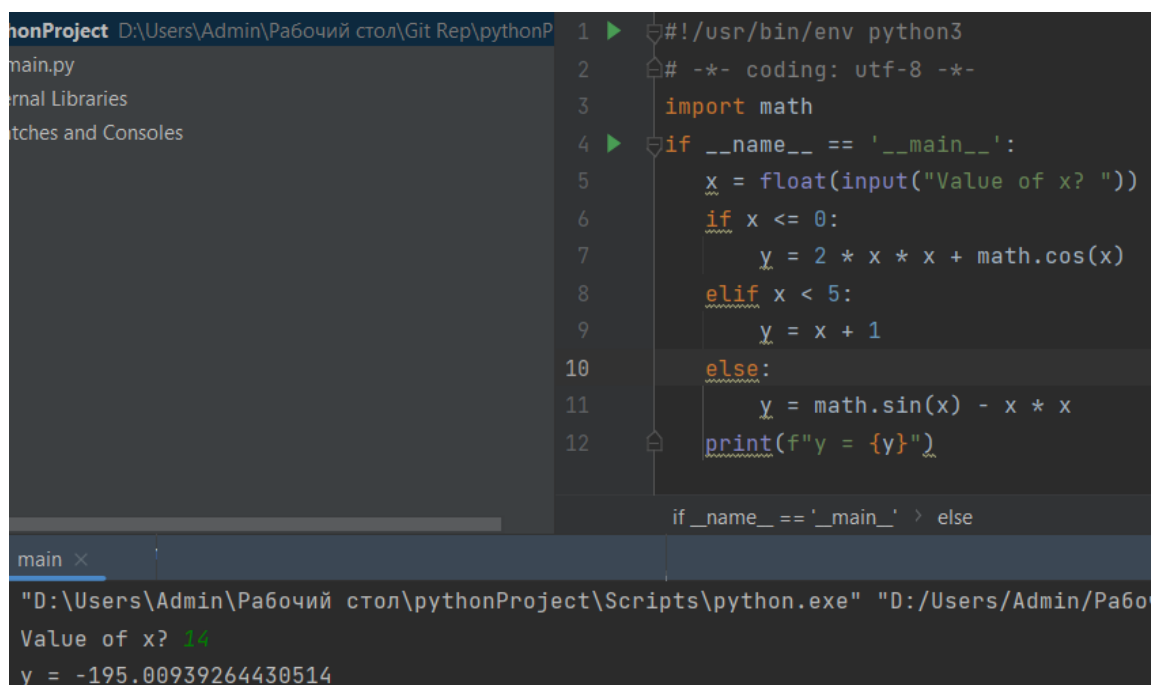
Имя	Дата измен
.git	18.11.2023
PyCharm	19.11.2023
.gitignore	18.11.2023
LICENSE	18.11.2023
README.md	18.11.2023

Рис 3. Созданная папка

### 4. Вычисление системы уравнений

$$y = \begin{cases} 2x^2 + \cos x, & x \leq 3.5, \\ x + 1, & 0 < x < 5, \\ \sin 2x - x^2, & x \geq 5. \end{cases}$$

Рис 4. Система уравнений



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import math
if __name__ == '__main__':
    x = float(input("Value of x? "))
    if x <= 0:
        y = 2 * x * x + math.cos(x)
    elif x < 5:
        y = x + 1
    else:
        y = math.sin(x) - x * x
    print(f"y = {y}")
```

main x

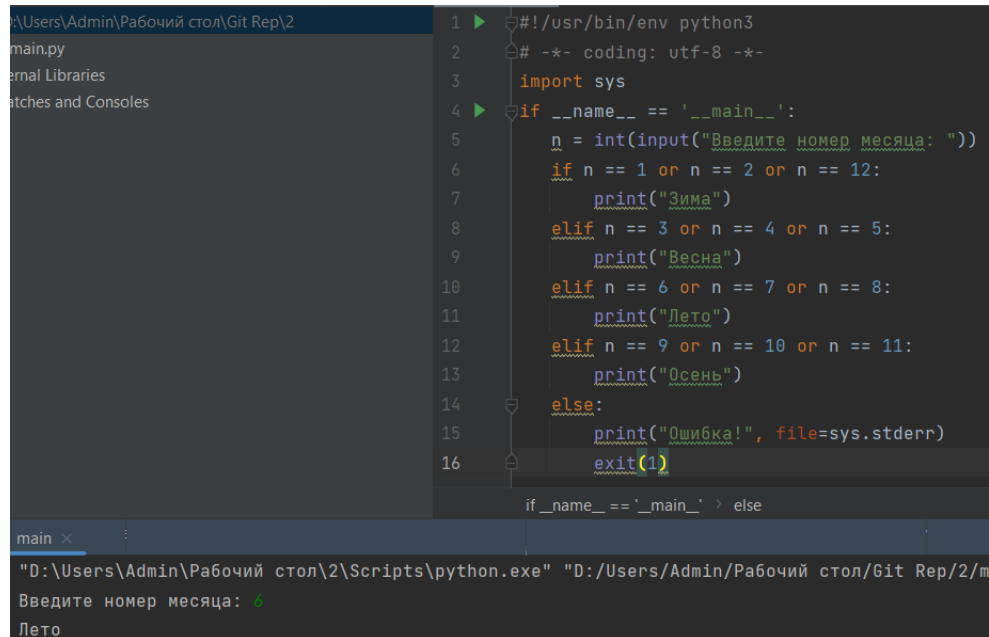
"D:\Users\Admin\Рабочий стол\pythonProject\Scripts\python.exe" "D:/Users/Admin/Рабочий стол\pythonProject\Scripts\python.exe"

Value of x? 14

y = -195.00939264430514

Рис 5. Реализация решения с помощью кода

5. С клавиатуры вводится номер месяца от 1 до 12, необходимо для этого номера месяца вывести наименование времени года.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import sys
4 if __name__ == '__main__':
5     n = int(input("Введите номер месяца: "))
6     if n == 1 or n == 2 or n == 12:
7         print("Зима")
8     elif n == 3 or n == 4 or n == 5:
9         print("Весна")
10    elif n == 6 or n == 7 or n == 8:
11        print("Лето")
12    elif n == 9 or n == 10 or n == 11:
13        print("Осень")
14    else:
15        print("Ошибка!", file=sys.stderr)
16        exit(1)
```

main x

"D:\Users\Admin\Рабочий стол\2\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/Git Rep/2/m

Введите номер месяца: 6

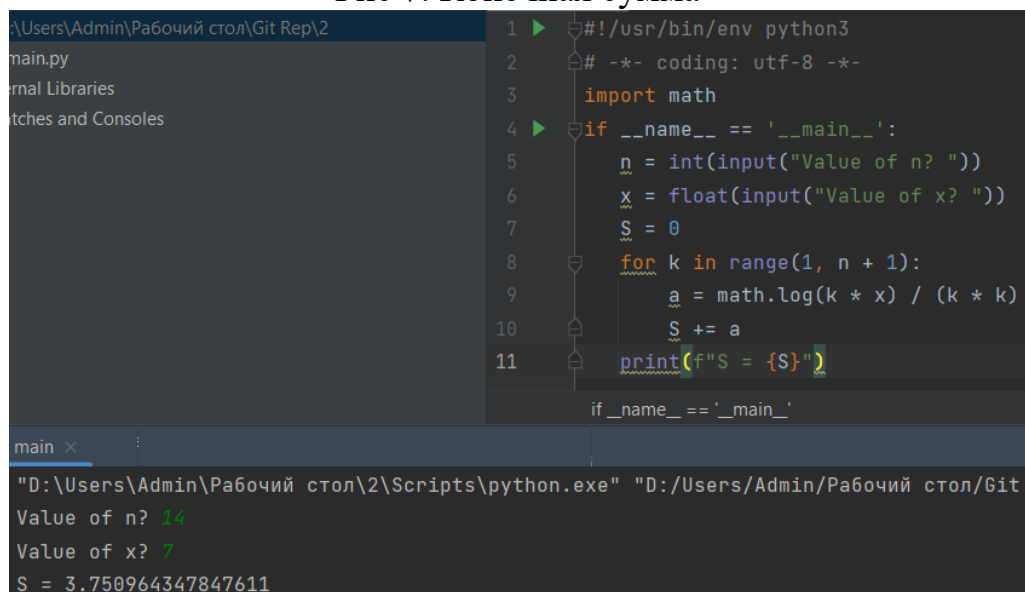
Лето

Рис 6. Реализация решения с помощью кода

6. Вычислить конечную сумму

$$S = \sum_{k=1}^n \frac{\ln kx}{k^2},$$

Рис 7. Конечная сумма



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import math
4 if __name__ == '__main__':
5     n = int(input("Value of n? "))
6     x = float(input("Value of x? "))
7     S = 0
8     for k in range(1, n + 1):
9         a = math.log(k * x) / (k * k)
10        S += a
11    print(f"S = {S}")
```

main x

"D:\Users\Admin\Рабочий стол\2\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/Git

Value of n? 14

Value of x? 7

S = 3.750964347847611

Рис 8. Реализация решения с помощью кода

7. Найти значение квадратного корня из положительного числа вводимого с клавиатуры, с некоторой заданной точностью с помощью рекуррентного соотношения. В качестве начального значения примем. Цикл должен выполняться до тех пор, пока не будет выполнено условие. Сравните со значением квадратного корня, полученным с использованием функций стандартной библиотеки.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import math
4  import sys
5  if __name__ == '__main__':
6      a = float(input("Value of a? "))
7      if a < 0:
8          print("Illegal value of a", file=sys.stderr)
9          exit(1)
10     x, eps = 1, 1e-10
11     while True:
12         xp = x
13         x = (x + a / x) / 2
14         if math.fabs(x - xp) < eps:
15             break
16     print("x = {x}\nX = {math.sqrt(a)}")

```

main < > :  
"D:\Users\Admin\Рабочий стол\4\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/Git Rep/4/main.py"  
Value of a? 10  
x = 3.4641016151377544  
X = 3.4641016151377544

Рис 9. Реализация решения с помощью кода

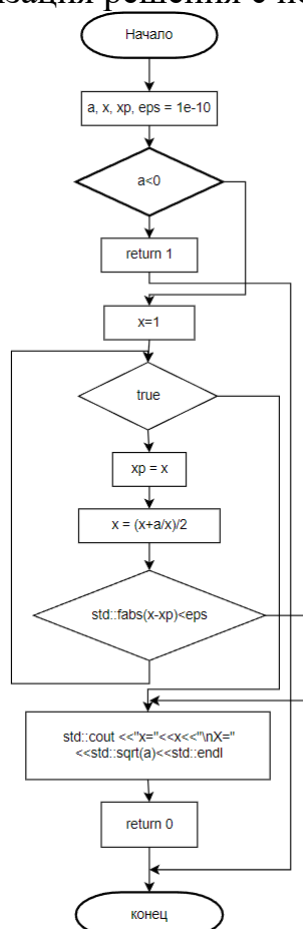


Рис 10. Блок-схема выполнения кода

## 8. Вычисление интегральной функции

$$\text{Ei}(x) = \int_{-\infty}^x \frac{\exp t}{t} dt = \gamma + \ln x + \sum_{k=1}^{\infty} \frac{x^k}{k \cdot k!},$$

Рис 11. Интегральная функция

```
"D:\Users\Admin\Рабочий стол\5\Scripts\python.exe" "D: 1 ▶ #!/usr/bin/env python3
Value of x? 56 2 # -*- coding: utf-8 -*-
Ei(56.0) = 3.804324292226219e+22 3 import math
4 import sys
5 # Постоянная Эйлера.
6 EULER = 0.5772156649015329
7 # Точность вычислений.
8 EPS = 1e-10
9 if __name__ == '__main__':
10 x = float(input("Value of x? "))
11 if x == 0:
12     print("Illegal value of x", file=sys.stderr)
13     exit(1)
14 a = x
15 S, k = a, 1
16 # Найти сумму членов ряда.
17 while math.fabs(a) > EPS:
18     a *= x * k / (k + 1) ** 2
19     S += a
20     k += 1
21 # Вывести значение функции.
22 print(f"Ei({x}) = {EULER + math.log(math.fabs(x)) + S}")
```

Рис 12. Реализация решения с помощью кода

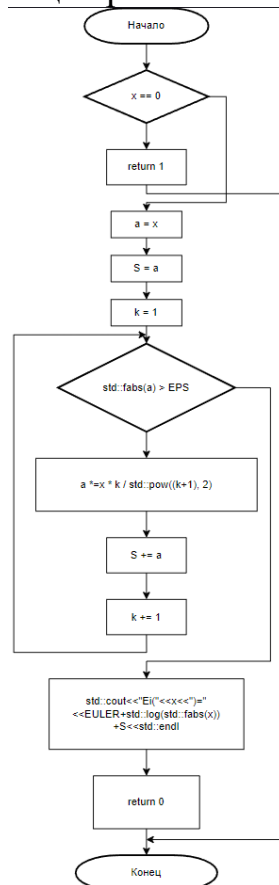


Рис 13. Блок-схема выполнения кода

9. Индивидуальное задание 1: С клавиатуры вводится цифра (от 1 до 4). Вывести на экран названия месяцев, соответствующих времени года с номером (считать зиму временем года № 1).

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def main():
5     number = int(input("Введите число от 1 до 4: "))
6
7     if number == 1:
8         print("Зима: декабрь, январь, февраль")
9     elif number == 2:
10        print("Весна: март, апрель, май")
11    elif number == 3:
12        print("Лето: июнь, июль, август")
13    elif number == 4:
14        print("Осень: сентябрь, октябрь, ноябрь")
15    else:
16        print("Некорректный ввод. Пожалуйста, введите число от 1 до 4.")
17
18 if __name__ == "__main__":
19     main()
```

main

"D:\Users\Admin\Рабочий стол\individ\_1\Scripts\python.exe" "D:\Users\Admin\Рабочий стол\Git Rep\individ\_1/main.py"

Введите число от 1 до 4: 1

Зима: декабрь, январь, февраль

Рис 14. Реализация решения с помощью кода

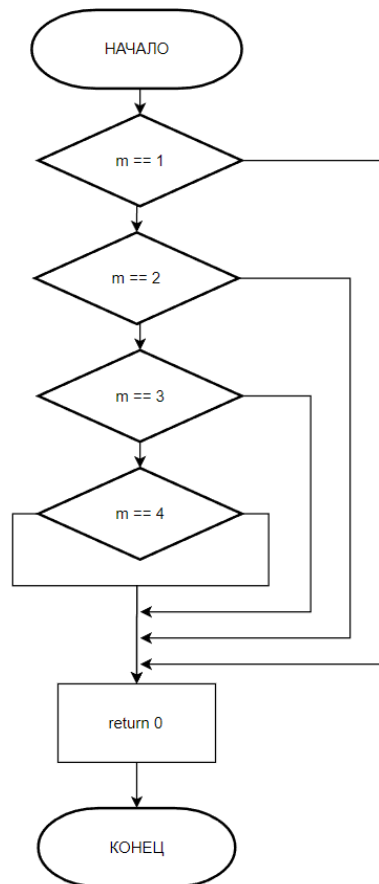
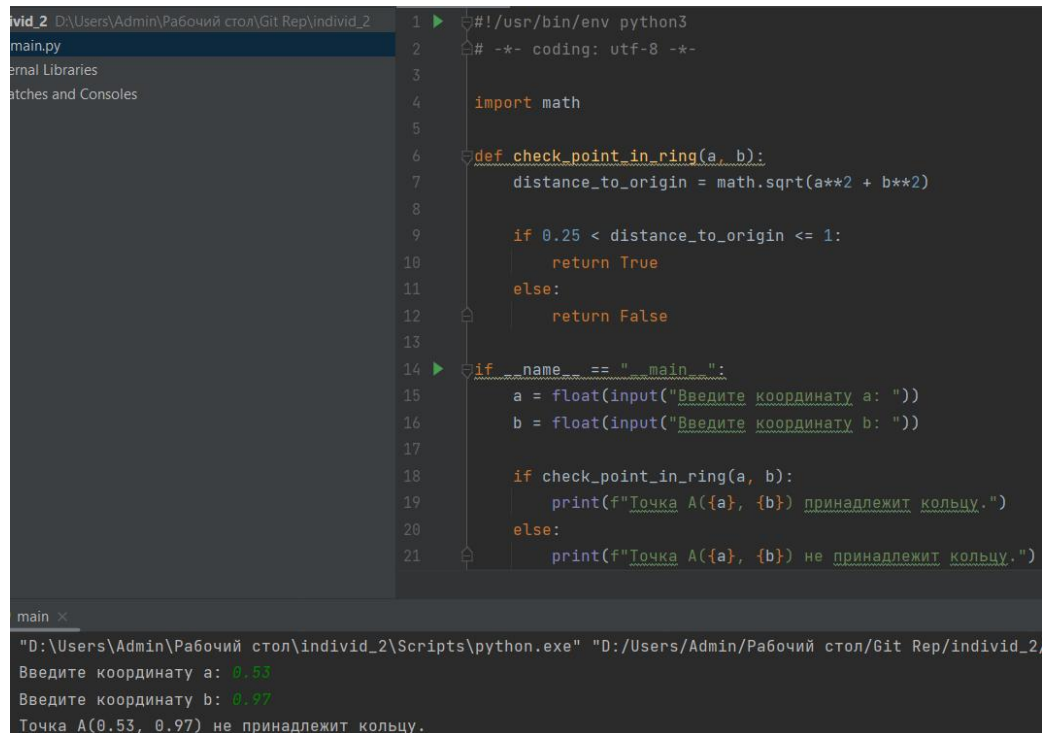


Рис 15. Блок-схема выполнения по коду

## 10. Индивидуальное задание 2:

Определить принадлежит ли точка  $A(a, b)$  кольцу определяемому окружностями  $x^2 + y^2 = 1$  и  $x^2 + y^2 = 0.25$ .



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import math
5
6 def check_point_in_ring(a, b):
7     distance_to_origin = math.sqrt(a**2 + b**2)
8
9     if 0.25 < distance_to_origin <= 1:
10         return True
11     else:
12         return False
13
14 if __name__ == "__main__":
15     a = float(input("Введите координату a: "))
16     b = float(input("Введите координату b: "))
17
18     if check_point_in_ring(a, b):
19         print(f"Точка A({a}, {b}) принадлежит кольцу.")
20     else:
21         print(f"Точка A({a}, {b}) не принадлежит кольцу.")
```

main x

"D:\Users\Admin\Рабочий стол\individ\_2\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/Git Rep/individ\_2/main.py"

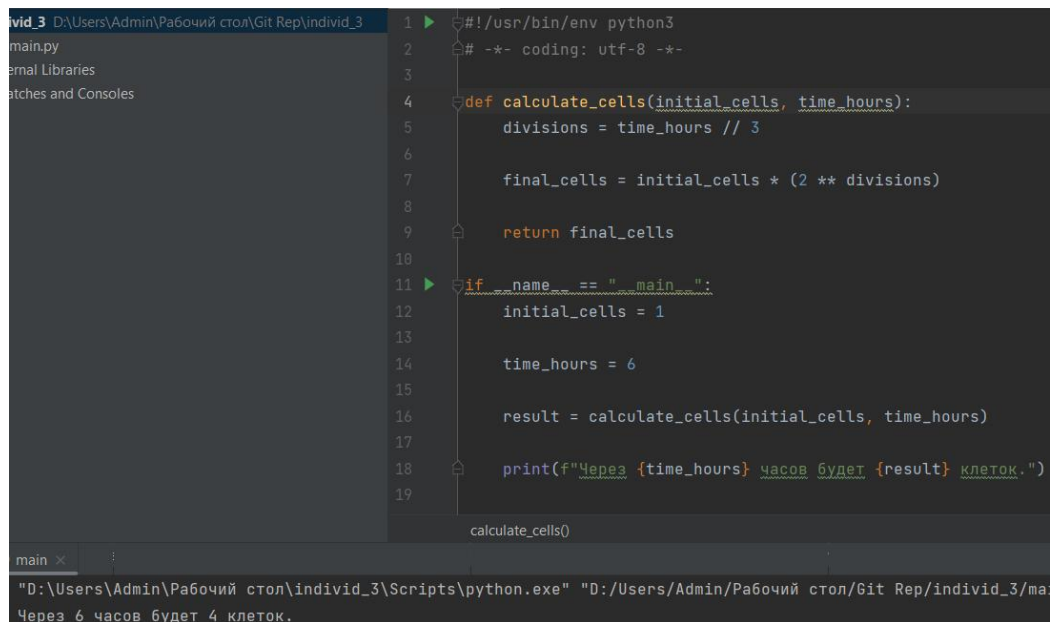
Введите координату a: 0.53

Введите координату b: 0.97

Точка A(0.53, 0.97) не принадлежит кольцу.

Рис 16. Реализация решения с помощью кода

## 11. Индивидуальное задание 3: Одноклеточная амеба каждые три часа делится на 2 клетки. Определить, сколько будет клеток через 6 часов.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def calculate_cells(initial_cells, time_hours):
5     divisions = time_hours // 3
6
7     final_cells = initial_cells * (2 ** divisions)
8
9     return final_cells
10
11 if __name__ == "__main__":
12     initial_cells = 1
13
14     time_hours = 6
15
16     result = calculate_cells(initial_cells, time_hours)
17
18     print(f"Через {time_hours} часов будет {result} клеток.")
```

calculate\_cells()

main x

"D:\Users\Admin\Рабочий стол\individ\_3\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/Git Rep/individ\_3/main.py"

Через 6 часов будет 4 клеток.

Рис 17. Реализация решения с помощью кода



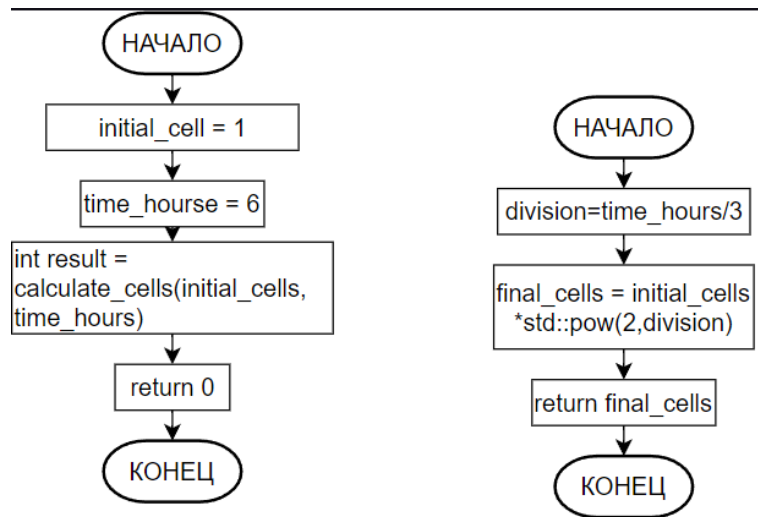


Рис 18. Блок-схема выполнения по коду

12. Выполнение усложненного индивидуального задания:

$$C(x) = \int_0^x \cos\left(\frac{\pi}{2}t^2\right) dt = \sum_{n=0}^{\infty} \frac{(-1)^n (\pi/2)^{2n}}{(2n)!(4n+1)},$$

Рис 19. Решение первого интеграла Френеля

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import math

def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

def frenel_integral(x, epsilon=1e-10):
    result = 0.0
    n = 0

    while True:
        term = ((-1) ** n) * (x ** (4 * n + 1)) / ((2 * n + 1) * factorial(2 * n + 1))
        result += term

        if abs(term) < epsilon:
            break

        n += 1

    result *= math.sqrt(2 / math.pi)

    return result

def main():
    x = float(input("Введите значение аргумента x: "))
    epsilon = 1e-10

    result = frenel_integral(x, epsilon)
    print("Значение первого интеграла Френеля при x = 6.0: 0.28940633512146695")

if __name__ == "__main__":
    main()

```

Рис 20. Реализация решения с помощью кода

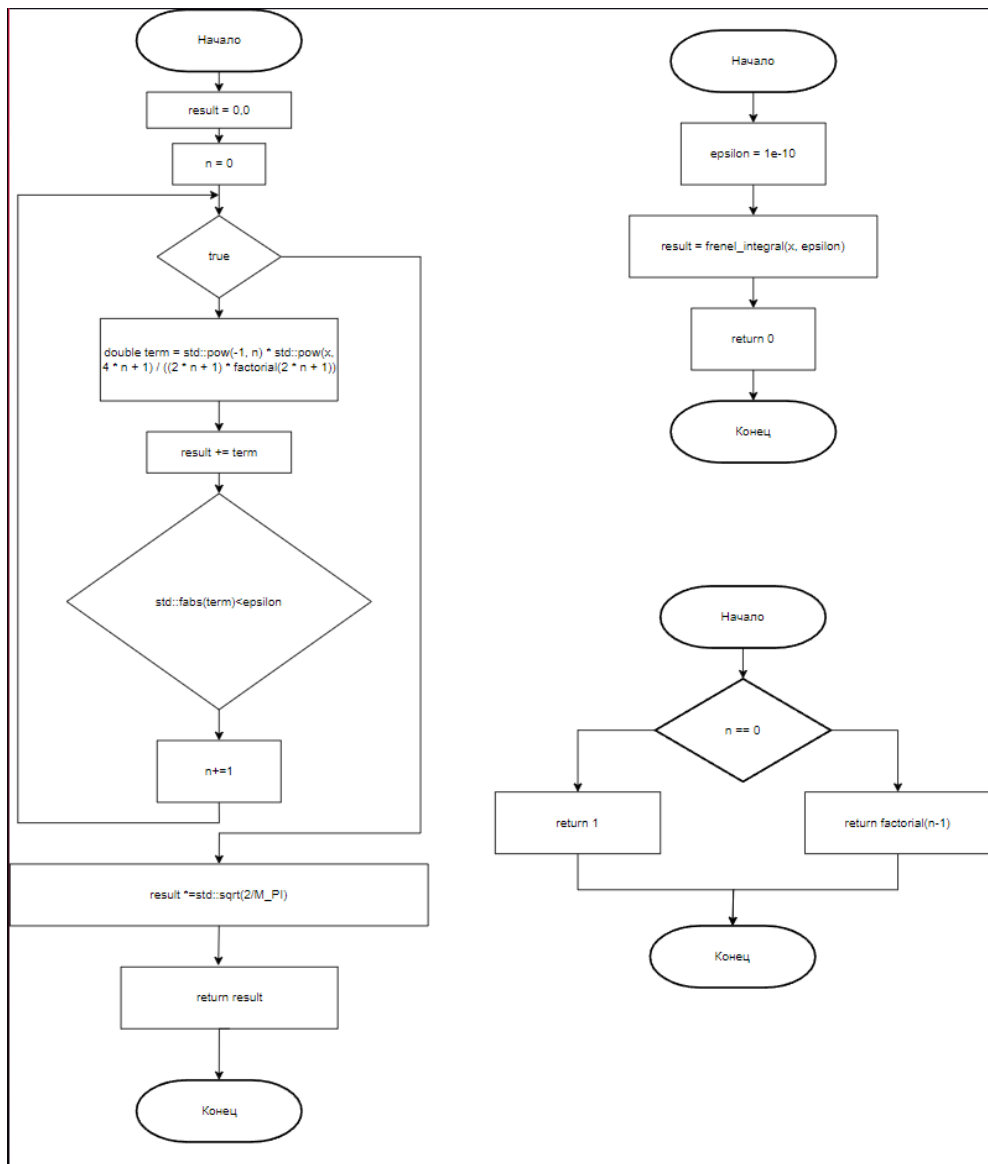


Рис 21. Блок-схема выполнения по коду