

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО  
ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ  
УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**«Исследование возможностей Git для работы с  
локальными репозиториями»**

**Отчет по лабораторной работе № 2.2  
по дисциплине «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1  
Степанов Д.А. .«14» сентября 2022г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверила Воронкин Р.А. \_\_\_\_\_  
(подпись)

№1

Отработка примеров лабораторной работы. Создание для каждого примера отдельного модуля языка Python. Для примеров 4 и 5 постройте UML-диаграмму деятельности

```
1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import math
4
5  ▶  if __name__ == '__main__':
6      x = float(input("Value of x? "))
7
8      if x <= 0:
9          y = 2 * x * x + math.cos(x)
10     elif x < 5:
11         y = x + 1
12     else:
13         y = math.sin(x) - x * x
14     print(f"y = {y}")
```

example\_1 x

C:\Users\dogim\AppData\Local\Programs\Python\Python310\python.exe

Value of x? 10

y = -100.54402111088937

Первый пример

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

example\_2 x

C:\Users\dogim\AppData\Local\Programs\Python\Python310\python.exe

Введите номер месяца: 12

Зима

Второй пример

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import math
4
5 if __name__ == '__main__':
6     n = int(input("Value of n? "))
7     x = float(input("Value of x? "))
8
9     S = 0.0
10    for k in range(1, n + 1):
11        a = math.log(k * x) / (k * k)
12        S += a
13    print(f"S = {S}")
14
15 if __name__ == '__main__':
```

example\_3 ×

C:\Users\dogim\AppData\Local\Programs\Python\Python310\python.exe  
Value of n? 10  
Value of x? 50  
S = 6.681229661432769

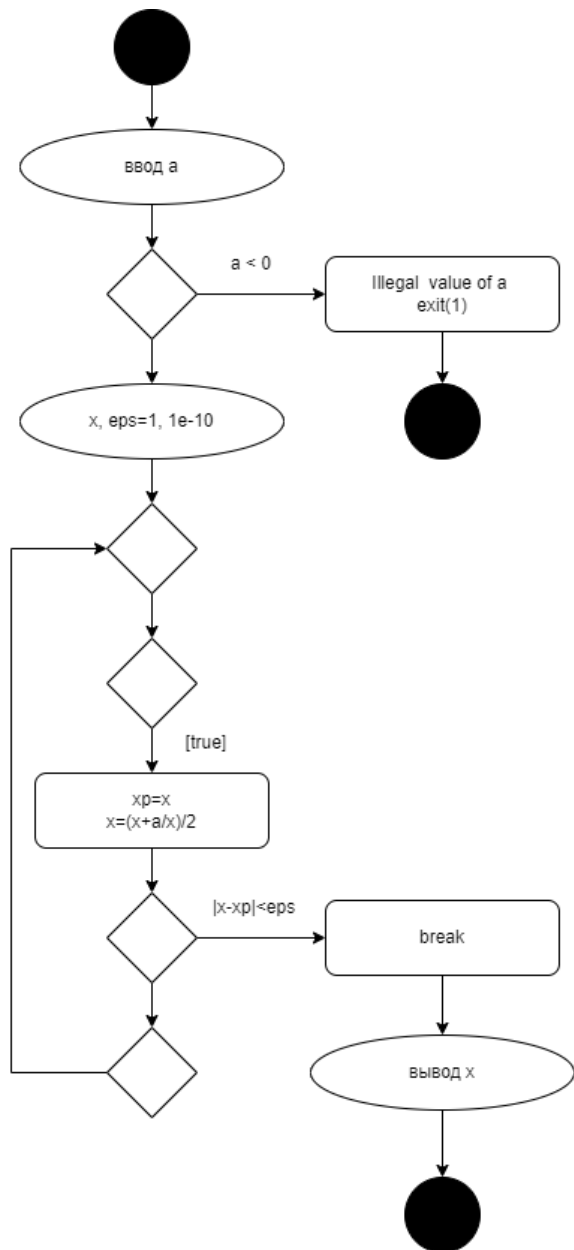
Третий пример

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import math
4 import sys
5
6 if __name__ == '__main__':
7     a = float(input("Value of a? "))
8     if a < 0:
9         print("Illegal value of a", file=sys.stderr)
10        exit(1)
11
12    x, eps = 1, 1e-10
13    while True:
14        xp = x
15        x = (x + a / x) / 2
16        if math.fabs(x - xp) < eps:
17            break
18    print(f"x = {x}\nX = {math.sqrt(a)}")
19
20 if __name__ == '__main__':
```

example\_4 ×

C:\Users\dogim\AppData\Local\Programs\Python\Python310\python.exe  
Value of a? 13  
x = 3.6055512754639896  
X = 3.605551275463989

Четвертый пример



```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math
5  import sys
6
7  # Постоянная Эйлера.
8  EULER = 0.5772156649015328606
9  # Точность вычислений.
10 EPS = 1e-10
11
12 if __name__ == '__main__':
13     x = float(input("Value of x? "))
14     if x == 0:
15         print("Illegal value of x", file=sys.stderr)
16         exit(1)
17
18     a = x
19     S, k = a, 1
20
21     # Найти сумму членов ряда.
22     while math.fabs(a) > EPS:
23         a *= x * k / (k + 1) ** 2
24         S += a
25         k += 1
26
27     # Вывести значение функции.
28     print(f"Ei({x}) = {EULER + math.log(math.fabs(x)) + S}")
29
30 if __name__ == '__main__':

```

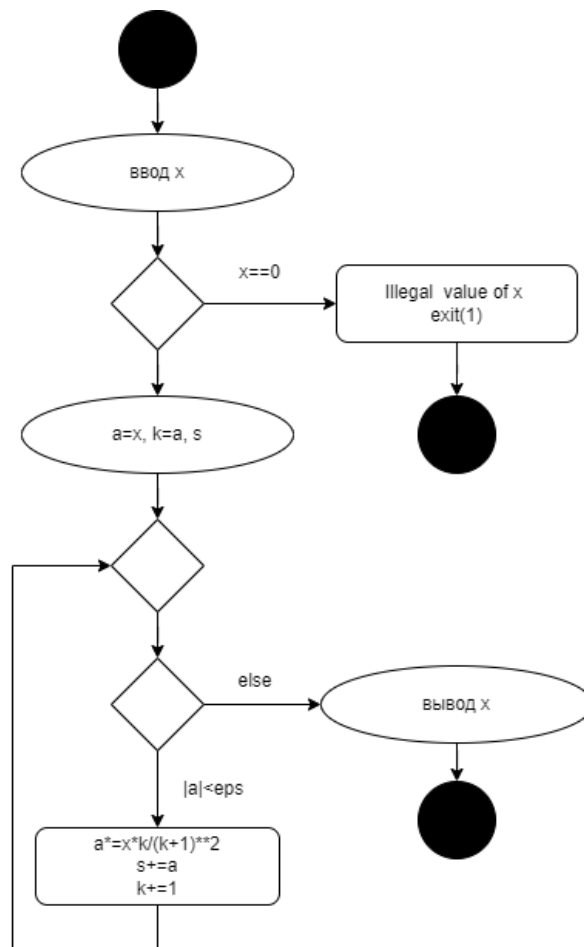
example\_5 x

C:\Users\dogim\AppData\Local\Programs\Python\Python310\python.exe C:\0PI\

Value of x? 21

Ei(21.0) = 66127186.355484925

### Пятый пример

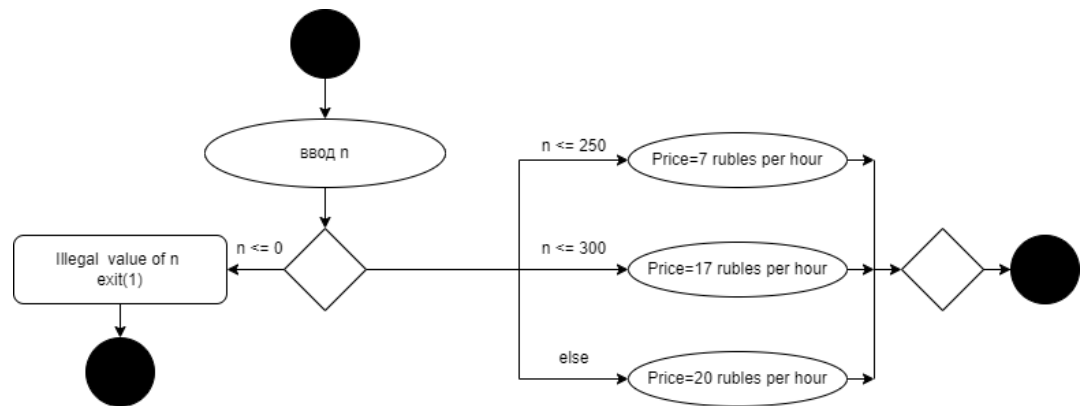


## Задание 1 (11 вариант)

11. Компания по снабжению электроэнергией взимает плату с клиентов по тарифу:

- 7 р. за 1 кВт/ч за первые 250 кВт/ч;
- 17 р. за кВт/ч, если потребление свыше 250, но не превышает 300 кВт/ч;
- 20 р. за кВт/ч, если потребление свыше 300 кВт/ч.

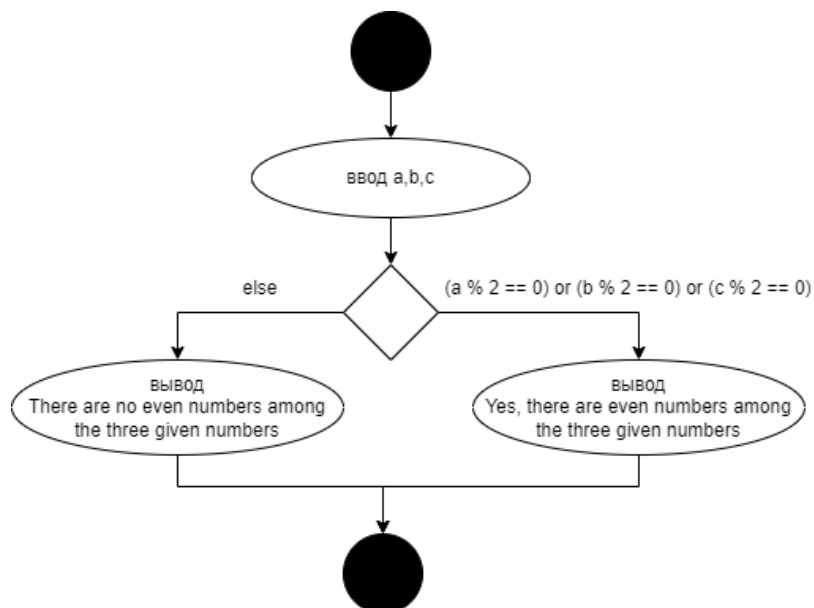
Потребитель израсходовал  $n$  кВт/ч. Подсчитать плату.



```
Value of n: 15
Price = 7 rubles per hour
```

## Задание 2 (11 вариант)

11. Определить, есть ли среди трёх заданных чисел чётные.



```
Enter 3 numbers: 2 1 3
Yes, there are even numbers among the three given numbers
```

### Задание 3 (11 вариант)

Составьте программу, которая печатает таблицу умножения натуральных чисел в десятичной системе счисления.

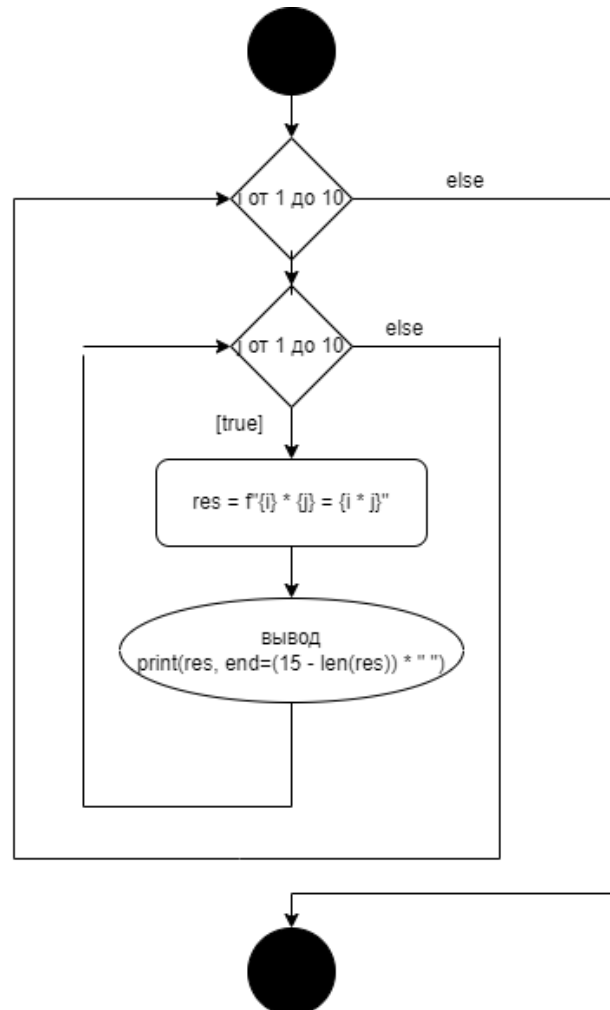
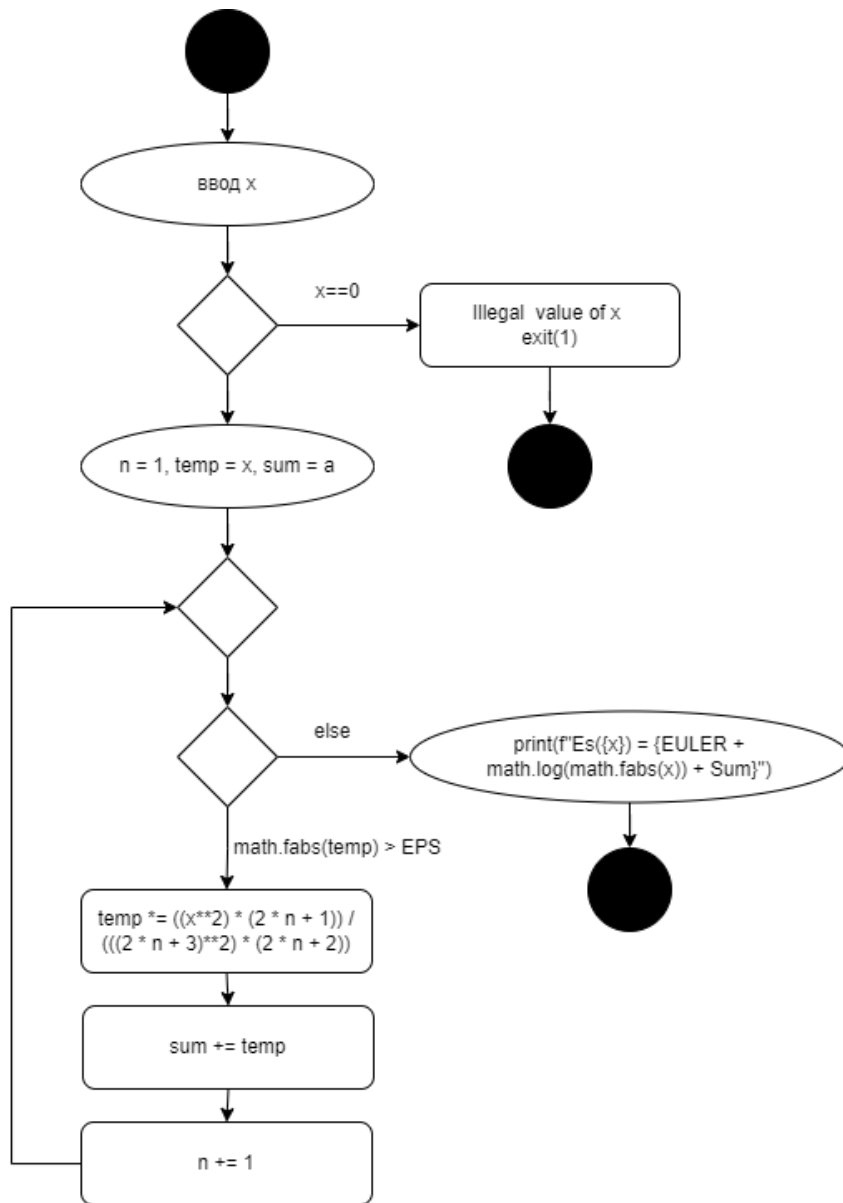


Таблица умножения:								
1 * 1 = 1	1 * 2 = 2	1 * 3 = 3	1 * 4 = 4	1 * 5 = 5	1 * 6 = 6	1 * 7 = 7	1 * 8 = 8	1 * 9 = 9
2 * 1 = 2	2 * 2 = 4	2 * 3 = 6	2 * 4 = 8	2 * 5 = 10	2 * 6 = 12	2 * 7 = 14	2 * 8 = 16	2 * 9 = 18
3 * 1 = 3	3 * 2 = 6	3 * 3 = 9	3 * 4 = 12	3 * 5 = 15	3 * 6 = 18	3 * 7 = 21	3 * 8 = 24	3 * 9 = 27
4 * 1 = 4	4 * 2 = 8	4 * 3 = 12	4 * 4 = 16	4 * 5 = 20	4 * 6 = 24	4 * 7 = 28	4 * 8 = 32	4 * 9 = 36
5 * 1 = 5	5 * 2 = 10	5 * 3 = 15	5 * 4 = 20	5 * 5 = 25	5 * 6 = 30	5 * 7 = 35	5 * 8 = 40	5 * 9 = 45
6 * 1 = 6	6 * 2 = 12	6 * 3 = 18	6 * 4 = 24	6 * 5 = 30	6 * 6 = 36	6 * 7 = 42	6 * 8 = 48	6 * 9 = 54
7 * 1 = 7	7 * 2 = 14	7 * 3 = 21	7 * 4 = 28	7 * 5 = 35	7 * 6 = 42	7 * 7 = 49	7 * 8 = 56	7 * 9 = 63
8 * 1 = 8	8 * 2 = 16	8 * 3 = 24	8 * 4 = 32	8 * 5 = 40	8 * 6 = 48	8 * 7 = 56	8 * 8 = 64	8 * 9 = 72
9 * 1 = 9	9 * 2 = 18	9 * 3 = 27	9 * 4 = 36	9 * 5 = 45	9 * 6 = 54	9 * 7 = 63	9 * 8 = 72	9 * 9 = 81

### Задача повышенной сложности

$$\text{Shi}(x) = \int_0^x \frac{\text{sh } t}{t} dt = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)(2n+1)!},$$



Value of x? 10

Es(10.0) = 225.38040899379104

Вопросы для защиты работы



1. Для чего нужны диаграммы деятельности UML?

Диаграммы деятельности представляют собой графическое представление рабочих процессов поэтапных действий и действий с поддержкой выбора, итерации и параллелизма. Они описывают поток управления целевой системой, такой как исследование сложных бизнес-правил и операций, а также описание прецедентов и бизнес-процессов. В UML диаграммы деятельности предназначены для моделирования как вычислительных, так и организационных процессов.

2. Что такое состояние действия и состояние деятельности?

Состояние действия — это состояние, внутренняя активность которого является действием.

Состояние деятельности — это состояние, внутренняя активность которого является деятельностью.

3. Какие нотации существуют для обозначения переходов и ветвлений в диаграммах деятельности?

Переходы, ветвление, алгоритм разветвляющейся структуры, алгоритм циклической структуры.

4. Какой алгоритм является алгоритмом разветвляющейся структуры?

Алгоритм разветвляющейся структуры - это алгоритм, в котором вычислительный процесс осуществляется либо по одной, либо по другой ветви, в зависимости от выполнения некоторого условия.

5. Чем отличается разветвляющийся алгоритм от линейного?

Линейный алгоритм - алгоритм, все этапы которого выполняются однократно и строго последовательно.

Разветвляющийся алгоритм - алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из нескольких возможных шагов.

6. Что такое условный оператор? Какие существуют его формы?

Оператор, конструкция языка программирования, обеспечивающая выполнение определённой команды (набора команд) только при условии

истинности некоторого логического выражения, либо выполнение одной из нескольких команд.

Условный оператор имеет полную и краткую формы.

7. Какие операторы сравнения используются в Python?

If, elif, else

8. Что называется простым условием? Приведите примеры.

Простым условием называется выражение, составленное из двух арифметических выражений или двух текстовых величин.

Пример: `x == y`

9. Что такое составное условие? Приведите примеры.

Составное условие – логическое выражение, содержащее несколько простых условий объединенных логическими операциями. Это операции `not`, `and`, `or`.

Пример: `(a == b or c == d)`

10. Какие логические операторы допускаются при составлении сложных условий?

`not`, `and`, `or`.

11. Может ли оператор ветвления содержать внутри себя другие ветвления?

Может.

12. Какой алгоритм является алгоритмом циклической структуры?

Циклический алгоритм — алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными.

13. Типы циклов в языке Python.

В Python есть 2 типа циклов: - цикл `while`, - цикл `for`.

14. Назовите назначение и способы применения функции `range`.

Функция `range` генерирует серию целых чисел, от значения `start` до `stop`, указанного пользователем. Мы можем использовать его для цикла `for` и обходить весь диапазон как список.

Функция `range` возвращает неизменяемую последовательность чисел в виде объекта `range`.

Параметры функции:

`start` - с какого числа начинается последовательность. По умолчанию - 0

`stop` - до какого числа продолжается последовательность чисел.

Указанное число не включается в диапазон

`step` - с каким шагом растут числа. По умолчанию 1

15. Как с помощью функции `range` организовать перебор значений от 15 до 0 с шагом 2?

`range(15, 0, -2)`

16. Могут ли быть циклы вложенными?

Могут.

17. Как образуется бесконечный цикл и как выйти из него?

Бесконечный цикл в программировании — цикл, написанный таким образом, что условие выхода из него никогда не выполняется.

18. Для чего нужен оператор `break`?

Оператор `break` предназначен для досрочного прерывания работы цикла.

19. Где употребляется оператор `continue` и для чего он используется?

Оператор `continue` используется только в циклах. В операторах `for`, `while` оператор `continue` запускает цикл заново, при этом код, расположенный после данного оператора, не выполняется.

20. Для чего нужны стандартные потоки `stdout` и `stderr`?

Ввод и вывод распределяется между тремя стандартными потоками:  
`stdin` – стандартный ввод (клавиатура), `stdout` – стандартный вывод (экран),  
`stderr` – стандартная ошибка (вывод ошибок на экран)

21. Как в Python организовать вывод в стандартный поток `stderr`?

Указать в `print(..., file=sys.stderr)`.

22. Каково назначение функции `exit`?

Функция `exit()` модуля `sys` - выход из Python.