

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

«Основы языка Python»

**Отчёт
по лабораторной работе №2.2
дисциплины
«Системы искусственного интеллекта»**

Выполнила:
Островская Ангелина Игоревна
2 курс, группа ППЖ-б-о-21-1
19.03.03 «Продукты питания животного происхождения»

(подпись)

Проверил:

(подпись)

Отчёт защищён с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.


Лабораторная работа 2.2 Условные операторы и циклы в языке Python

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

Ссылка на репозиторий: <https://github.com/OstrovskayAngelina/Angelina-Python-2.3.git>

Изучение теоретического материала

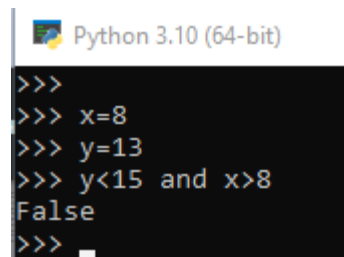
Логические операторы.



```
Python 3.10 (64-bit)
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022)
Type "help", "copyright", "credits" or "license()"
>>> a=10
>>> b=5
>>> a+b>14
True
>>> a<14-b
False
>>> a<=b+5
True
>>> a==b
False
>>> c=a==b
>>> a,b,c
(10, 5, False)
>>> _
```

Рисунок 1 – Использование в Python математических знаков

Допустим, переменной x было присвоено значение 8 ($x = 8$), переменной y присвоили 13 ($y = 13$). Логическое выражение $y < 15$ and $x > 8$ будет выполняться следующим образом. Сначала выполнится выражение $y < 15$. Его результатом будет True. Затем выполнится выражение $x > 8$. Его результатом будет False. Далее выражение сведется к True and False , что вернет False.



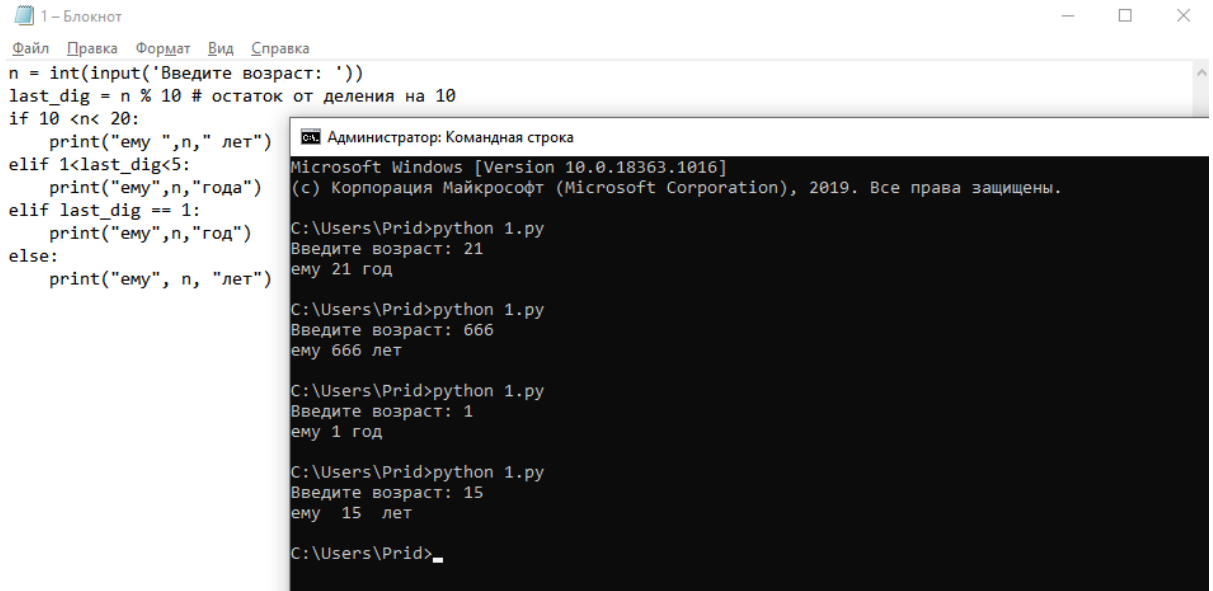
```
Python 3.10 (64-bit)
>>>
>>> x=8
>>> y=13
>>> y<15 and x>8
False
>>> _
```

Рисунок 1 – Использование в Python математических знаков

Практическая часть. Ход работы:

Задание 1.

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



```
1 - Блокнот
Файл  Правка  Формат  Вид  Справка
n = int(input('Введите возраст: '))
last_dig = n % 10 # остаток от деления на 10
if 10 < n < 20:
    print("ему ", n, " лет")
elif 1 < last_dig < 5:
    print("ему", n, "года")
elif last_dig == 1:
    print("ему", n, "год")
else:
    print("ему", n, "лет")

Администратор: Командная строка
Microsoft Windows [Version 10.0.18363.1016]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\Prid>python 1.py
Введите возраст: 21
ему 21 год

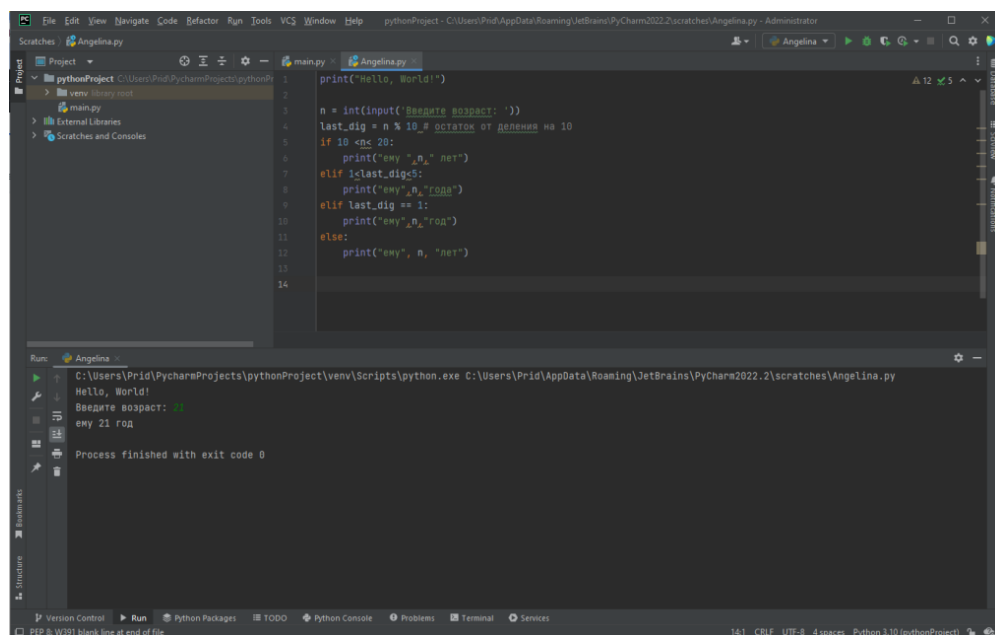
C:\Users\Prid>python 1.py
Введите возраст: 666
ему 666 лет

C:\Users\Prid>python 1.py
Введите возраст: 1
ему 1 год

C:\Users\Prid>python 1.py
Введите возраст: 15
ему 15 лет

C:\Users\Prid>
```

Рисунок 1 – Командная строка, запуск Python программы



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - C:\Users\Prid\AppData\Roaming\JetBrains\PyCharm2022.2\scratches\Angeline.py - Administrator
Scratches | Angeline.py
Project | pythonProject | C:\Users\Prid\PycharmProjects\pythonProject\venv\Scripts\python.exe
main.py | Angeline.py
1 print('hello, World!')
2
3 n = int(input('Введите возраст: '))
4 last_dig = n % 10 # остаток от деления на 10
5 if 10 < n < 20:
6     print("ему ", n, " лет")
7 elif 1 < last_dig < 5:
8     print("ему", n, "года")
9 elif last_dig == 1:
10    print("ему", n, "год")
11 else:
12    print("ему", n, "лет")
13
14

Run | Angeline |
C:\Users\Prid\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\Prid\AppData\Roaming\JetBrains\PyCharm2022.2\scratches\Angeline.py
Hello, World!
Введите возраст: 21
ему 21 год
Process finished with exit code 0

PEP 8: W391 blank line at end of file 141 CRLF UTF-8 4 spaces Python 3.10 (pythonProject)
```

Рисунок 2 – Программа PyCharm, запуск кода Python

3. Дано число (). Вывести на экран название дня недели, который соответствует этому номеру.

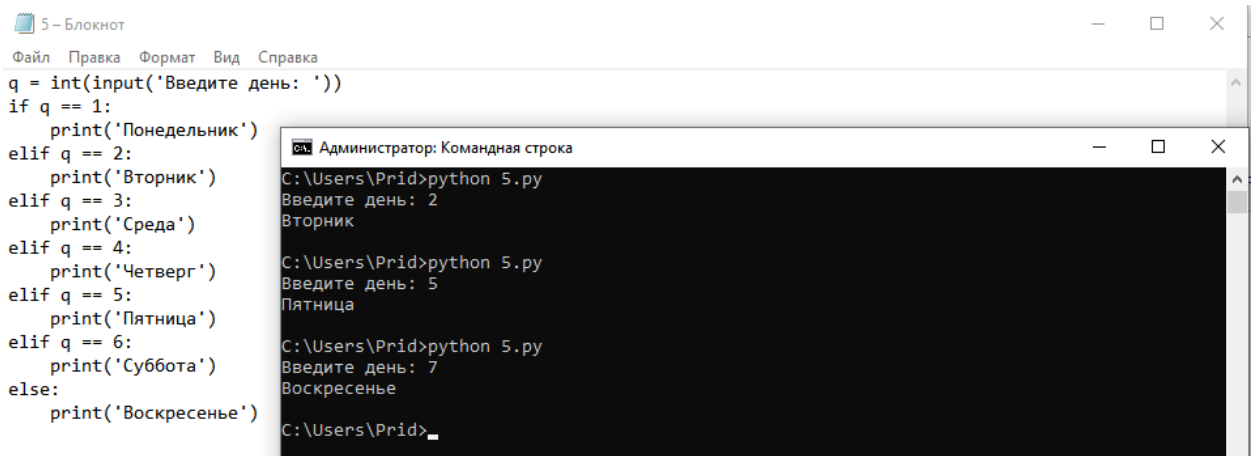


Рисунок 3 - Открытие файла 5.py

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

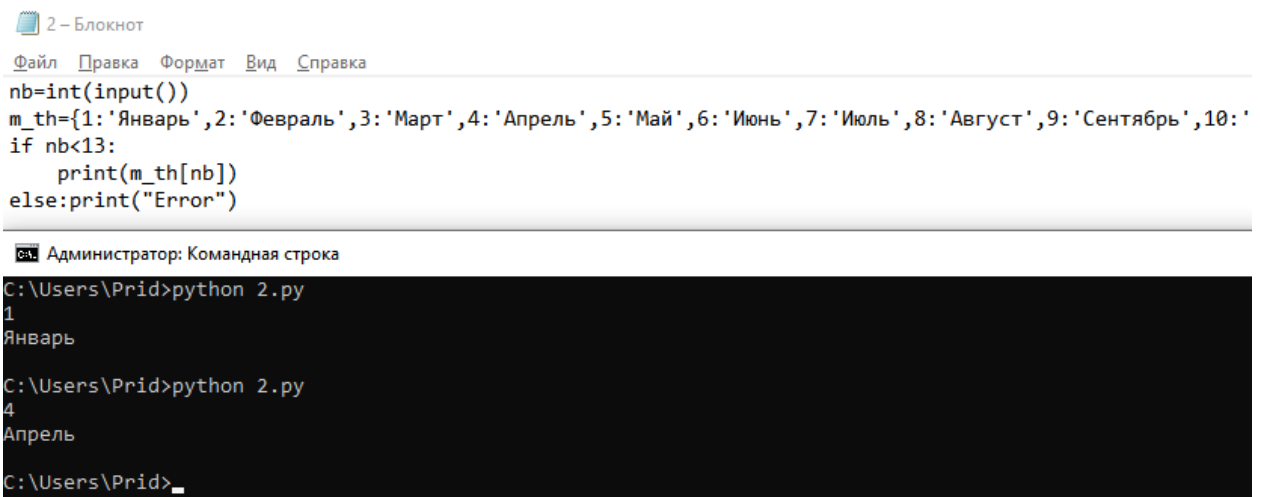


Рисунок 4 – Открытие файла 2.py

Задание 2.

14. Составить программу нахождения из трех чисел наибольшего и наименьшего

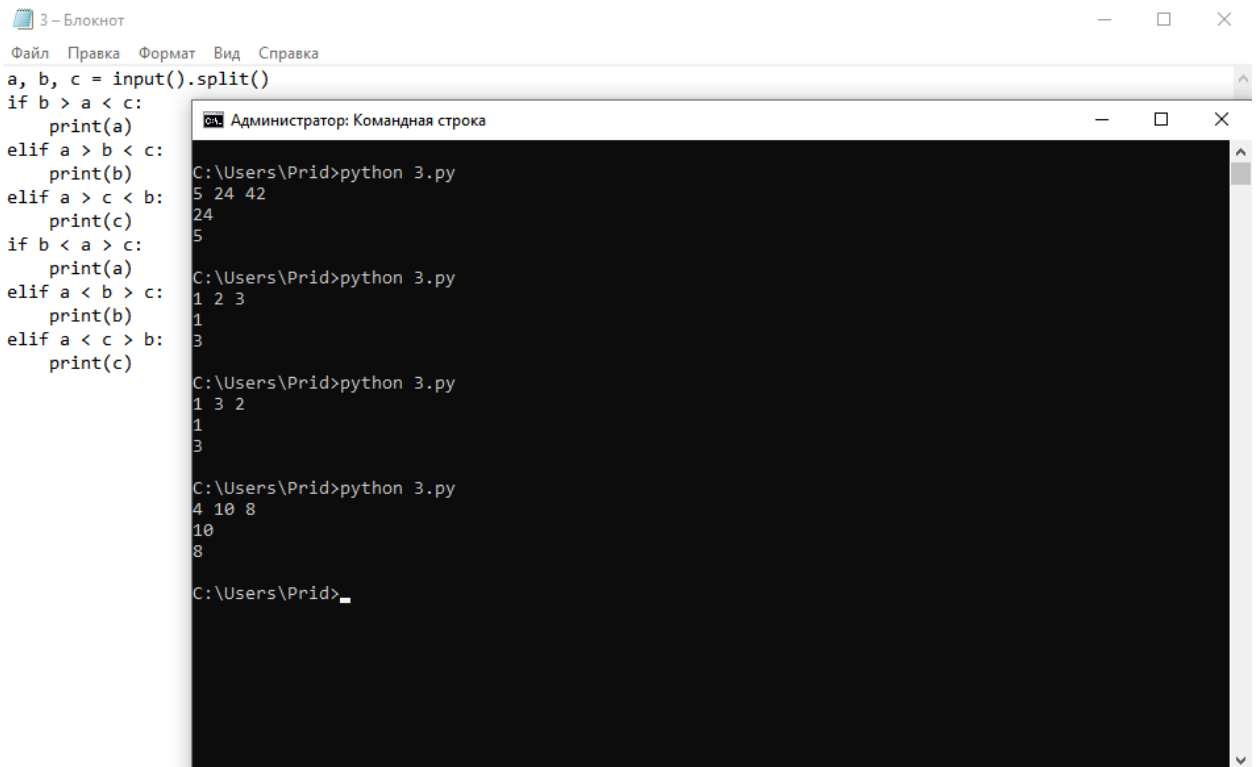


Рисунок 5 – Открытие файла 3.py

17. Определить, есть ли среди трёх заданных чисел нечётные.

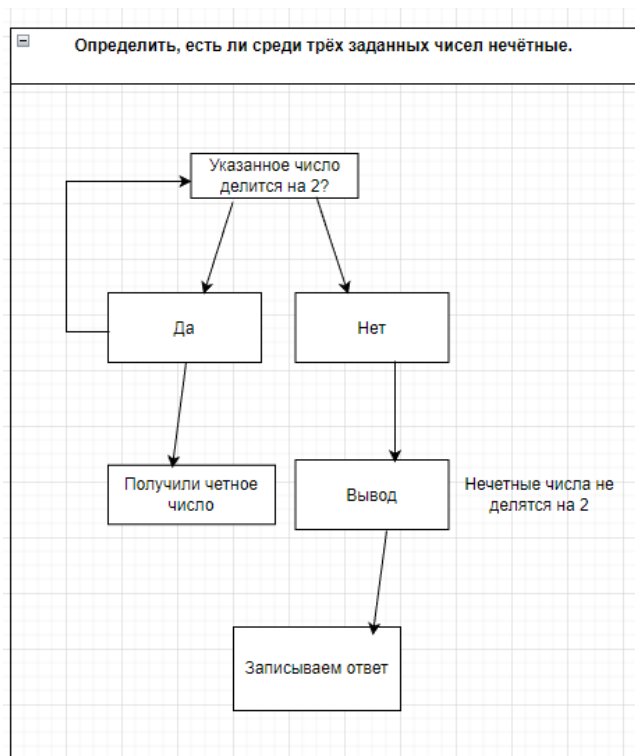


Рисунок 4 – Диаграмма для файла 4.py

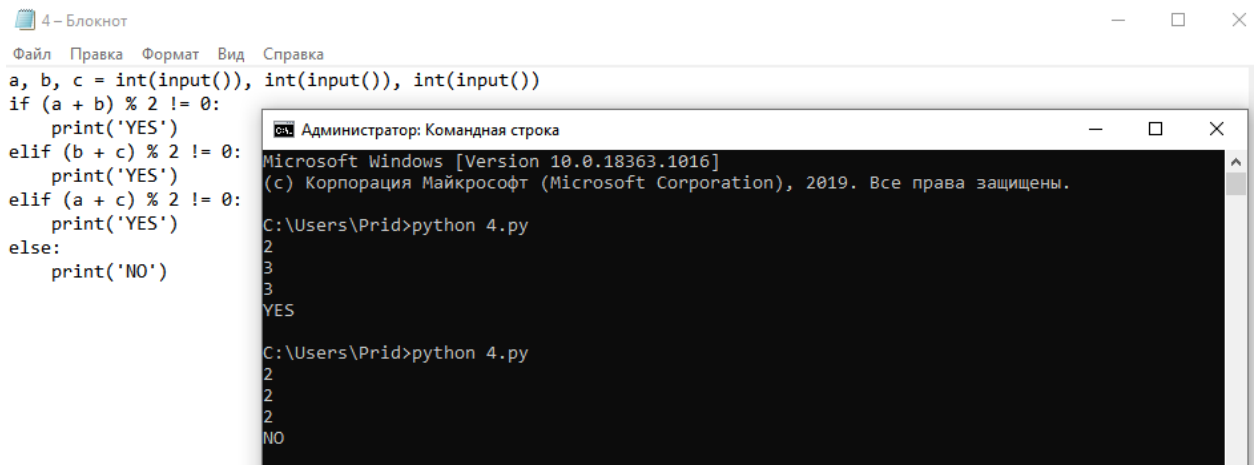


Рисунок 6 – Открытие файла 4.py при помощи cmd

1. Для чего нужны диаграммы деятельности UML?
2. Что такое состояние действия и состояние деятельности?
3. Какие нотации существуют для обозначения переходов и ветвлений в диаграммах деятельности?
4. Какой алгоритм является алгоритмом разветвляющейся структуры?
5. Чем отличается разветвляющийся алгоритм от линейного?
6. Что такое условный оператор? Какие существуют его формы?
7. Какие операторы сравнения используются в Python?
8. Что называется простым условием? Приведите примеры.
9. Что такое составное условие? Приведите примеры.
10. Какие логические операторы допускаются при составлении сложных условий?
11. Может ли оператор ветвления содержать внутри себя другие ветвления?
12. Какой алгоритм является алгоритмом циклической структуры?
13. Типы циклов в языке Python.
14. Назовите назначение и способы применения функции range .
15. Как с помощью функции range организовать перебор значений от 15 до 0 с шагом 2?
16. Могут ли быть циклы вложенными?
17. Как образуется бесконечный цикл и как выйти из него?
18. Для чего нужен оператор break ?
19. Где употребляется оператор continue и для чего он используется?
20. Для чего нужны стандартные потоки stdout и stderr?
21. Как в Python организовать вывод в стандартный поток stderr?
22. Каково назначение функции exit ?

Ответы на вопросы:

1. Унифицированный язык моделирования (UML) является стандартным инструментом для создания «чертежей» программного обеспечения. 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.
2. 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.

4. Алгоритм разветвляющейся структуры - это алгоритм, в котором вычислительный процесс осуществляется либо по одной, либо по другой ветви, в зависимости от выполнения некоторого условия. Программа разветвляющейся структуры реализует такой алгоритм. 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

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

7. Говоря на естественном языке (например, русском) мы обозначаем сравнения словами "равно", "больше", "меньше". В языках программирования используются специальные знаки, подобные тем, которые используются в математике: `>` (больше), `<` (меньше), `>=` (больше или равно), `<=` (меньше или равно), `==` (равно), `!=` (не равно).

8. Простое условие - **два выражения, связанные одним из знаков отношений: `=` (равно) `>` (больше) `<` (меньше) `>=` (больше либо равно) `<=` (меньше либо равно) `<>` (не равно)**. Примеры: `a > 0`; `a + c <= 0`; `x <> 0`; `a <= d`. Составные условия - условия, состоящие из двух или более простых условий, соединенных с помощью логических операций `and` (и), `or` (или), `not` (не). Простые условия при этом заключаются в круглые скобки.

9. **Примеры:** `a > 0`; `a + c <= 0`; `x <> 0`; `a <= d` **Составные условия - условия**, состоящие из двух или более простых **условий**, соединенных с помощью логических операций `and` (и), `or` (или), `not` (не).

10. Допустим, переменной `x` было присвоено значение 8 (`x = 8`), переменной `y` присвоили 13 (`y = 13`). Логическое выражение `y < 15 and x > 8` будет выполняться следующим образом. Сначала выполнится выражение `y < 15`. Его результатом будет `True`. Затем выполнится выражение `x > 8`. Его результатом будет `False`. Далее выражение сведется к `True and False`, что вернет `False`.

11. Да. Простые последовательные переходы встречаются наиболее часто, но их одних недостаточно для моделирования любого потока управления. В точку ветвления может входить ровно один переход, а выходить - два или более.

12. Алгоритм циклической структуры - это алгоритм, в котором происходит многократное повторение одного и того же участка программы. Такие повторяемые участки вычислительного процесса называются циклами. Операторы цикла используются для организации многократно повторяющихся вычислений. Любой цикл состоит из тела цикла, то есть тех операторов, которые выполняются несколько раз, начальных установок, модификации параметра цикла и проверки условия продолжения выполнения цикла.

13. В Python есть два типа циклов: **`for`** и **`while`**. В отличие от многих си-подобных языков и от привычного многим Pascal, в Python нет аналога цикла `do-while` или `repeat-until`.

14. Функция `range` возвращает неизменяемую последовательность чисел в виде объекта `range`. Параметры функции: `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

```
>>> list(range(10, 0, -2))
[10, 8, 6, 4, 2]
```

15.

16. Вложенные циклы – это циклы, организованные в теле другого цикла. Вложенный цикл в тело другого цикла, называется внутренним циклом. Цикл, в теле которого существует вложенный цикл, называется внешним.

17. **Цикл** — это управляющая конструкция, которая раз за разом выполняет серию команд (тело цикла) до тех пор, пока условие для выполнения является истинным.

В большинстве случаев, бесконечные циклы появляются из-за логических ошибок программиста (например, когда условие цикла `while` при любых вариантах равно `True`). Поэтому следует внимательно следить за условием, при котором цикл будет завершаться. Оператор `break` предназначен для досрочного прерывания работы цикла `while`.

18. Оператор `break` предназначен для досрочного прерывания работы цикла `while`.

19. Оператор `continue` запускает цикл заново, при этом код, расположенный после данного оператора, не выполняется

20. В операционной системе по умолчанию присутствуют стандартных потока вывода на консоль: буферизованный поток `stdout` для вывода данных и информационных сообщений.

21. Небуферизованный поток `stderr` для вывода сообщений об ошибках. По умолчанию функция `print` использует поток `stdout`.

22. В Python завершить программу и передать операционной системе заданный код возврата можно посредством функции `exit`.