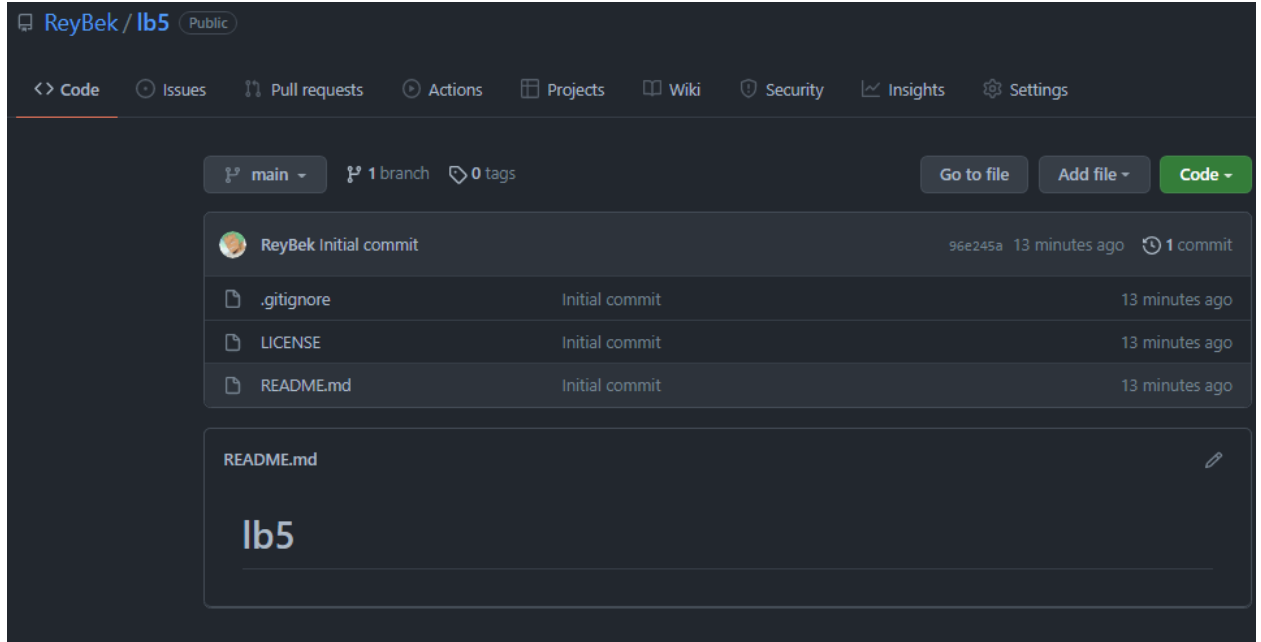


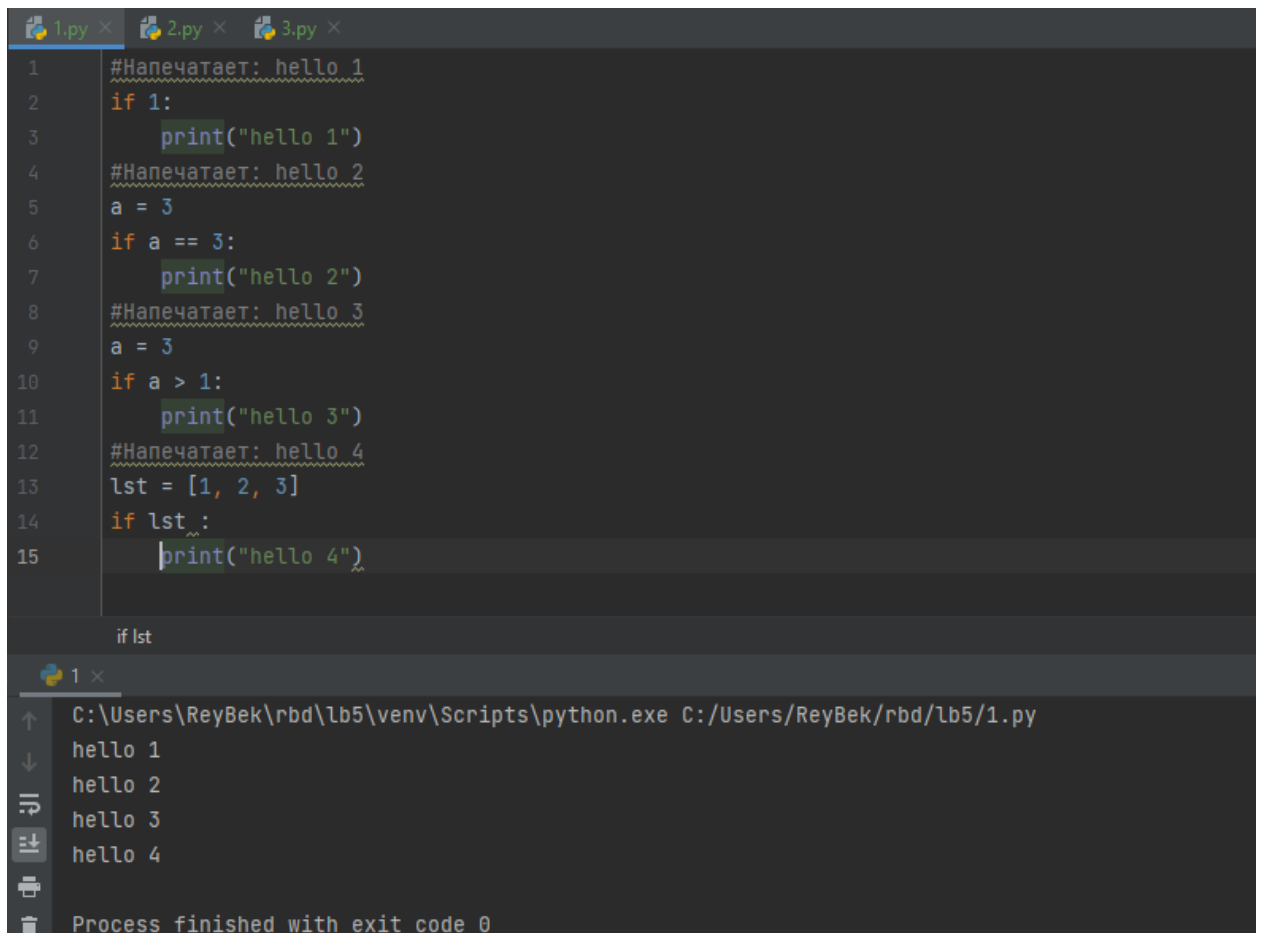
Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

ВЫПОЛНЕНИЕ

1) Создание репозитория



2) Конструкция if



Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

3) Конструкция if – else

```
1.py x
1 #Напечатает: H
2 a = 3
3 if a > 2:
4     print("H")
5 else:
6     print("L")
7 # Напечатает: L
8 a = 1
9 if a > 2:
10    print("H")
11 else:
12    print("L")
13 #В результате выполнения такого кода будет напечатано: True
14 a = 17
15 b = True if a > 10 else False
16 print(b)
```

```
1 x
C:\Users\ReyBek\rbd\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/1.py
H
L
True
Process finished with exit code 0
```

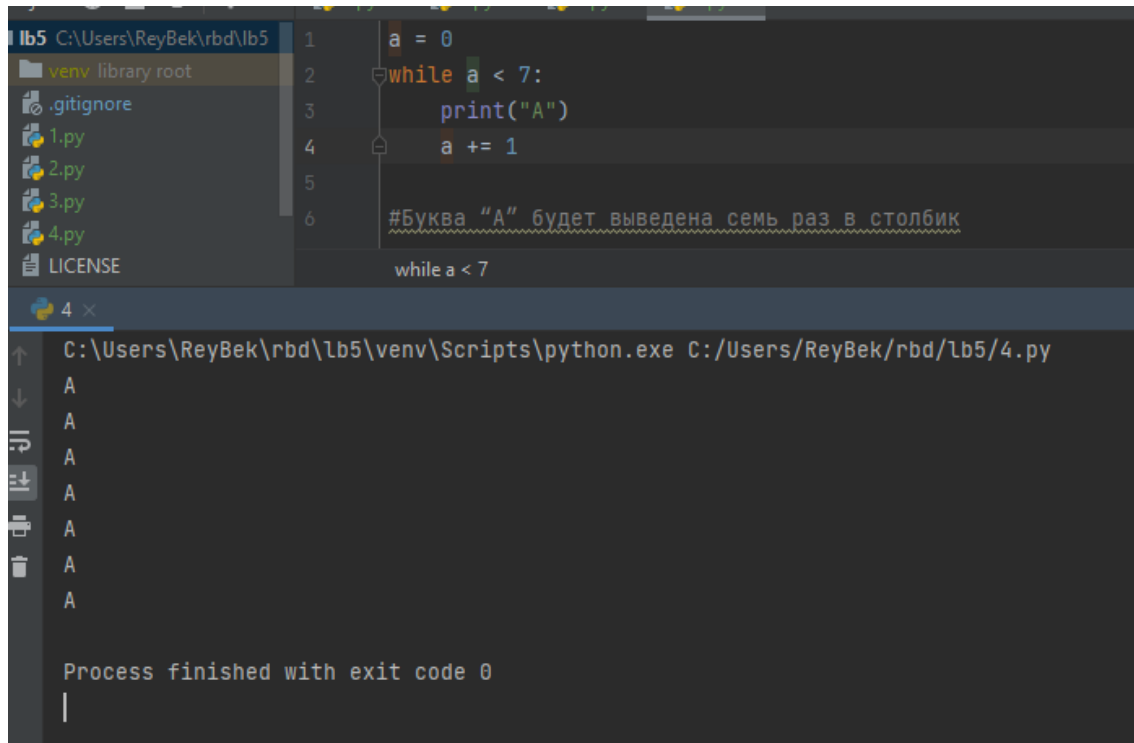
4) Конструкция if – elif – else

```
1.py x 2.py x 3.py x
1 #Конструкция if - elif - else
2 a = int(input("введите число:"))
3 if a < 0:
4     print("Neg")
5 elif a == 0:
6     print("Zero")
7 else:
8     print("Pos")
```

```
3 x
C:\Users\ReyBek\rbd\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/3.py
введите число:6
Pos
Process finished with exit code 0
```

Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

5) Оператор цикла while



The screenshot shows a code editor with a Python script and a terminal window below it.

Code Editor:

```
1 a = 0
2 while a < 7:
3     print("A")
4     a += 1
5
6 #Буква "А" будет выведена семь раз в столбик
```

Terminal:

```
C:\Users\ReyBek\rbd\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/4.py
A
A
A
A
A
A
A

Process finished with exit code 0
```

6) Операторы break и continue

```
1.py x 2.py x 3.py x 4.py x 5.py x  
1 a = 0  
2 while a >= 0:  
3     if a == 7:  
4         break  
5     a += 1  
6     print("A")  
7 a = -1  
8 #В приведенном выше коде, выход из цикла произойдет при достижении переменной а значения  
9 7. Если бы не было этого условия, то цикл выполнялся бы бесконечно..  
10 print(" ")  
11 while a < 10:  
12     a += 1  
13     if a >= 7:  
14         continue  
15     print("A")  
16 #При запуске данного кода символ "А" будет напечатан 7 раз, несмотря на то, что всего будет
```

Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

7) Оператор цикла for

```
1.py x 2.py x 3.py x 4.py x 5.py x 6.py x
1 #Оператор цикла for
2 for i in range(5):
3     print("Hello")#В результате "Hello" будет выведено пять раз
4 lst = [1, 3, 5, 7, 9]
5 for i in lst:
6     print(i ** 2) #Также можно пройти по всем буквам в строке.
7 word_str = "Hello, world!"
8 for l in word_str:
9     print(l) #Строка "Hello, world!" будет напечатана в столбик.
```

```
6 x
Hello
1
9
25
49
81
H
e
l
l
o
,
w
o
r
l
d
!
```

8) Функция range

```
1.py x 2.py x 3.py x 4.py x 5.py x 6.py x 7.py x
1 range(5) #Самый простой вариант range - передать только значение stop
2 range(0, 5)
3 list(range(5))
4 [0, 1, 2, 3, 4]
5 list(range(1, 5)) #Если передаются два аргумента, то первый используется как start, а второй - как stop
6 [1, 2, 3, 4]
7 list(range(0, 10, 2)) #И чтобы указать шаг последовательности надо передать три аргумента
8 [0, 2, 4, 6, 8]
9 list(range(0, 10, 3))
10 [0, 3, 6, 9]
11 list(range(10, 0, -1)) #С помощью range можно генерировать и убывающие последовательности чисел
12 [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
13 list(range(5, -1, -1))
14 [5, 4, 3, 2, 1, 0]
15 list(range(10, 0, -2)) #В убывающей последовательности шаг тоже может быть разным
16 [10, 8, 6, 4, 2]
17 list(range(-10, 0, 1))
18 [-10, -9, -8, -7, -6, -5, -4, -3, -2, -1] #Функция поддерживает отрицательные значения start и stop
19 list(range(0, -10, -1))
20 [0, -1, -2, -3, -4, -5, -6, -7, -8, -9]
```

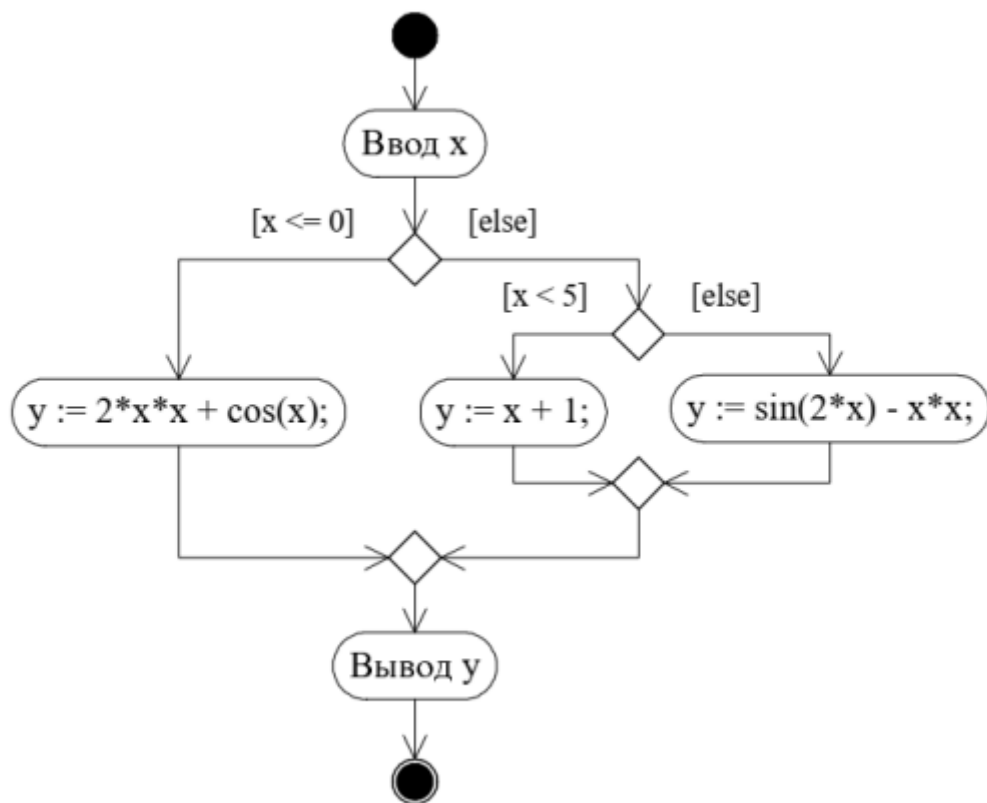
```
7 x
C:\Users\ReyBek\rbd\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/7.py

Process finished with exit code 0
```

9) Выполнение примеров

Пример 1:

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



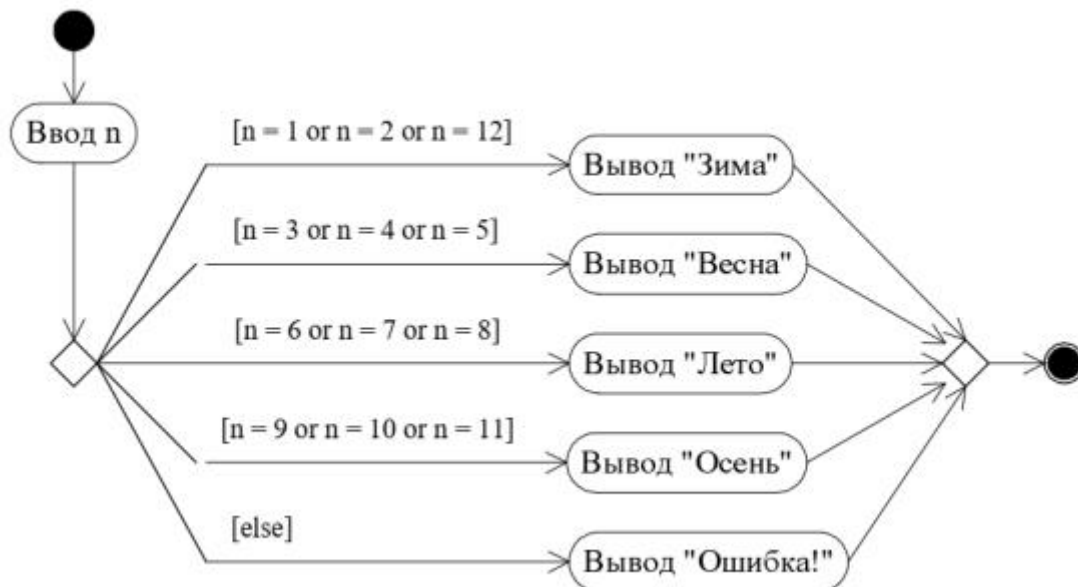
Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

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

if __name__ == '__main__':
8 x
C:\Users\ReyBek\rbd\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/8.py
Value of x? 5
y = -25.95892427466314

Process finished with exit code 0
```

Пример 2:

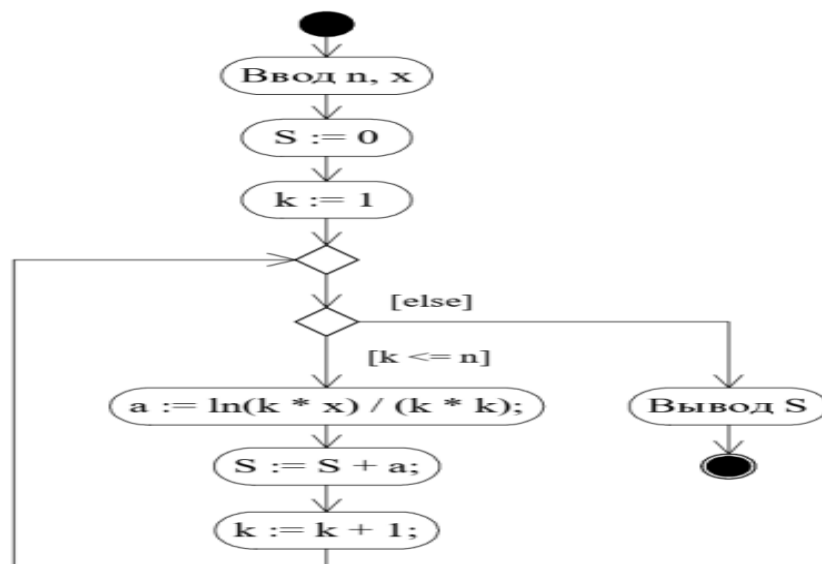


Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

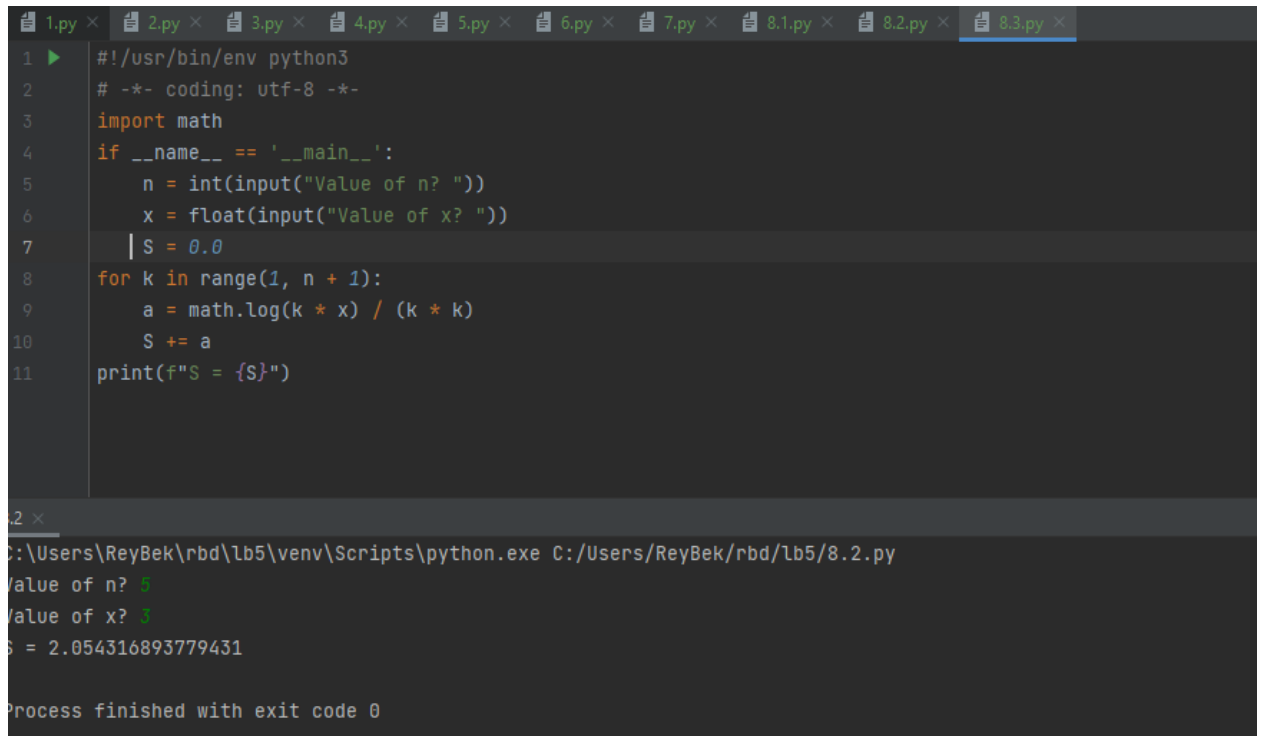
```
1.py x 2.py x 3.py x 4.py x 5.py x 6.py x 7.py x 8.1.py x 8.2.py x
1  > #!/usr/bin/env python3
2  > # -*- coding: utf-8 -*-
3      import sys
4  > if __name__ == '__main__':
5      n = int(input("Введите номер месяца: "))
6      if n == 1 or n == 2 or n == 12:
7          print("Зима")
8      elif 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)
    if __name__ == '__main__' > else
8.2 x
C:\Users\ReyBek\rbd\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/8.2.py
Введите номер месяца: 5
Весна
Process finished with exit code 0
```

Пример 3:

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



Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.



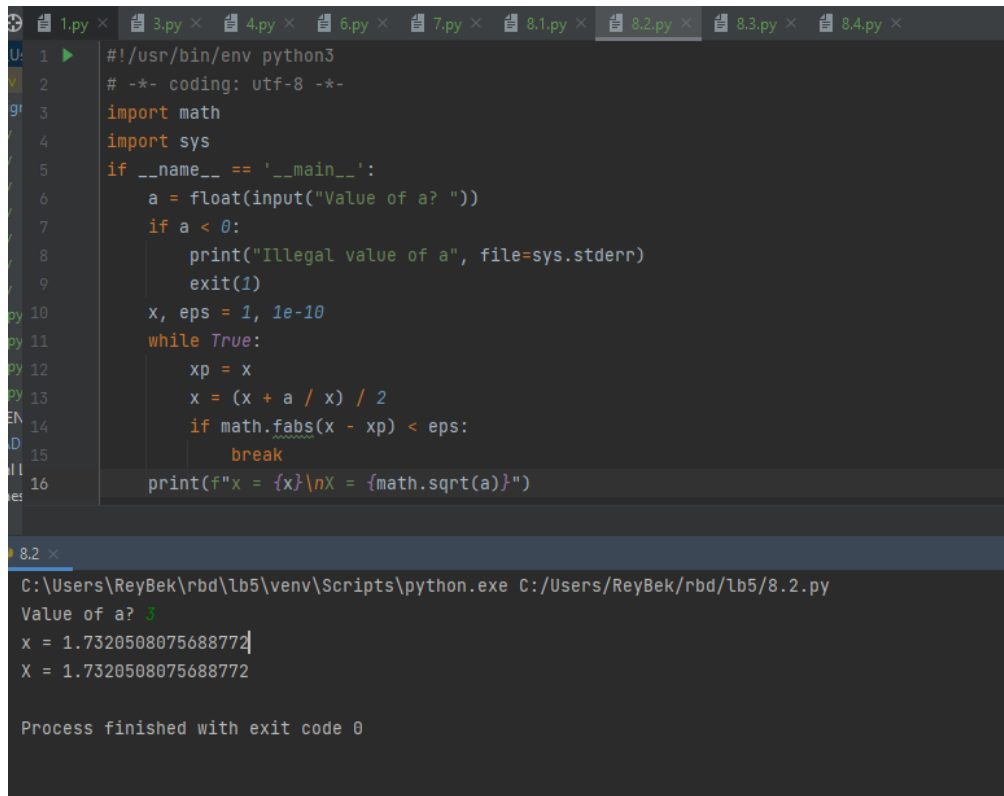
```
1 2 3 4 5 6 7 8 9 10 11
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import math
if __name__ == '__main__':
    n = int(input("Value of n? "))
    x = float(input("Value of x? "))
    S = 0.0
    for k in range(1, n + 1):
        a = math.log(k * x) / (k * k)
        S += a
    print(f"S = {S}")

C:\Users\ReyBek\rbd\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/8.2.py
Value of n? 5
Value of x? 3
S = 2.054316893779431
Process finished with exit code 0
```

Пример 4:

$$x_{n+1} = \frac{1}{2} \cdot \left(x_n + \frac{a}{x_n} \right).$$

Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.



```
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(f"x = {x}\nX = {math.sqrt(a)}")
```

8.2 x

C:\Users\ReyBek\rbd\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/8.2.py

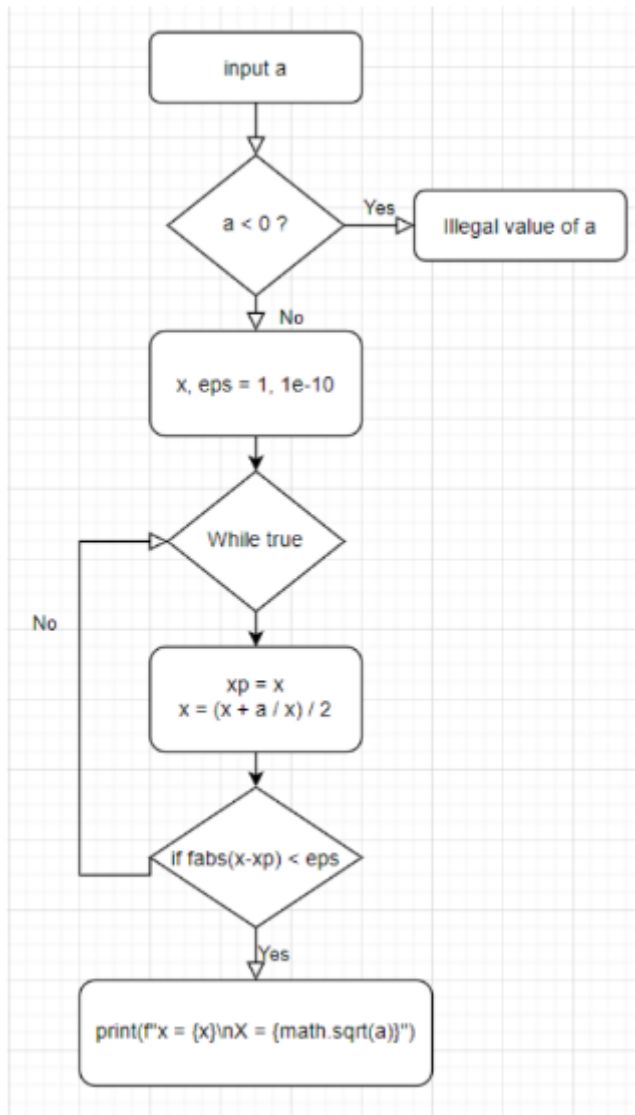
Value of a? 3

x = 1.7320508075688772

X = 1.7320508075688772

Process finished with exit code 0

Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.



Пример 5:

$$a_1 = \frac{x^1}{1 \cdot 1!} = x.$$

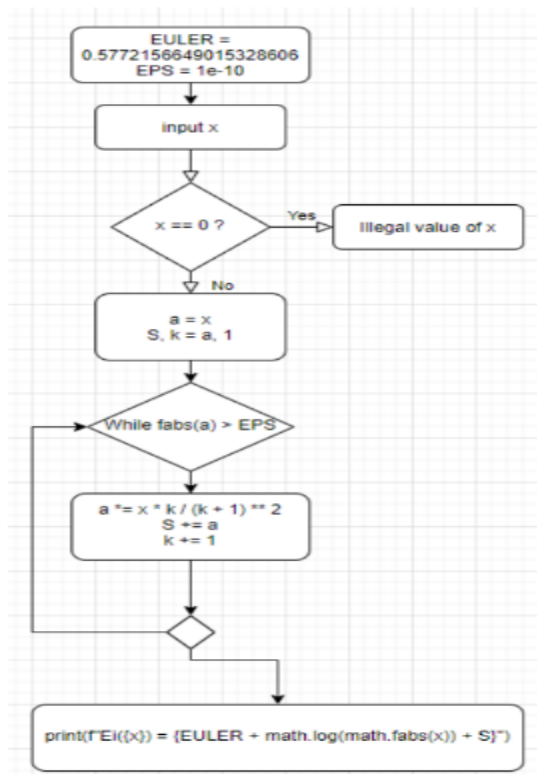
Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import math
4  import sys
5  # Постоянная Эйлера.
6  EULER = 0.5772156649015328606
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"Fi({x}) = {EULER + math.log(math.fabs(x)) + S}")
```

8.2 ×

C:\Users\ReyBek\rbdlb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/8.2.py
Value of x? 4
Ei(4.0) = 19.63087447005282

Process finished with exit code 0

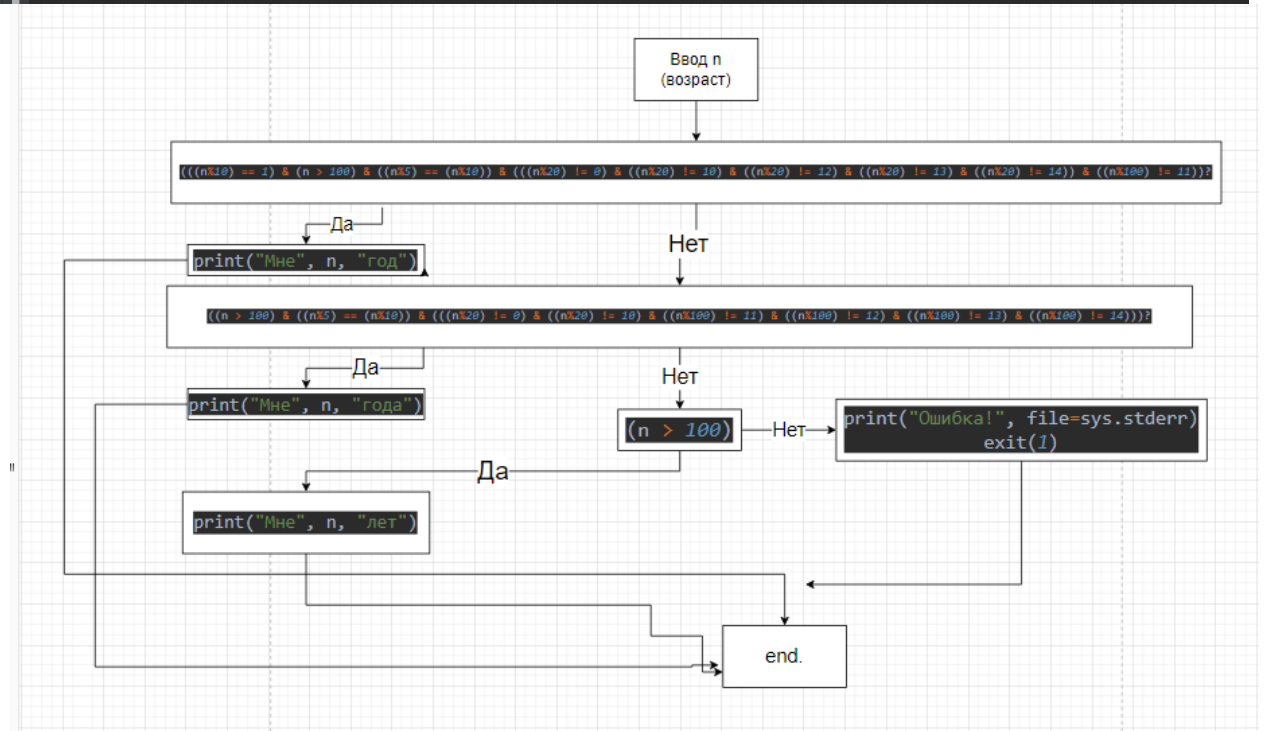


Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

Индивидуальное задание 1

```
1 import sys
2 if __name__ == '__main__':
3     n = int(input("Введите ваш возраст: "))
4     if (((n%10) == 1) & (n > 100) & ((n%5) == (n%10)) & (((n%20) != 0) & ((n%20) != 10) & ((n%20) != 12) & ((n%20) != 13) & ((n%20) != 14)) & ((n%100) != 11)):
5         print("Мне", n, "год")
6     elif ((n > 100) & ((n%5) == (n%10)) & (((n%20) != 0) & ((n%20) != 10) & ((n%100) != 11) & ((n%100) != 12) & ((n%100) != 13) & ((n%100) != 14))):
7         print("Мне", n, "года")
8     elif (n > 100):
9         print("Мне", n, "лет")
10    else:
11        print("Ошибка!", file=sys.stderr)
12        exit(1)
13
```

C:\Users\ReyBek\rbdb\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rb5/lb5/ind1.py
Введите ваш возраст: 131
Мне 131 год
Process finished with exit code 0



Индивидуальное задание 2

Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

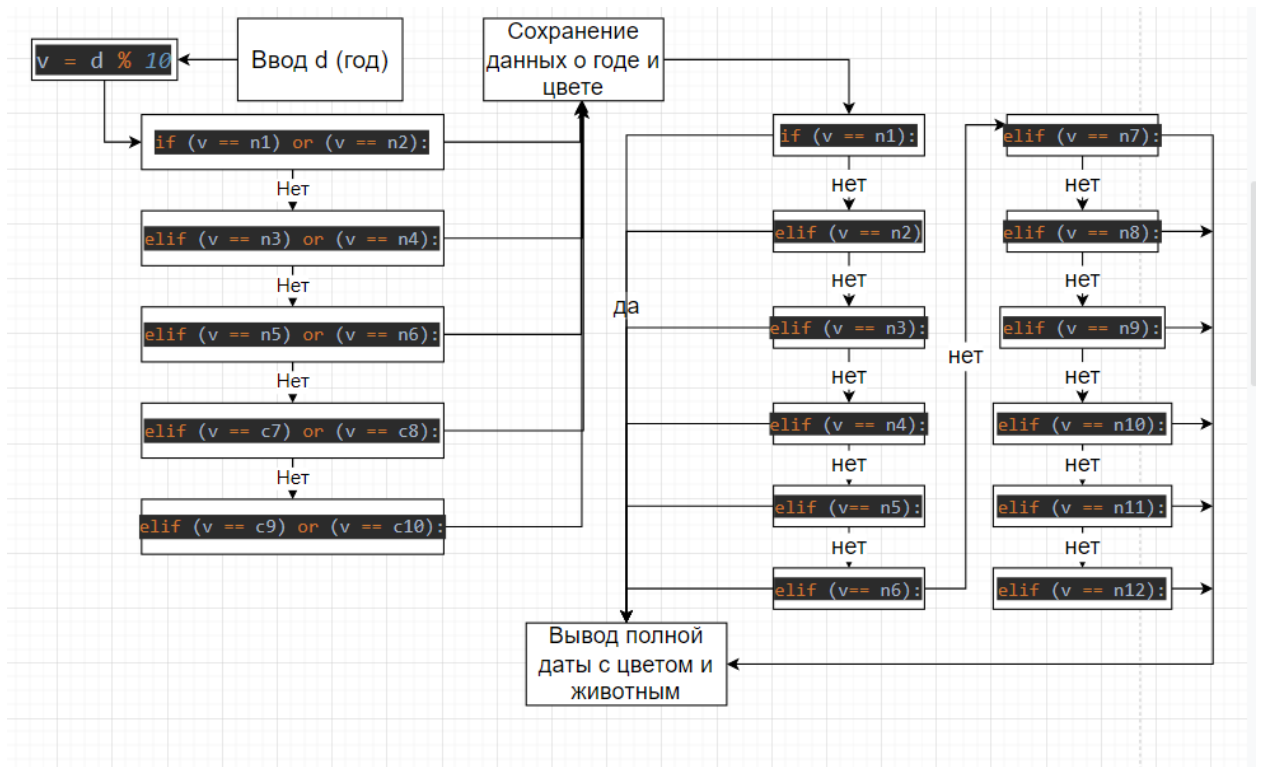
```
ind2.py x
1 import math
2 import sys
3 d = int(input("Введите год: "))
4 n1 = 4
5 n2 = 5
6 n3 = 6
7 n4 = 7
8 n5 = 8
9 n6 = 9
10 n7 = 10
11 n8 = 11
12 n9 = 0
13 n10 = 1
14 n11 = 2
15 n12 = 3
16 c7 = 1
17 c8 = 2
18 c9 = 3
19 c10 = 4
```

```
ind2.py x
C:\Users\ReyBek\rbd\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/ind2.py
Введите год: 1984
1984 - Год зеленой мыши
Process finished with exit code 0
```

```
ind2.py x
1 import math
2 import sys
3 d = int(input("Введите год: "))
4 n1 = 4
5 n2 = 5
6 n3 = 6
7 n4 = 7
8 n5 = 8
9 n6 = 9
10 n7 = 10
11 n8 = 11
12 n9 = 0
13 n10 = 1
14 n11 = 2
15 n12 = 3
16 c7 = 1
17 c8 = 2
18 c9 = 3
19 c10 = 4
20 v = d % 10
21 if (v == n1) or (v == n2):
22     s = ("- Год зеленой")
23 elif (v == n3) or (v == n4):
24     s = ("- Год красной")
25 elif (v == n5) or (v == n6):
26     s = ("- Год желтой")
27 elif (v == c7) or (v == c8):
28     s = ("- Год белой")
29 elif (v == c9) or (v == c10):
30     s = ("- Год черной")
31 else:
```

Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

```
ind2.py x
26     s =("- Год желтой")
27     elif (v == c7) or (v == c8):
28         s =("- Год белой")
29     elif (v == c9) or (v == c10):
30         s =("- Год черной")
31     else:
32         print("ошибка1")
33     if (v == n1):
34         print(d, s, "мыши")
35     elif (v == n2):
36         print(d, s, "коровы")
37     elif (v == n3):
38         print(d, s, "тигра")
39     elif (v == n4):
40         print(d, s, "зайца")
41     elif (v == n5):
42         print(d, s, "дракона")
43     elif (v == n6):
44         print(d, s, "змеи")
45     elif (v == n7):
46         print(d, s, "лошади")
47     elif (v == n8):
48         print(d, s, "овцы")
49     elif (v == n9):
50         print(d, s, "обезьяны")
51     elif (v == n10):
52         print(d, s, "курицы")
53     elif (v == n11):
54         print(d, s, "собаки")
55     elif (v == n12):
56         print(d, s, "свиньи")
```



Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

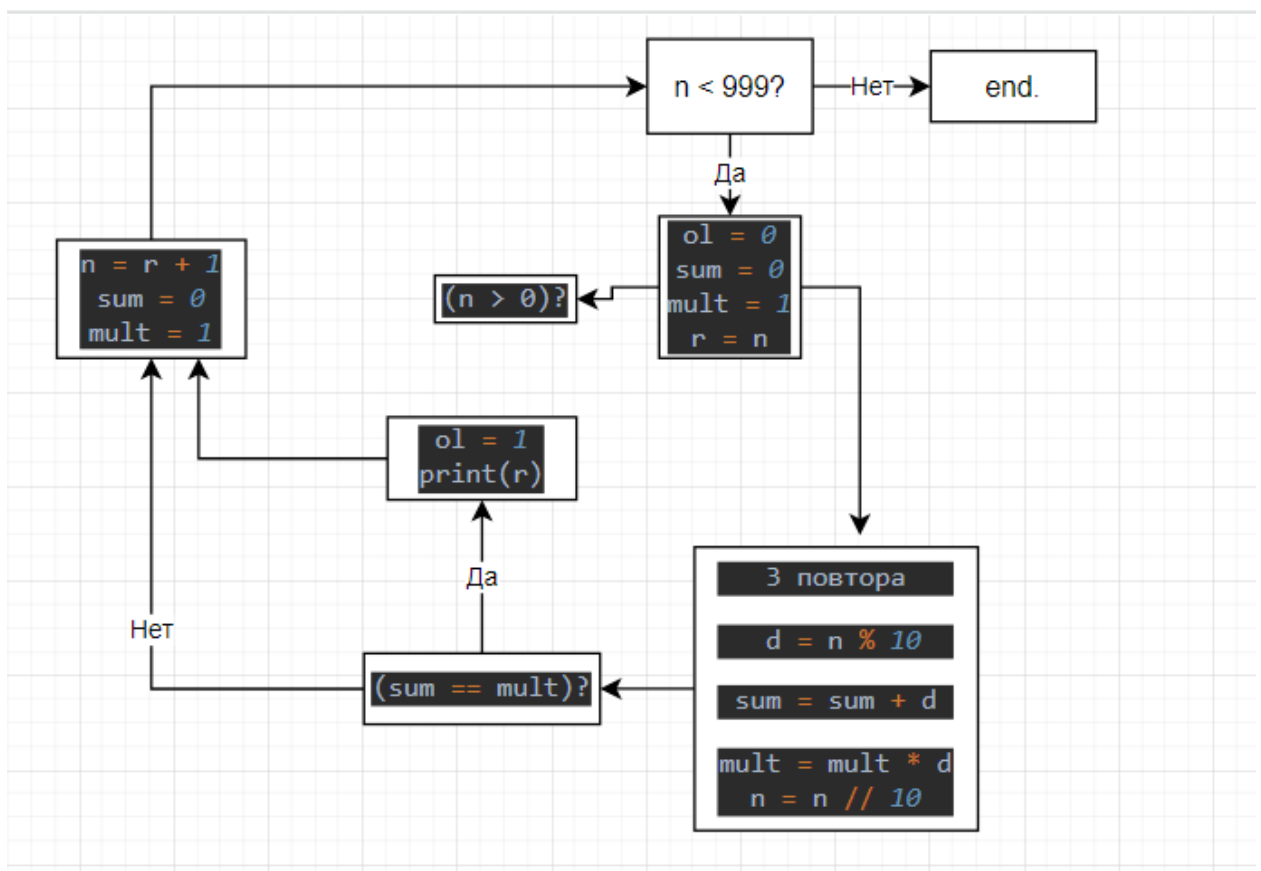
Индивидуальное задание 3

```
ind3.py
1  n = 123
2  while n < 999:
3      ol = 0
4      sum = 0
5      mult = 1
6      r = n
7      if (n > 0):
8          d = n % 10
9          sum = sum + d
10         mult = mult * d
11         n = n // 10
12         d = n % 10
13         sum = sum + d
14         mult = mult * d
15         n = n // 10
16         d = n % 10
17         sum = sum + d
18         mult = mult * d
19         n = n // 10
20         if sum == mult:
21             ol = 1
22         if ol == 1:
23             print(r)
24         n = r + 1
25         sum = 0
26         mult = 1
```

Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

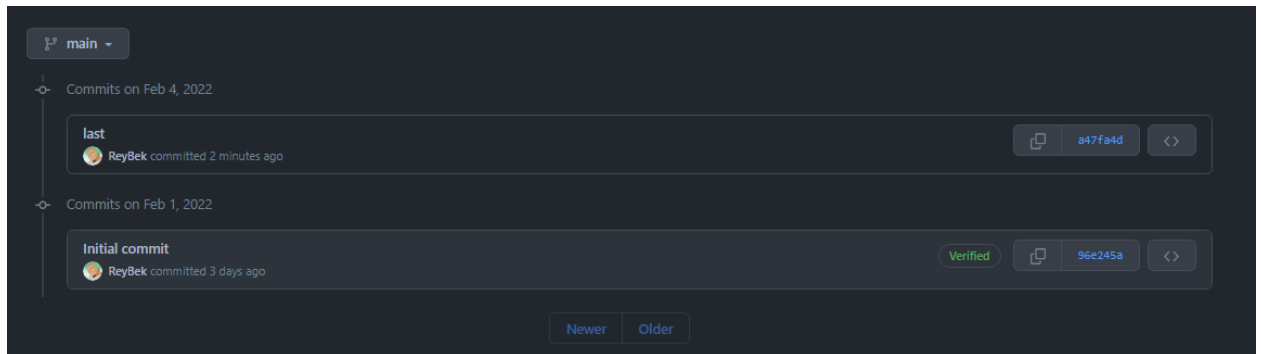
```
ind3.py x
5      mult = 1
6      r = n
7      if (n > 0):
8          d = n % 10
9          sum = sum + d
10         mult = mult * d
11         n = n // 10
12         d = n % 10
13         sum = sum + d
14         mult = mult * d
15         n = n // 10
16         d = n % 10

ind3.py x
C:\Users\ReyBek\rbd\lb5\venv\Scripts\python.exe C:/Users/ReyBek/rbd/lb5/ind3.py
123
132
213
231
312
321
```



10)История коммитов

Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.



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 результат 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. ($x \geq 5$)

Отчет №5 по дисциплине
“Основы программной инженерии”
Группа: ПИЖ-б-о-20-1, Бекишев М. М.

9. Составное условие – условия, состоящие из двух или более простых условий, связанных между собой логическими операторами. а. $(x \geq 5)$ 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 польз. сценариями поразному.
21. Для использования потока stderr нужно импортировать библиотеку sys и использовать конструкцию `print("Error!", file=sys.stderr)`
22. Exit используется для завершения программы. Передача значений отличного от нуля сообщает системе об ошибках в выполнении программы.