

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Основы кроссплатформенного программирования**

**Отчет по лабораторной работе №3**

**«Условные операторы и циклы в языке Python3»**

Выполнил студент группы

ИТС-б-о-20-1(2)

Аблаев Д.К. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

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

Проверил доцент

Кафедры инфокоммуникаций, старший  
преподаватель

Воронкин Р.А.

\_\_\_\_\_  
(подпись)

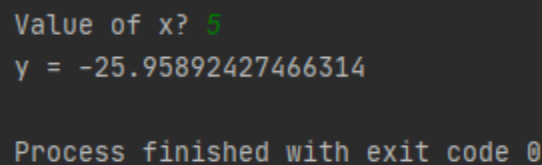
Ставрополь 2021

**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.

**Выполнение работы:**

Репозиторий: <https://github.com/AblaevDaniil/LR3>

**Пример 1**

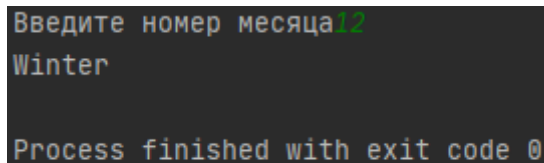


```
Value of x? 5
y = -25.95892427466314

Process finished with exit code 0
```

Рисунок 1. Окно вывода программы

**Пример 2**

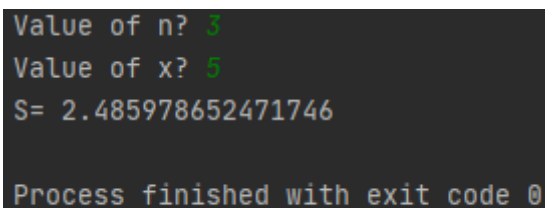


```
Введите номер месяца12
Winter

Process finished with exit code 0
```

Рисунок 2. Окно вывода программы

**Пример 3**

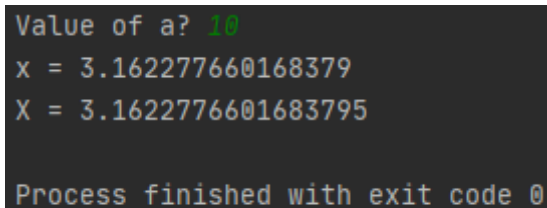


```
Value of n? 3
Value of x? 5
S= 2.485978652471746

Process finished with exit code 0
```

Рисунок 3. Окно вывода программы

**Пример 4**



```
Value of a? 10
x = 3.162277660168379
X = 3.1622776601683795

Process finished with exit code 0
```

Рисунок 4. Окно вывода программы

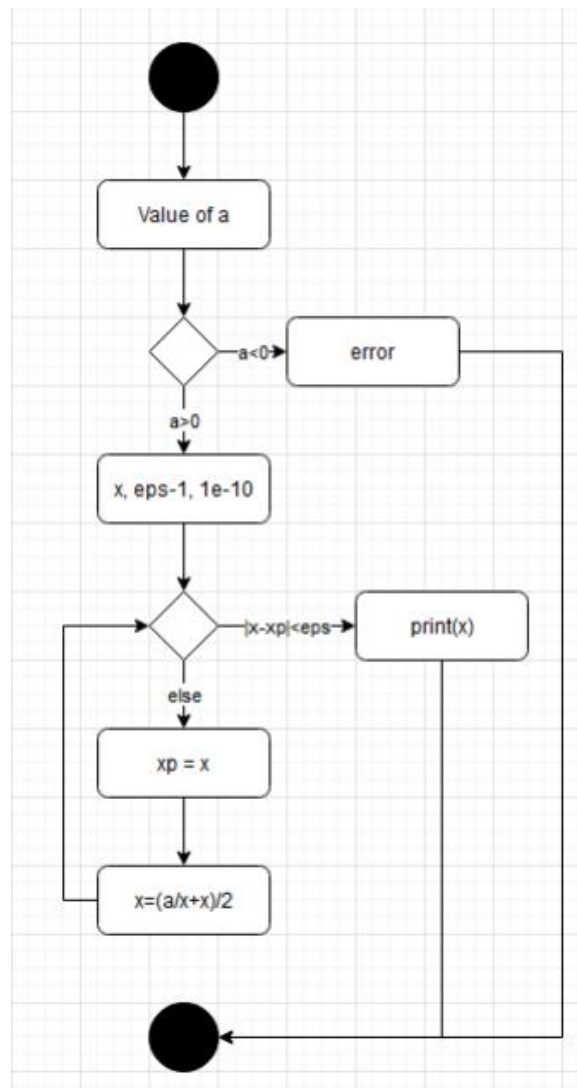


Рисунок 5. UML диаграмма

### Пример 5

```

Value of x?100
Ei(100.0) = 2.7155527448538826e+41

Process finished with exit code 0
  
```

Рисунок 6. Окно вывода программы

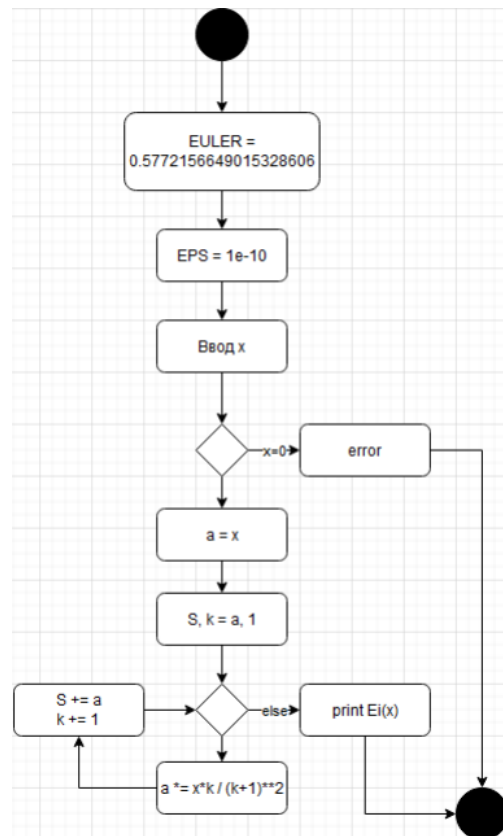


Рисунок 7. UML диаграмма

### Задание 1.

1. Дано натуральное число  $n > 100$ . Вывести на экран фразу мне  $n$  лет, учитывая, что при некоторых значениях  $n$  слово «лет» надо заменить на слово «год» или «года».

```

Введите вас возраст: 61
мне 61 год

Process finished with exit code 0
  
```

Рисунок 8. Окно вывода программы

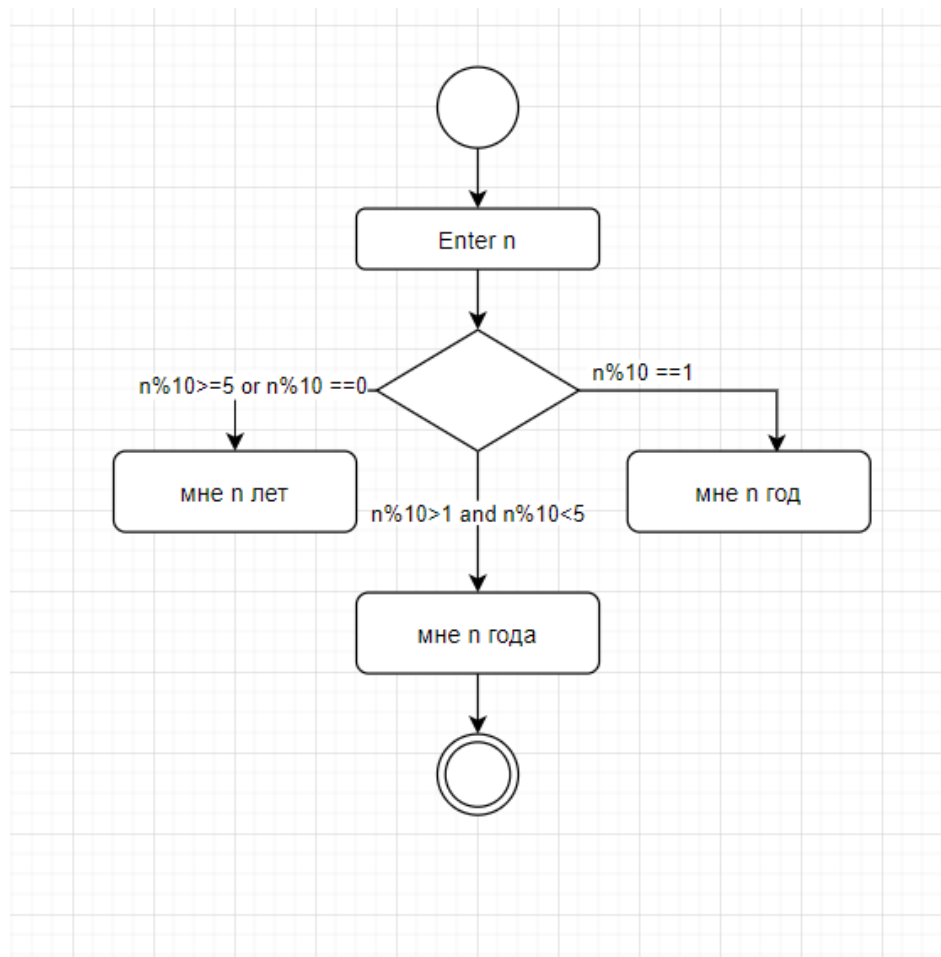


Рисунок 9. UML диаграмма

Задание 2.

23. Симметричны ли точки  $M_1(x_1, y_1)$  и  $M_2(x_2, y_2)$  относительно оси  $ox$  или относительно оси  $oy$ ?

```

3
1
2
3
Точки не симметричны
Process finished with exit code 0
  
```

Рисунок 10. Окно вывода программы

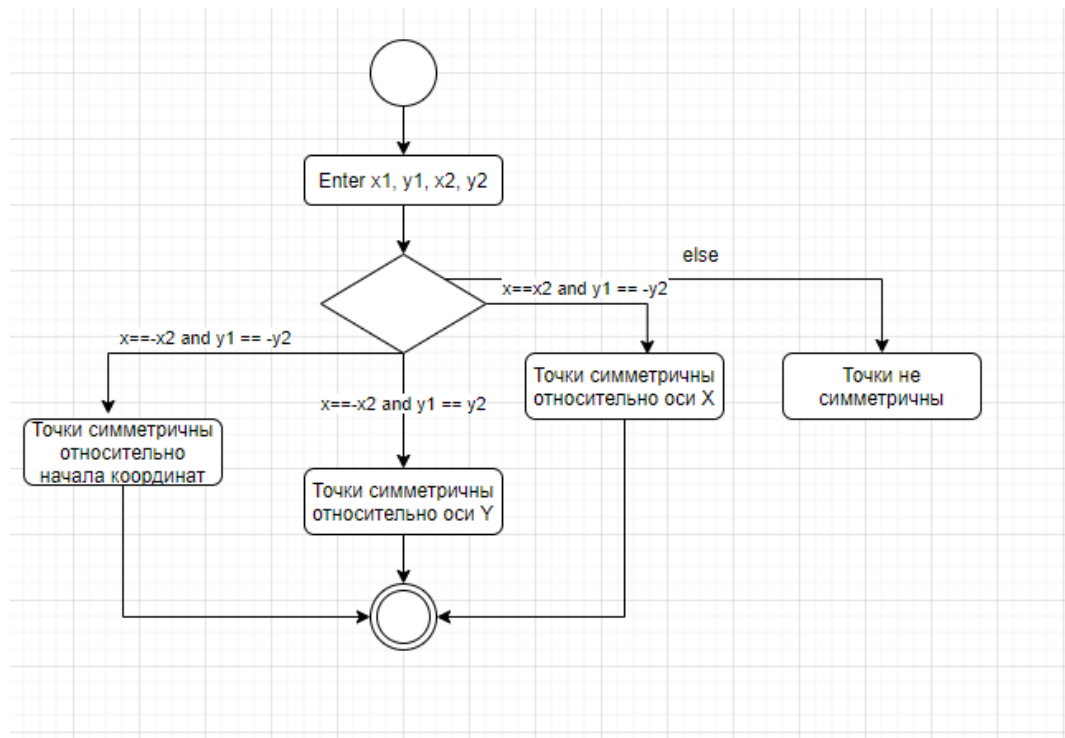


Рисунок 11. UML диаграмма

### Задание 3.

21. Ежемесячная стипендия студента составляет  $A$  р., а расходы на проживание превышают стипендию и составляют  $B$  р. в месяц. Рост цен ежемесячно увеличивает расходы на 3%. Составьте программу расчета необходимой суммы денег, которую надо одновременно просить у родителей, чтобы можно было прожить учебный год (10 месяцев), используя только эти деньги и стипендию.

```

ежемесячная стипендия2000
ежемесячные расходы5000
37319.40

Process finished with exit code 0
  
```

Рисунок 12. Окно вывода программы

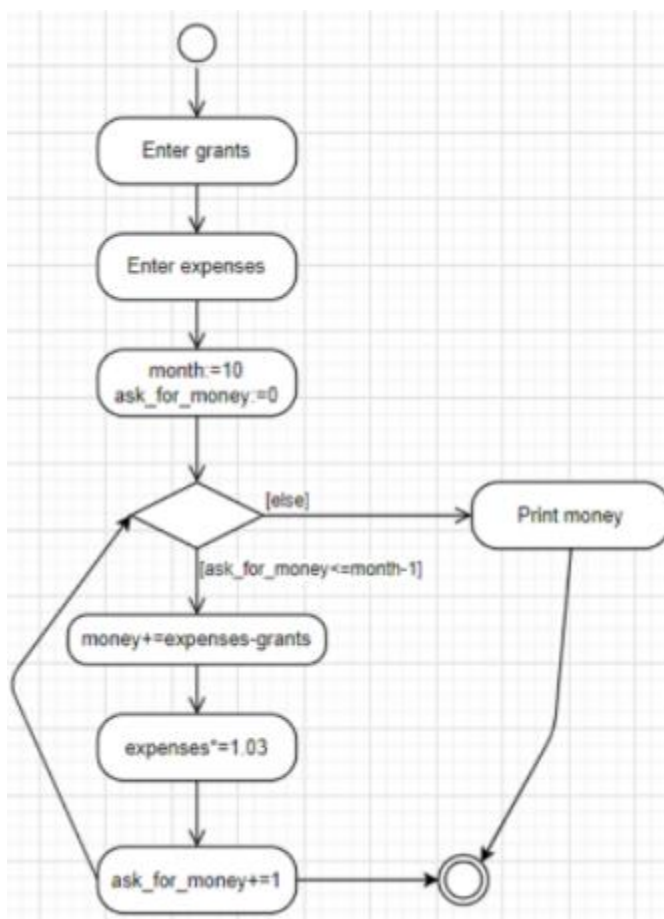


Рисунок 13. UML диаграмма

#### Задание повышенной сложности

7. Функция Бесселя первого рода  $I_n(x)$ , значение  $n = 0, 1, 2, \dots$  также должно вводиться с клавиатуры

$$I_n(x) = \left(\frac{x}{2}\right)^n \sum_{k=0}^{\infty} \frac{(x^2/4)^k}{k!(k+n)!}$$

```

Введите x -->2
Введите n -->5
1440.0

Process finished with exit code 0

```

Рисунок 14. Окно вывода программы

**Вывод:** в ходе выполнения лабораторной работы были приобретены навыки программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоены операторы языка Python версии 3 - if , while , for , break и continue позволяющих реализовывать разветвляющиеся

алгоритмы и алгоритмы циклической структуры. Тык же были построены UML диаграммы.

Контрольные вопросы:

1. Для чего нужны диаграммы деятельности UML?

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. Это очень выразительный язык, позволяющий рассмотреть систему со всех точек зрения, имеющих отношение к ее разработке и последующему развертыванию. Несмотря на обилие выразительных возможностей, этот язык прост для понимания и использования.

2. Что такое состояние действия и состояние деятельности?

Состояния действия не могут быть подвергнуты декомпозиции. Кроме того, они атомарны. Это значит, что внутри них могут происходить различные события, но выполняемая в состоянии действия работа не может быть прервана. Обычно предполагается, что длительность одного состояния действия занимает неощутимо малое время. В противоположность этому состояния деятельности могут быть подвергнуты дальнейшей декомпозиции, вследствие чего выполняемую деятельность можно представить с помощью других диаграмм деятельности. Состояния деятельности не являются атомарными, то есть могут быть прерваны. Предполагается, что для их завершения требуется заметное время. Можно считать, что состояние действия - это частный вид состояния деятельности, а конкретнее - такое состояние, которое не может быть подвергнуто дальнейшей декомпозиции. А состояние деятельности можно представлять себе как составное состояние, поток управления которого включает только другие состояния деятельности и действий.



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

В UML переход представляется простой линией со стрелкой. Точка ветвления представляется ромбом. В точку ветвления может входить ровно один переход, а выходить - два или более

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. В программе разветвляющейся структуры имеется один или несколько условных операторов. Для программной реализации условия используется логическое выражение. В сложных структурах с большим числом ветвей применяют оператор выбора.

5. Чем отличается разветвляющийся алгоритм от линейного?

В отличие от линейных алгоритмов, в которых команды выполняются последовательно одна за другой, в алгоритмическую структуру «ветвление» входит условие, в зависимости от выполнения или невыполнения которого реализуется та или иная последовательность команд.

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

Оператор ветвления `if` позволяет выполнить определенный набор инструкций в зависимости от некоторого условия. Возможны следующие варианты использования.

.Конструкция `if`;

Конструкция `if – else`;

Конструкция `if – elif – else`.

7. Какие операторы сравнения используются в Python?

Операторы сравнения используются для сравнения двух значений:

`==` Проверяет равны ли оба операнда. Если да, то условие становится истинным.

!= Проверяет равны ли оба операнда. Если нет, то условие становится истинным.

<> Проверяет равны ли оба операнда. Если нет, то условие становится истинным.

> Проверяет больше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.

< Проверяет меньше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.

>= Проверяет больше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.

<= Проверяет меньше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.

8. Что называется простым условием? Приведите примеры.

Простое условие — это два выражения, связанные одним из знаков отношений: = (равно), > (больше), < (меньше), >= (больше либо равно), <= (меньше либо равно), != (не равно). Если условие выполняется, то говорят, что условие истинно, в противном случае — условие ложно.

a < 0

b > 0

a + 2 \* b == c / 3

9. Что такое составное условие? Приведите примеры.

Составные условия — это условия, состоящие из двух или более простых условий, соединенных с помощью логических операций: and, or, not. Простые условия при этом заключаются в скобки. (x == 0) or (x == 8) not (a == 0) or (b == 0)

10. Какие логические операторы допускаются при составлении сложных условий?

СЛОЖНОЕ УСЛОВИЕ - состоит из двух или нескольких простых отношений (условий), которые объединяются с помощью логических операций: И - логическое умножение - на языке Python записывается как and,

ИЛИ - логическое умножение - на языке Python записывается как `or`, НЕ - логическое умножение -на языке Python записывается как `not`.

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

Ветви оператора `if` могут содержать любые допустимые к использованию внутри функции конструкции. Например, объявление переменных, ветвления и т.д. Ветвления могут быть вложены друг в друга.

12. Какой алгоритм является алгоритмом циклической структуры?

Циклический алгоритм – описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие. Перечень повторяющихся действий называют телом цикла.

13. Типы циклов в языке Python.

В Python есть два типа циклов: `for` и `while`.

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

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

`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. Могут ли быть циклы вложенными?

Вложенными называются циклы, которые выполняются внутри других циклов.

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

Бесконечный цикл – это цикл, написанный таким образом, что он никогда не прекратит своё выполнение, так как никогда не выполнится условие выхода из этого цикла.

17. Для чего нужен оператор `break`? В Python выражение `break` дает вам возможность выйти из цикла при активации внешнего условия.

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

Этот оператор используется в циклах `for`, `while` и необходим для того, чтобы прервать выполнение текущей итерации цикла с места его вызова и перейти к следующей итерации. Естественно, как и `break` оператор `continue` оператор является оператором завершающим тело условия.

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

Стандартный вывод — программа пользователя записывает обычную информацию в этот дескриптор файла. Вывод возвращается через стандартный вывод (`stdout`).

Стандартная ошибка — программа пользователя записывает информацию об ошибке в этот дескриптор файла. Ошибки возвращаются через стандартную ошибку (`stderr`).

20. Каково назначение функции `exit`?

Функция `exit()` модуля `sys` - выход из Python. Она реализуется путем вызова исключения `SystemExit`, поэтому выполняются действия по очистке, указанные в предложениях `finally` операторов `try` и можно перехватить попытку выхода на внешнем уровне.