

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

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.2

Условные операторы и циклы в языке Python3

Выполнил студент группы

ИВТ-б-о-21-1 (2)

Стригалов Д.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2022

Цель работы: приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоить операторы языка Python версии 3 if, while, for, break и continue, позволяющие реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

Ход работы:

Пример 1.

```
C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe F:/Programs/GitHub/Lab-2.1/Python/arithmetic.py
Value of x? 3
y = 4.0
Process finished with exit code 0
```

Рисунок 1 - Результат выполнения программы

Пример 2.

```
C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe F:/Programs/GitHub/Lab-2.1/Python/primer2.py
Введите номер месяца 8
Summer
```

Рисунок 2 - Результат выполнения программы

Пример 3.

```
C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe F:/Programs/GitHub/Lab-2.1/Python/primer3.py
Value of n? 2
Value of x? 1
S= 1.5465521559751236
```

Рисунок 3 - Результат выполнения программы

Пример 4.

```
C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe F:/Programs/GitHub/Lab-2.1/Python/primer4.py
Value of a? 0
x = 2.449489742783178
X = 2.449489742783178
```

Рисунки 4 - Результат выполнения программы

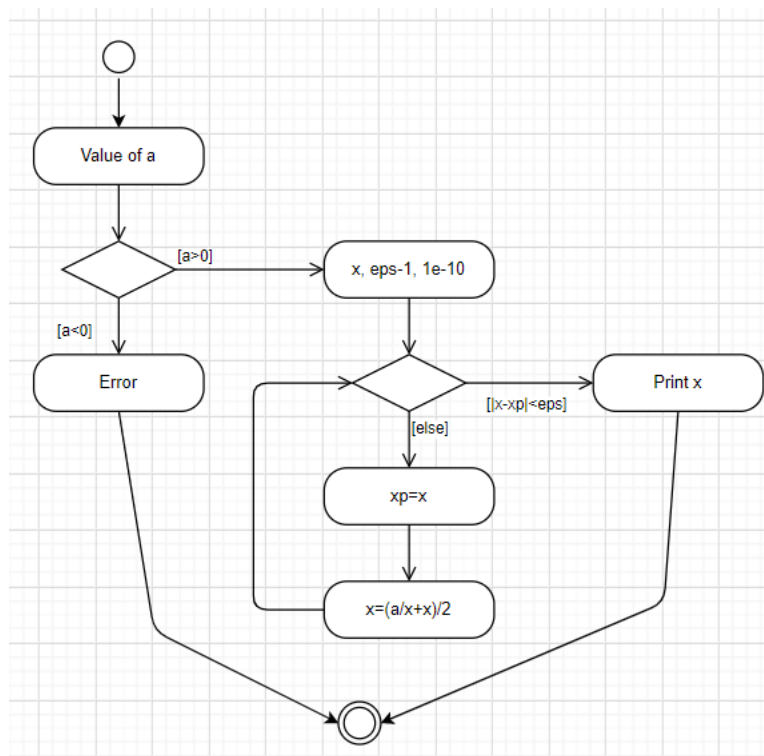


Рисунок 5 - UML-диаграмма

Пример 5.

```

C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe F:/Programs/GitHub/Lab-2.1/Python/primer5.py
Value of x? 7
Ei(7.0) = 191.50474333549477
  
```

Рисунок 6 - Результат выполнения программы

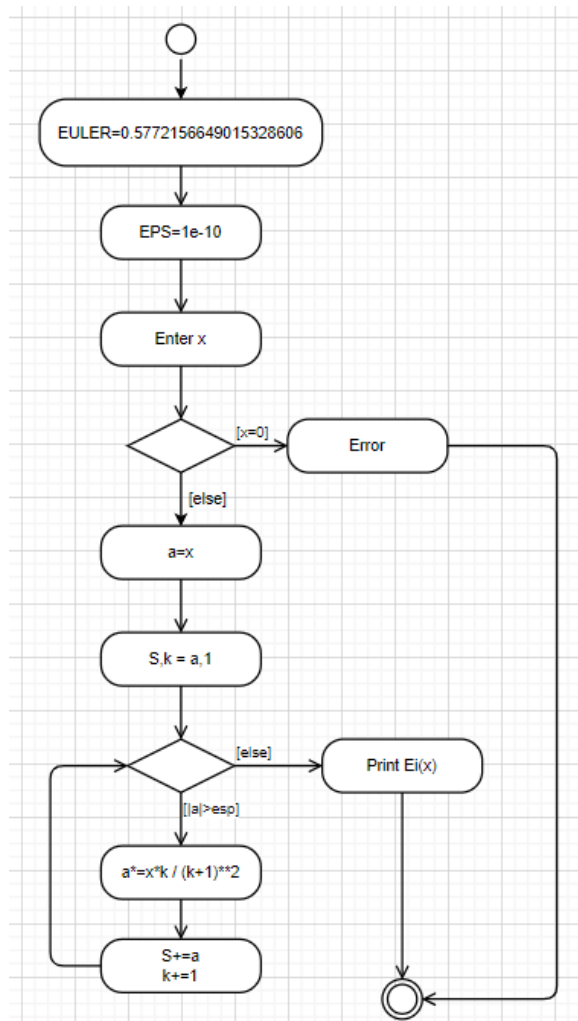


Рисунок 7 - UML-диаграмма

Индивидуальные задания:

Задание 1. С клавиатуры вводится цифра (от 1 до 12). Вывести на экран название месяца, соответствующего цифре.

```

C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe F:/Programs/GitHub/Lab-2.1/Python/individual.py
Введите номер месяца: 3
Март
  
```

Рисунок 8 - Результат выполнения программы

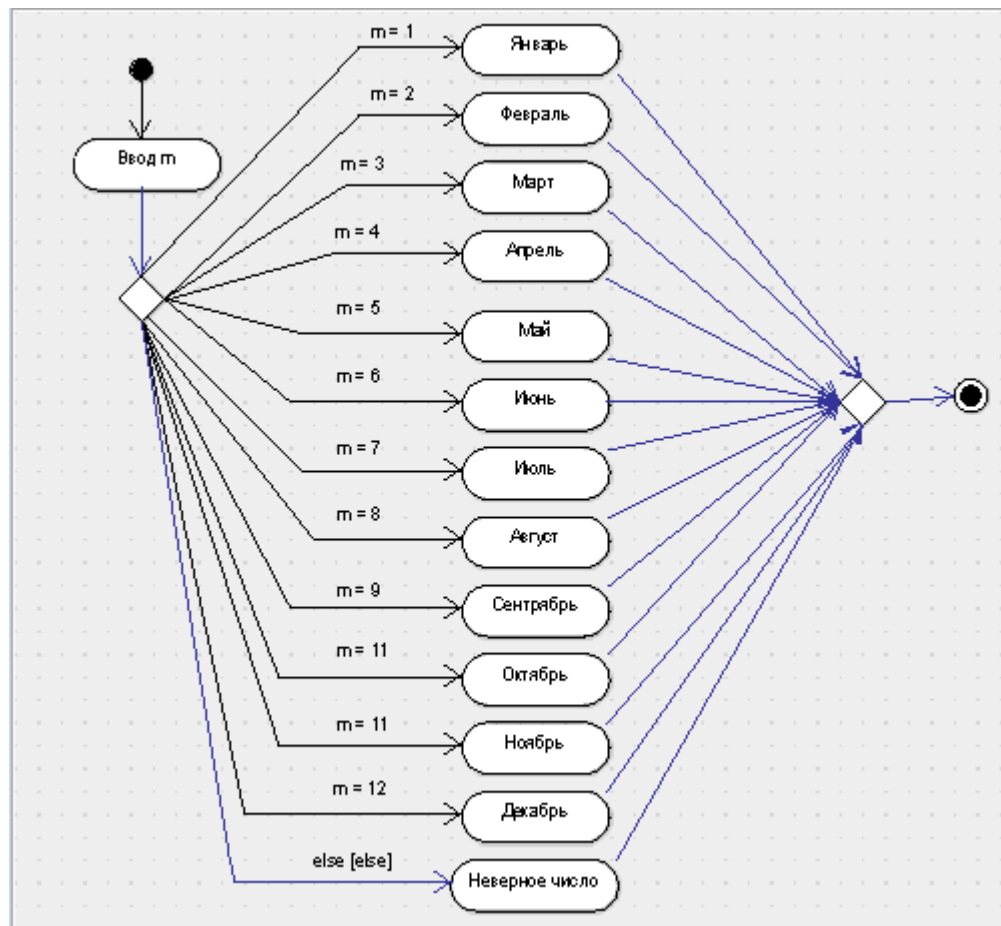


Рисунок 9 - UML-диаграмма

Задание 2. Определить, есть ли среди трёх заданных чисел нечётные.

```

Введите первое число: 15
Введите второе число: 10
Введите третье число: 12
Есть нечётное число
  
```

Рисунок 10 - Результат выполнения программы

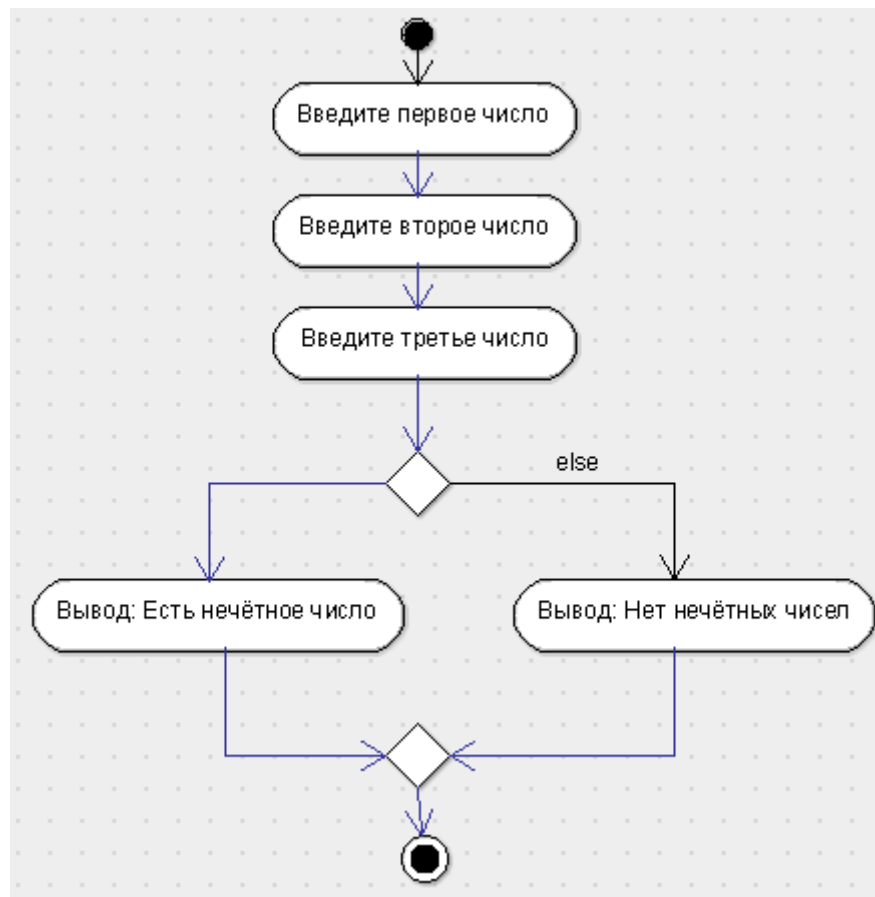


Рисунок 11 - UML-диаграмма

Задание 3. У гусей и кроликов вместе 64 лапы. Сколько могло быть кроликов и гусей (указать все сочетания, которые возможны).

```

0 Кроликов и 32 Гусей
1 Кроликов и 30 Гусей
2 Кроликов и 28 Гусей
3 Кроликов и 26 Гусей
4 Кроликов и 24 Гусей
5 Кроликов и 22 Гусей
6 Кроликов и 20 Гусей
7 Кроликов и 18 Гусей
8 Кроликов и 16 Гусей
9 Кроликов и 14 Гусей
10 Кроликов и 12 Гусей
11 Кроликов и 10 Гусей
12 Кроликов и 8 Гусей
13 Кроликов и 6 Гусей
14 Кроликов и 4 Гусей
15 Кроликов и 2 Гусей
16 Кроликов и 0 Гусей
  
```

Рисунок 12 - Результат выполнения программы

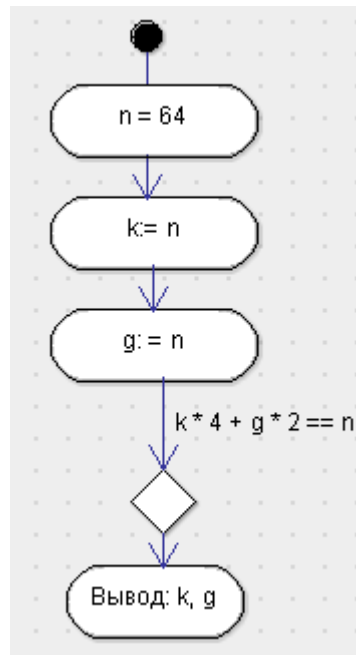


Рисунок 13 - UML-диаграмма

Задание повышенной сложности:

```

x = 28
Ci(28.0) = 1.425564543852677e+150

```

Рисунок 14 - Результат выполнения программы

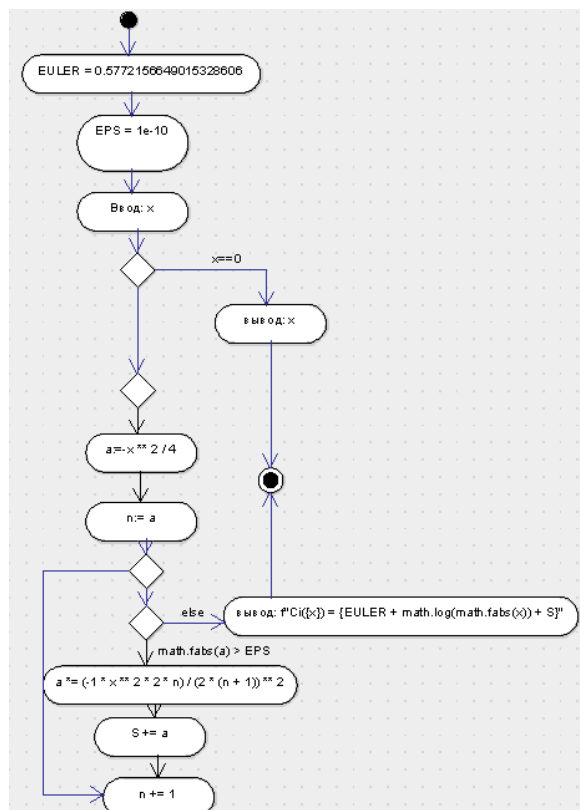


Рисунок 15 - UML-диаграмма

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

Контрольные вопросы:

1. Для чего нужны диаграммы деятельности UML?

Позволяет наглядно визуализировать алгоритм программы.

2. Что такое состояние действия и состояние деятельности?

Состояние действия - частный вид состояния деятельности, а конкретнее – такое состояние, которое не может быть подвергнуто дальнейшей декомпозиции.

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?

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: `a == b`

9. Что такое составное условие? Приведите примеры.

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

Пример: `(a == b or a == c)`

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 — это вид алгоритма, в процессе выполнения которого одно или несколько действий нужно повторить.

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.

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?

Используется для выхода из цикла.

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

Оператор continue используется только в циклах. В операторах for , while , do while , оператор continue выполняет пропуск оставшейся части кода тела цикла и переходит к следующей итерации цикла.

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.