

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет о лабораторной работе №5 по дисциплине
«Основы программной инженерии»

Выполнил:
Прокопов Дмитрий Владиславович,
2 курс, группа ПИЖ-б-о-21-1,
Проверил: Доцент
кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2022 г

Ход работы

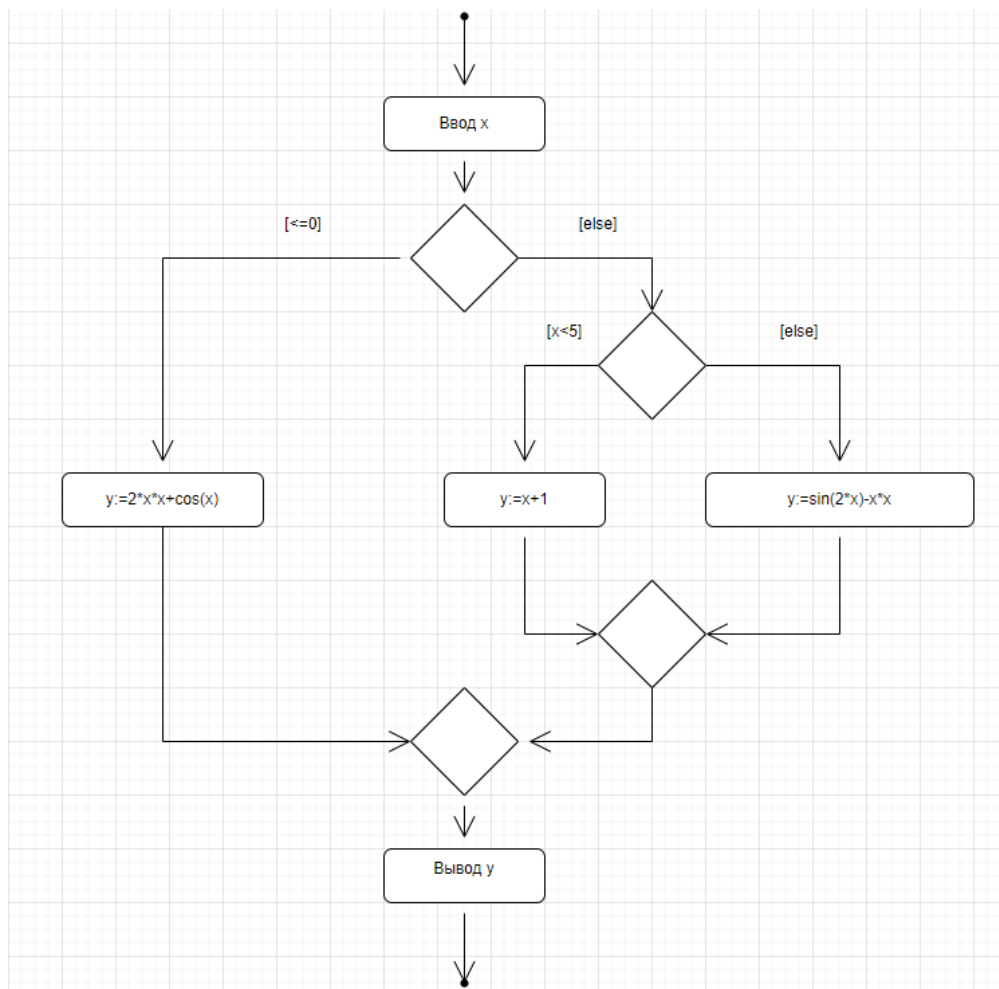
Пример №1

```
#!/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}")
```

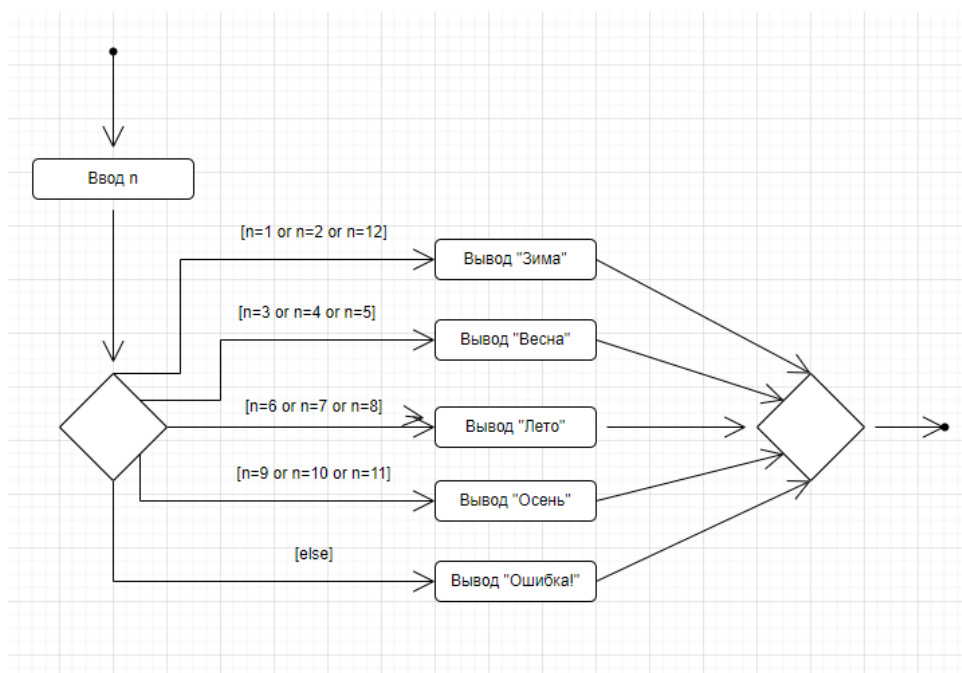
```
C:\Users\dmidt\PycharmProjects\lb5\venv\Scripts\python.exe
value of x? 7
y=-48.34301340128121
```



Пример №2

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

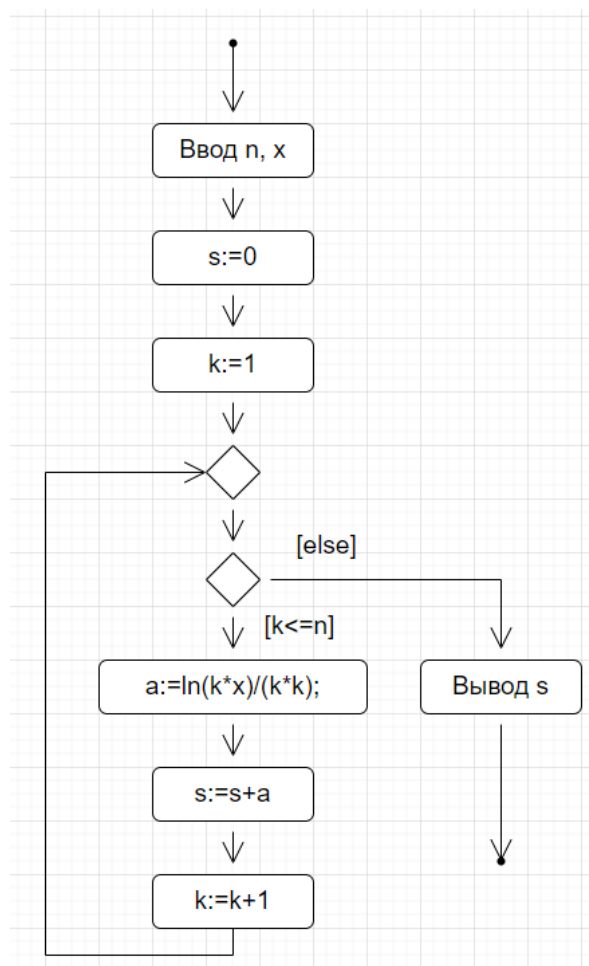
C:\Users\dmidt\PycharmProjects\lb5\env\Scripts\python.exe
 Введите номер месяца: 12
 Зима



Пример №3

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-
import math
if __name__ == '__main__':
    n=int(input("Value of n? "))
    x=float(input("Value of x? "))
    s=0.0
    for k in range(1, n+1):
        a=math.log(k*x)/(k*k)
        s +=a
    print(f"s={s}")
```

```
C:\Users\dmidt\PycharmProjects\lb5\venv\Scripts\python.exe
Value of n? 3
Value of x? 5.3
s=2.565289110751602
```



Пример №4

```

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

import math
import sys

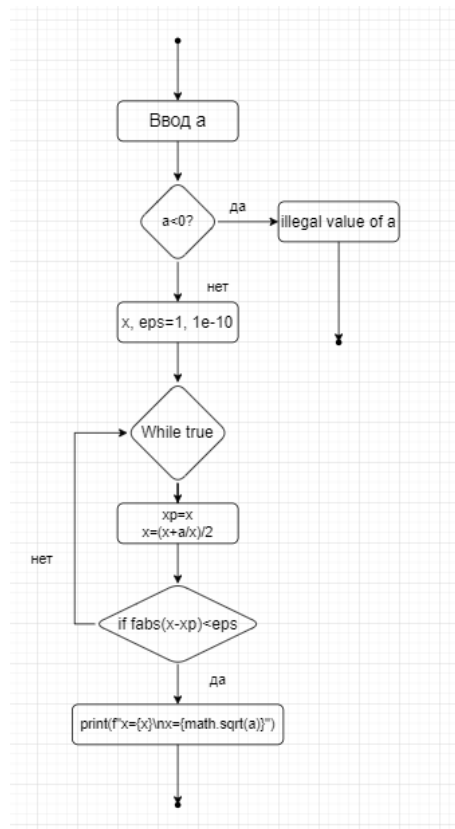
if __name__ == '__main__':
    a=float(input("Value of a? "))
    if a<0:
        print("Illegal value of a", file=sys.stderr)
        exit(1)
    x, eps=1, 1e-10
    while True:
        xp=x
        x=(x+a/x)/2
        if math.fabs(x-xp)<eps:
            break
    print(f"x={x}\nx={math.sqrt(a)}")

```

```

C:\Users\dmidt\PycharmProjects\lb5\venv\Scripts\python.exe
Value of a? 3
x=1.7320508075688772
x=1.7320508075688772

```



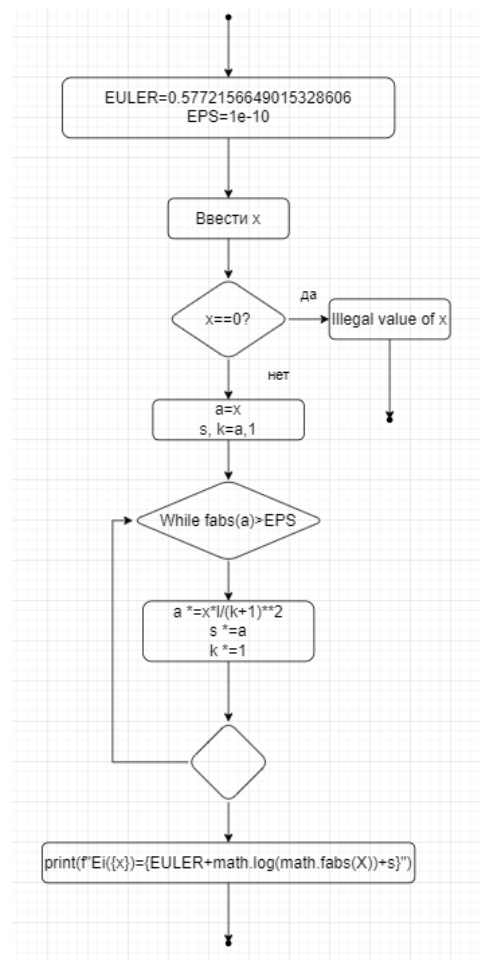
```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import math
import sys

EULER=0.5772156649015328606
EPS=1e-10

if __name__ == '__main__':
    x=float(input("Value of x? "))
    if x==0:
        print("Illegal value of x", file=sys.stderr)
        exit(1)
    a=x
    s, k=a, 1
    while math.fabs(a)>EPS:
        a *=x*k/(k+1)**2
        s +=a
        k +=1
    print(f"Ei({x}) = {EULER+math.log(math.fabs(x))+s}")
```

```
C:\Users\dmidt\PycharmProjects\lb5\venv\Scripts\python.exe
Value of x? 3
Ei(3.0) = 13.675827953281642
```



Индивидуальное задание №1

```

1 ls1=('один', 'два', 'три', 'четыре', 'пять', 'шесть', 'семь', 'восемь')
2 ls2=('минус один', 'минус два', 'минус три', 'минус четыре', 'минус пять', 'минус шесть', 'минус семь', 'минус восемь')
3 c=int(input("Введите с "))
4 if c==0:
5     print("ноль")
6 elif c>1:
7     print(ls1[c-1])
8 else:
9     print(ls2[c+1])

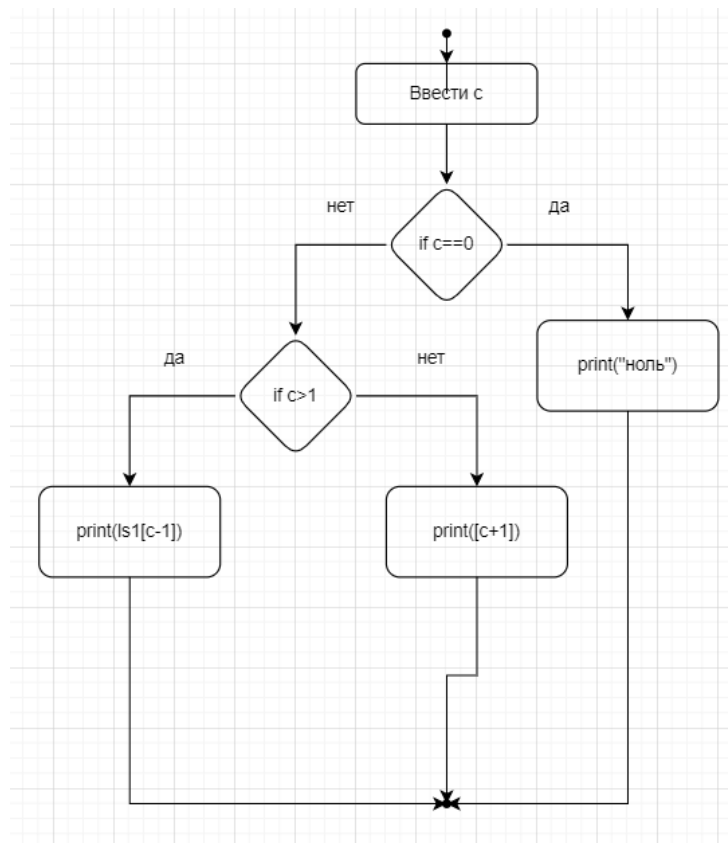
```

individual1

C:\Users\dmidt\PycharmProjects\lb5\venv\Scripts\python.exe C:/Users/dmidt/PycharmProjects/lb5/individual1.py

Введите с -5

минус пять



Индивидуальное задание №2

```

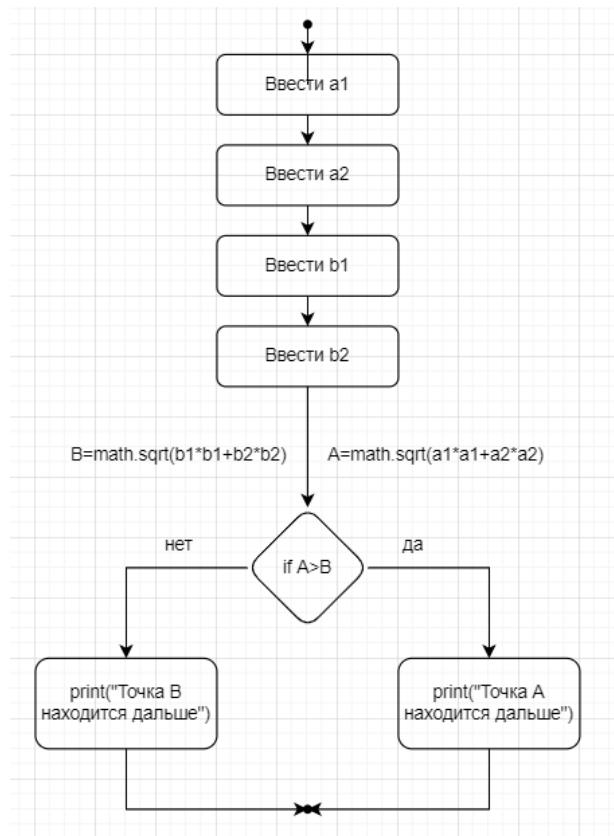
1  #!/usr/bin/env python3
2  #-*- coding: utf-8 -*-
3  import math
4  a1=int(input("a1= "))
5  a2=int(input("a2= "))
6  b1=int(input("b1= "))
7  b2=int(input("b2= "))
8  A=math.sqrt(a1*a1+a2*a2)
9  B=math.sqrt(b1*b1+b2*b2)
10 if A>B:
11     print("Точка A находится дальше")
12 else:
13     print("Точка B находится дальше")
14
else

```

individual2 x

C:\Users\dmidt\PycharmProjects\lb5\venv\Scripts\

a1= 3
a2= 4
b1= 2
b2= 5
Точка B находится дальше



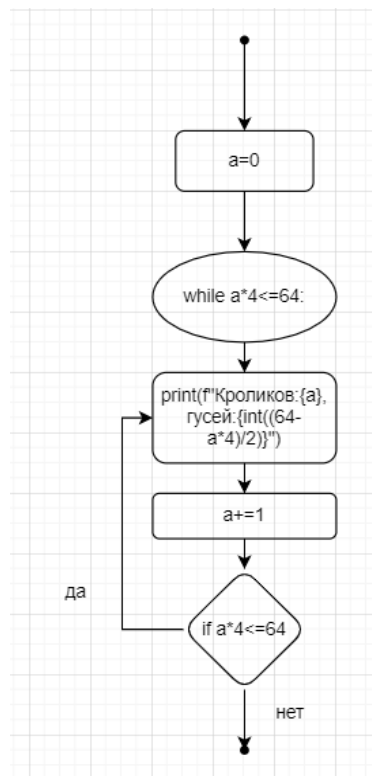
Индивидуальное задание №3

```

User 1  a=0
y lit 2  while a*4<=64:
vidu 3      print(f"Кроликов:{a}, гусей:{int((64-a*4)/2)}")
vidu 4      a+=1
vidu
мер
мер
мер
мер
мер
мер
Lib
es a

while a*4<=64

individual3 x
C:\Users\dmidt\PycharmProjects\lb5\venv\Scripts\python.exe C:/Us
Кроликов:0, гусей:32
Кроликов:1, гусей:30
Кроликов:2, гусей:28
Кроликов:3, гусей:26
Кроликов:4, гусей:24
  
```



Ответы на вопросы

1. С помощью UML можно визуализировать, специфицировать, конструировать и документировать артефакты программных систем.
2. Состояния деятельности – атомарные вычисления, которые не могут быть подвержены декомпозиции. Состояние деятельности - составное состояние, поток управления которого включает только другие состояния деятельности и действий.
3. Линии 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. В линейном алгоритме есть только один исход выполнения программы, когда в разветвляющемся результате выполнения программы зависит от того, по каким веткам пройдет ход выполнения программы.

6. Условный оператор проверяет его выражение на истинность и в зависимости от этого выполняется то или иное условие. Если говорить про Python, то в нем есть конструкции:

a. If

b. If – else

c. If – elif – else

7. Больше, меньше, меньше либо равно, больше либо равно, равно? (>, <=, >=, ==)

8. Простое условие – два выражения, связанные одним из операторов сравнения. (x >= 5)

9. Составное условие – условия, состоящие из двух или более простых условий, связанных между собой логическими операторами. а. (x >= 5) and (x < 10)

10. And, or

11. Да, может.

12. Алгоритм циклической структуры — это алгоритм, в котором происходит многократное повторение одного и того же участка программы.

13. Цикл while (по условию), цикл for (по количеству итераций)

14. Функция range возвращает неизменяемую последовательность чисел в виде объекта range a. С помощью range можно генерировать и убывающие последовательности чисел b. Можно использовать для указания итераций цикла

15. range(15, 0, 2)

16. Да, могут

17. Бесконечные циклы появляются, когда инструкции в цикле никогда не будут противоречить условию цикла. Для избегания таких ситуаций можно использовать оператор break совместно с условными операторами внутри цикла.

18. Оператор break останавливает выполнение цикла.

19. Оператор continue запускает цикл заново.

20. Поток stdout – буферизованный поток и необходим для вывода данных. Поток stderr – небуферизованный поток для вывода ошибок.

Разделение этих потоков необходима, так как потоки могут обрабатываться ОС и пользователями поразному.

21. Для использования потока `stderr` нужно импортировать библиотеку `sys` и использовать конструкцию `print("Error!", file=sys.stderr)`

`Exit` используется для завершения программы. Передача значений отличного от нуля сообщает системе об ошибках в выполнении