

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 Python 3

Выполнил студент группы

ИТС-б-о-20-1

Попов Д.В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2021

Условные операторы и циклы в языке Python3

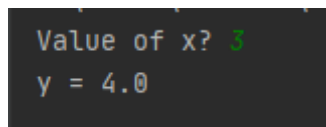
Цель работы: приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры.

Освоить операторы языка Python версии 3 if , while , for , break и continue позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

Порядок выполнения работы:

1. Проработаем примеры лабораторной работы. Создадим для каждого примера отдельный модуль языка Python. Зафиксируем изменения в репозитории (ссылка).

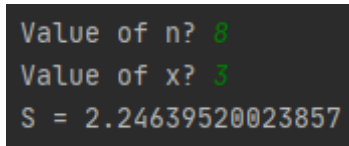
Пример 1.



```
Value of x? 3
y = 4.0
```

Рисунок 1 – Окно вывода программы примера 1

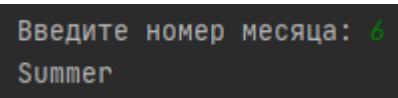
Пример 2.



```
Value of n? 8
Value of x? 3
S = 2.24639520023857
```

Рисунок 2 – Окно вывода программы примера 2

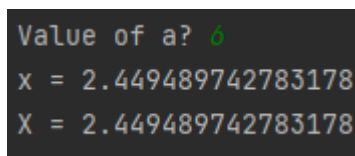
Пример 3.



```
Введите номер месяца: 6
Summer
```

Рисунок 3 – Окно вывода программы примера 3

Пример 4.



```
Value of a? 6
x = 2.449489742783178
X = 2.449489742783178
```

Рисунок 4 – Окно вывода программы примера 4

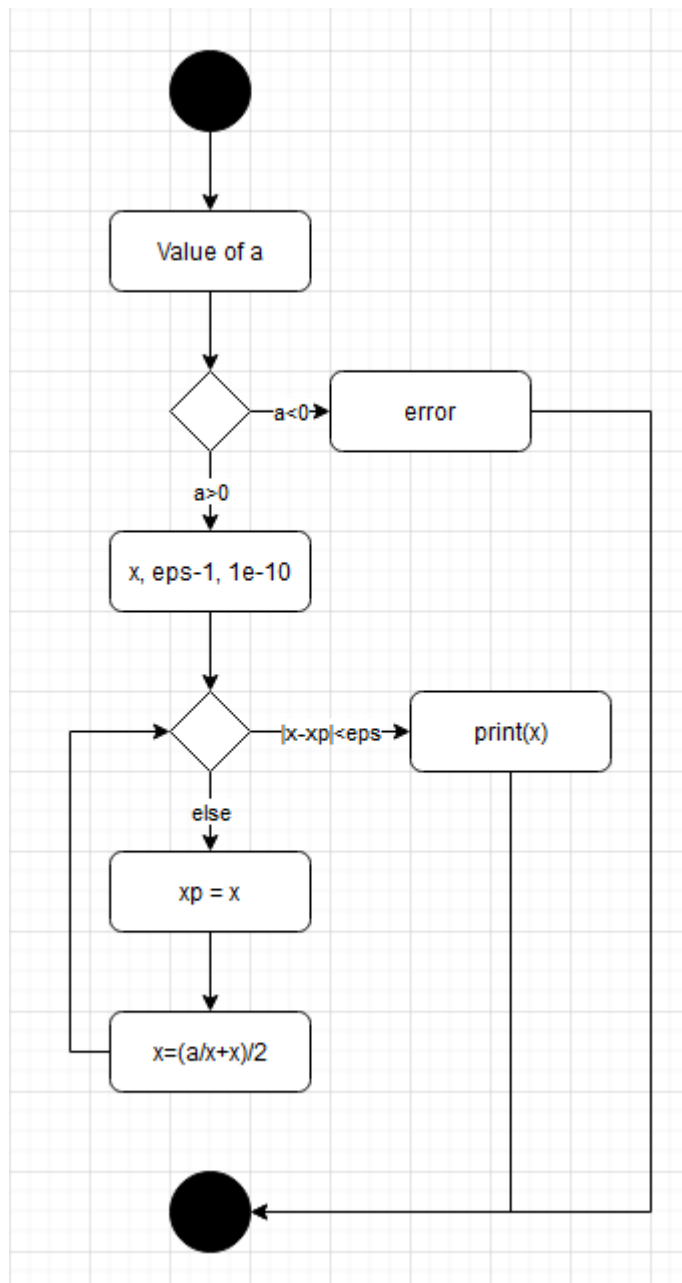


Рисунок 5 – UML диаграмма для примера 4

Пример 5.

```

Value of x? 60
Ei(60.0) = 1.936182213929278e+24
  
```

Рисунок 6 – Окно вывода программы примера 5

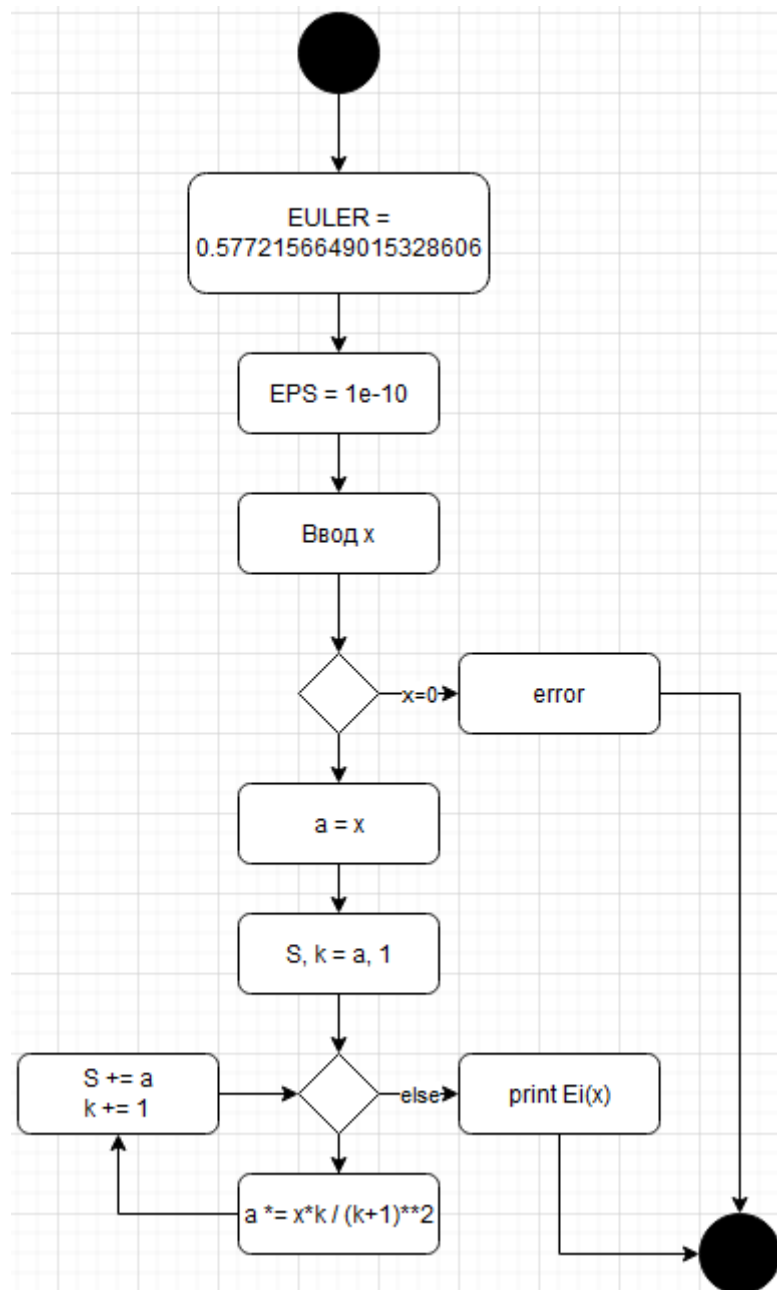


Рисунок 7 – UML диаграмма для примера 5

Индивидуальные задания:

Задание 1.

2. Дано число m ($1 \leq m \leq 12$). Определить, сколько дней в месяце с номером m .

```

Введите номер месяца 3
30 days

Process finished with exit code 0
  
```

Рисунок 8 – Пример работы программы

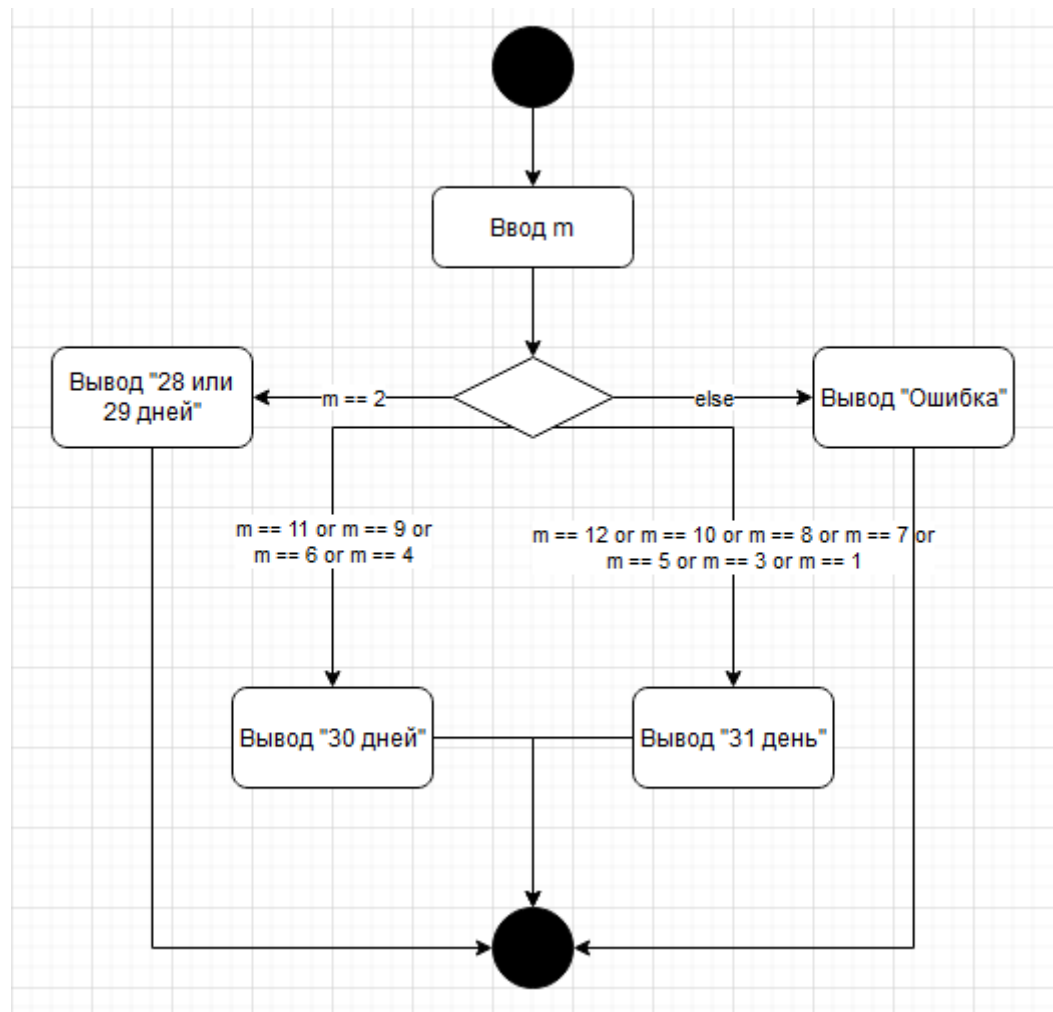


Рисунок 9 – UML диаграмма задания 1

Задание 2.

14. Составить программу нахождения из трех чисел наибольшего и наименьшего.

```

3
5
7
7
3

Process finished with exit code 0
  
```

Рисунок 10 – Пример работы программы

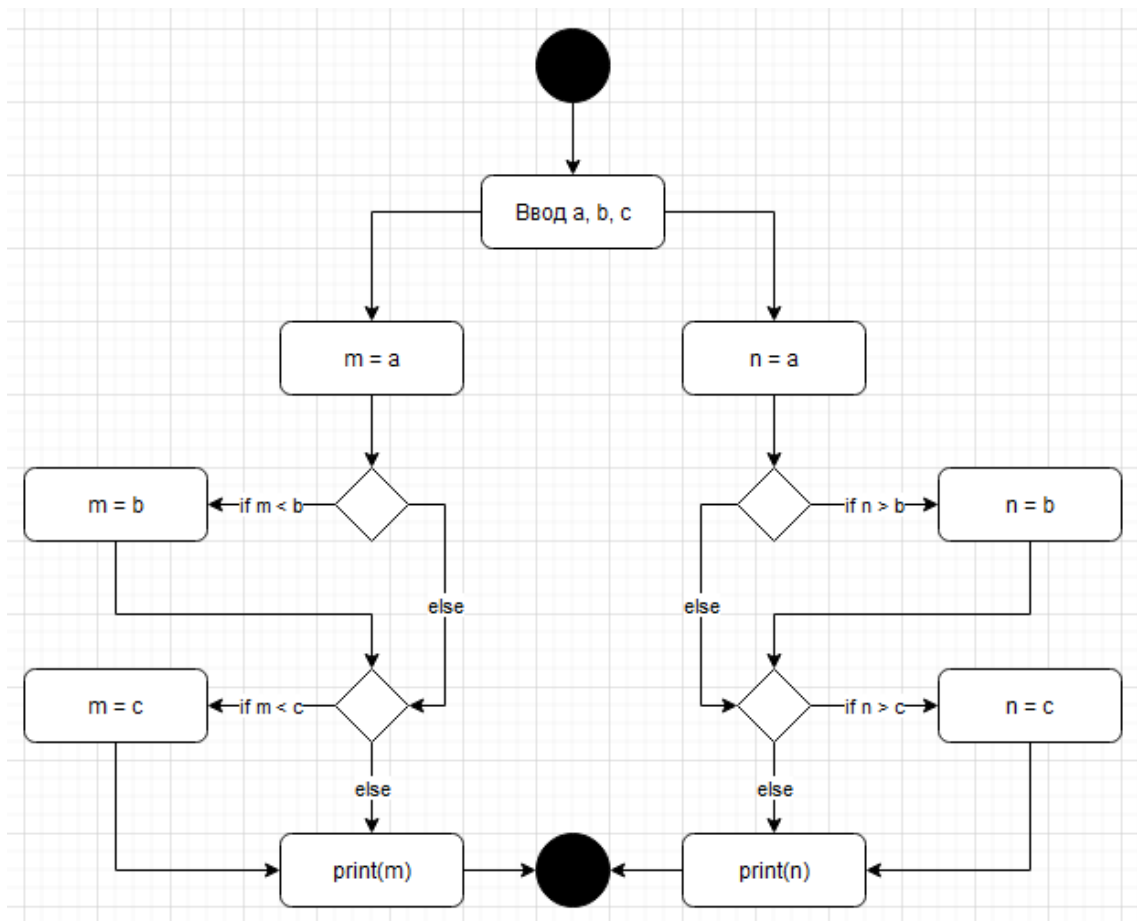


Рисунок 11 – UML диаграмма задания 2

Задание 3.

13. Дано натуральное число n . Получить все его натуральные делители.

```

96
1  2  3  4  6  8  12  16  24  32  48  96
Process finished with exit code 0
|
  
```

Рисунок 12 – Пример работы программы

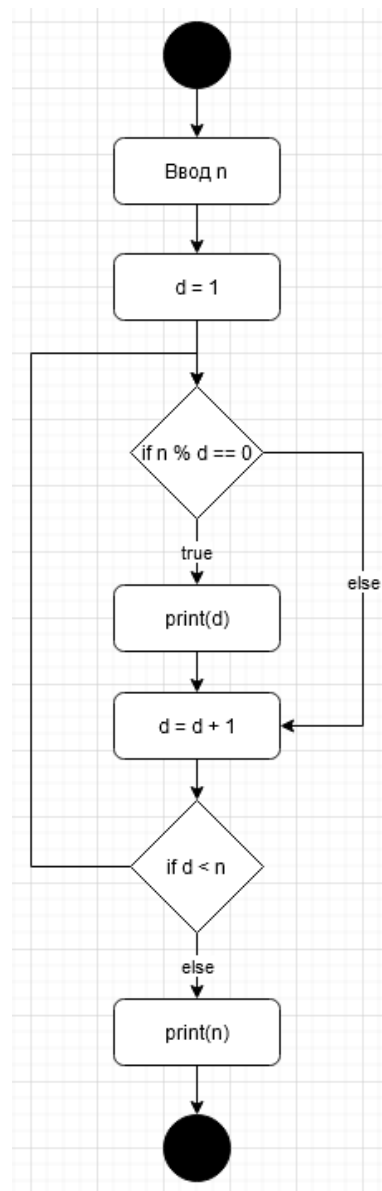


Рисунок 13 – UML диаграмма задания 3

Вывод: приобрел навыки программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоил операторы языка Python версии 3 if, while, for, break и continue, позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

Контрольные работы:

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.

1. Конструкция `if`;
2. Конструкция `if – else`;
3. 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.

7. 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).

Если условие выполняется, то говорят, что условие истинно, в противном случае — условие ложно.

`a0`

`b`

`a+2* b =c/3`

8. 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)`

9. 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`.

10. 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.

11. 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.

12. Типы циклов в языке Python.

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

13. 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

14. Могут ли быть циклы вложенными?

Вложенными называются циклы, которые выполняются внутри других циклов.

15. 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.

16. Для чего нужен оператор `break`?

В Python выражение `break` дает вам возможность выйти из цикла при активации внешнего условия.

17. 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` является оператором завершающим тело условия.

18. 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`).

19. 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.

<https://github.com/12W300/third>