

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Плещенко Данила Георгиевич
1 курс, группа ИТС-б-о-21-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук, доцент
кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

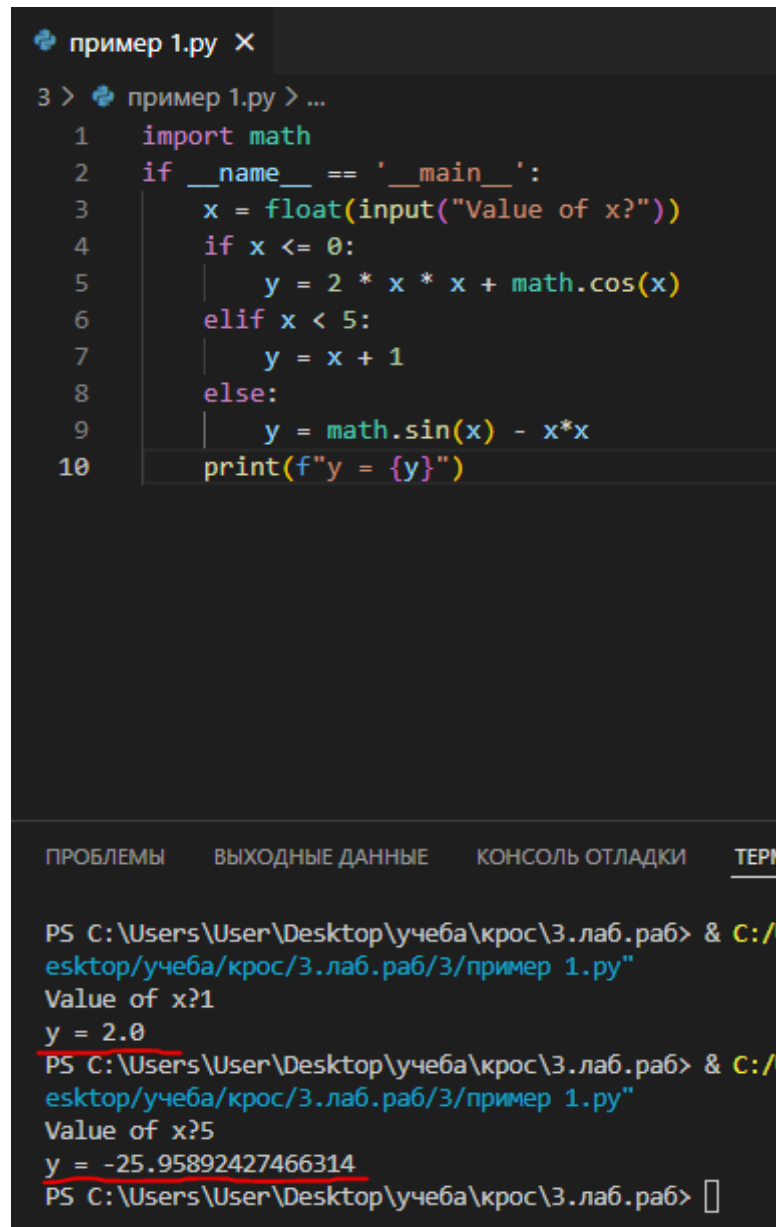
Тема: Условные операторы и циклы в языке Python

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.

Ход работы

Создал новый репозиторий - [danilaple/3.lab.rab \(github.com\)](https://github.com/danilaple/3.lab.rab) и начал проверку примеров.

Пример 1:



```

3 > пример 1.py X
1  import math
2  if __name__ == '__main__':
3      x = float(input("Value of x?"))
4      if x <= 0:
5          y = 2 * x * x + math.cos(x)
6      elif x < 5:
7          y = x + 1
8      else:
9          y = math.sin(x) - x*x
10     print(f"y = {y}")

ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ

PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/Desktop/учеба/крос/3.лаб.раб/3/пример 1.py
Value of x?1
y = 2.0
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/Desktop/учеба/крос/3.лаб.раб/3/пример 1.py
Value of x?5
y = -25.95892427466314
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб>

```

Рисунок 1. Пример работы программы из «Пример 1»

Пример 2:

```
3 > пример 2.py X
1 import sys
2 if __name__ == '__main__':
3     n = int(input("Введите номер месяца: "))
4     if n == 1 or n == 2 or n == 12:
5         print("Зима")
6     elif n == 3 or n == 4 or n == 5:
7         print("Весна")
8     elif n == 6 or n == 7 or n == 8:
9         print("Лето")
10    elif n == 9 or n == 10 or n == 11:
11        print("Осень")
12    else:
13        print("Ошибка!", file=sys.stderr)
14        exit(1)
```

ПРОБЛЕМЫ Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```
esktop/учеба/крос/3.лаб.раб/3/пример 2.py"
Введите номер месяца: 1
Зима
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/Desktop/учеба/крос/3.лаб.раб/3/пример 2.py"
Введите номер месяца: 6
Лето
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/Desktop/учеба/крос/3.лаб.раб/3/пример 2.py"
Введите номер месяца: 15
Ошибка!
```

Рисунок 2. Пример работы программы из «Пример 2»

Пример 3:

```
3 > пример 3.py ...
1 import math
2 if __name__ == '__main__':
3     n = int(input("Value of n? "))
4     x = float(input("Value of x? "))
5     S = 0.0
6     for k in range(1, n + 1):
7         a = math.log(k * x) / (k * k)
8         S += a
9     print(f"S = {S}")
```

ПРОБЛЕМЫ Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/Desktop/учеба/крос/3.лаб.раб/3/пример 3.py"
Value of n? 1
Value of x? 2
S = 0.6931471805599453
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/Desktop/учеба/крос/3.лаб.раб/3/пример 3.py"
Value of n? 3
Value of x? 4
S = 2.1822554854051828
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб>
```

Рисунок 3. Пример работы программы из «Пример 3»

Пример 4:

```
3 > пример 4.py > ...
1 import math
2 import sys
3 if __name__ == '__main__':
4     a = float(input("Value of a? "))
5     if a < 0:
6         print("Illegal value of a", file=sys.stderr)
7         exit(1)
8     x, eps = 1, 1e-10
9     while True:
10         xp = x
11         x = (x + a / x) / 2
12         if math.fabs(x - xp) < eps:
13             break
14     print(f"x = {x}\nX = {math.sqrt(a)}")
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/AppData/Local/Programs/Python/Python39-64/Python.exe C:\Users\User\Desktop\учеба\крос\3.лаб.раб\3\пример 4.py
Value of a? 12
x = 3.4641016151377544
X = 3.4641016151377544
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> █

Рисунок 4. Пример работы программы из «Пример 4»

Пример 5:

```
3 > пример 5.py > ...
4 EULER = 0.5772156649015328606
5 # Точность вычислений.
6 EPS = 1e-10
7
8 if __name__ == '__main__':
9     x = float(input("Value of x? "))
10    if x == 0:
11        print("Illegal value of x", file=sys.stderr)
12        exit(1)
13    a = x
14    S, k = a, 1
15    # Найти сумму членов ряда.
16    while math.fabs(a) > EPS:
17        a *= x * k / (k + 1) ** 2
18        S += a
19        k += 1
20    # Вывести значение функции.
21    print(f"Ei({x}) = {EULER + math.log(math.fabs(x)) + S}")
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/AppData/Local/Programs/Python/Python39-64/Python.exe C:\Users\User\Desktop\учеба\крос\3.лаб.раб\3\пример 5.py
Value of x? 14
Ei(14.0) = 93192.51363396535
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> █

Рисунок 5. Пример работы программы из «Пример 5»

Индивидуальное задание 1: С клавиатуры вводится натуральное число . В зависимости от значения остатка при делении числа на 7 вывести на экран число в виде $n = 7*k + r$. Если , то вывести на экран $n = 7*k$.

```
индив.1.py > ...
1  n = int(input("Value of n? "))
2  k = n // 7
3  r = n % 7
4  if r != 0:
5      a = 7*k + r
6      print("(n=7*k + r)", "Ответ:", a)
7  elif r == 0:
8      b = 7*k
9      print("(n=7*k)", "Ответ:", b)
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/Desktop/учеба/крос/3.лаб.раб/индив.1.py
Value of n? 13
(n=7*k + r) Ответ: 13
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/Desktop/учеба/крос/3.лаб.раб/индив.1.py
Value of n? 7
(n=7*k) Ответ: 7
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> █
```

Рисунок 6. Решение на 1-е индивидуальное задание

Индивидуальное задание 2: Определить, есть ли среди трёх заданных чисел нечётные.

```
индив.2.py > ...
1  a = int(input())
2  if (a % 2 == 0):
3      print(a, "чётное")
4  else:
5      print(a, "нечётное")
6  b = int(input())
7  if (b % 2 == 0):
8      print(b, "чётное")
9  else:
10     print(b, "нечётное")
11  c = int(input())
12  if (c % 2 == 0):
13     print(c, "чётное")
14  else:
15     print(c, "нечётное")
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> & C:/Users/User/Desktop/учеба/крос/3.лаб.раб/индив.2.py
1  нечётное
2  чётное
3  нечётное
PS C:\Users\User\Desktop\учеба\крос\3.лаб.раб> █
```

Рисунок 7. Решение на 2-е индивидуальное задание

Индивидуальное задание 3: Составьте программу, которая печатает таблицу сложения натуральных чисел в десятичной системе счисления.

```
индив.3.py > ...
1 print("Таблица сложения: ")
2 for i in range(1,10):
3     for j in range(1,10):
4         print(i,"+",j,"=",i+j)
```

ПРОБЛЕМЫ	ВЫХОДНЫЕ ДАННЫЕ	КОНСОЛЬ ОТЛАДКИ
6 + 4 = 10		
6 + 5 = 11		
6 + 6 = 12		
6 + 7 = 13		
6 + 8 = 14		
6 + 9 = 15		
7 + 1 = 8		
7 + 2 = 9		
7 + 3 = 10		
7 + 4 = 11		
7 + 5 = 12		
7 + 6 = 13		

Рисунок 8. Решение на 3-е индивидуальное задание

UML-диаграммы:

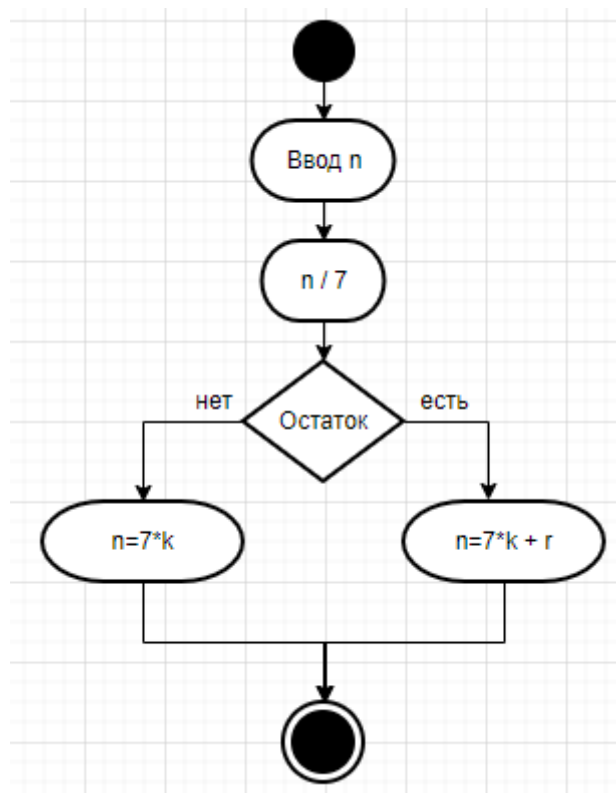


Рисунок 9. UML-диаграмма на индивидуальное задание 1

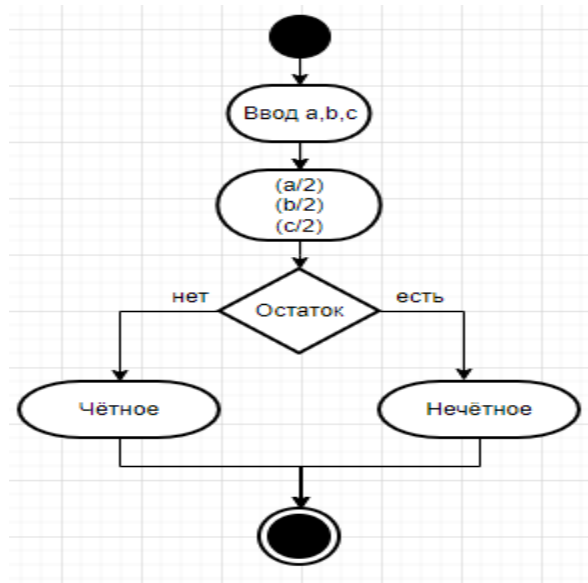


Рисунок 10. UML-диаграмма на индивидуальное задание 2

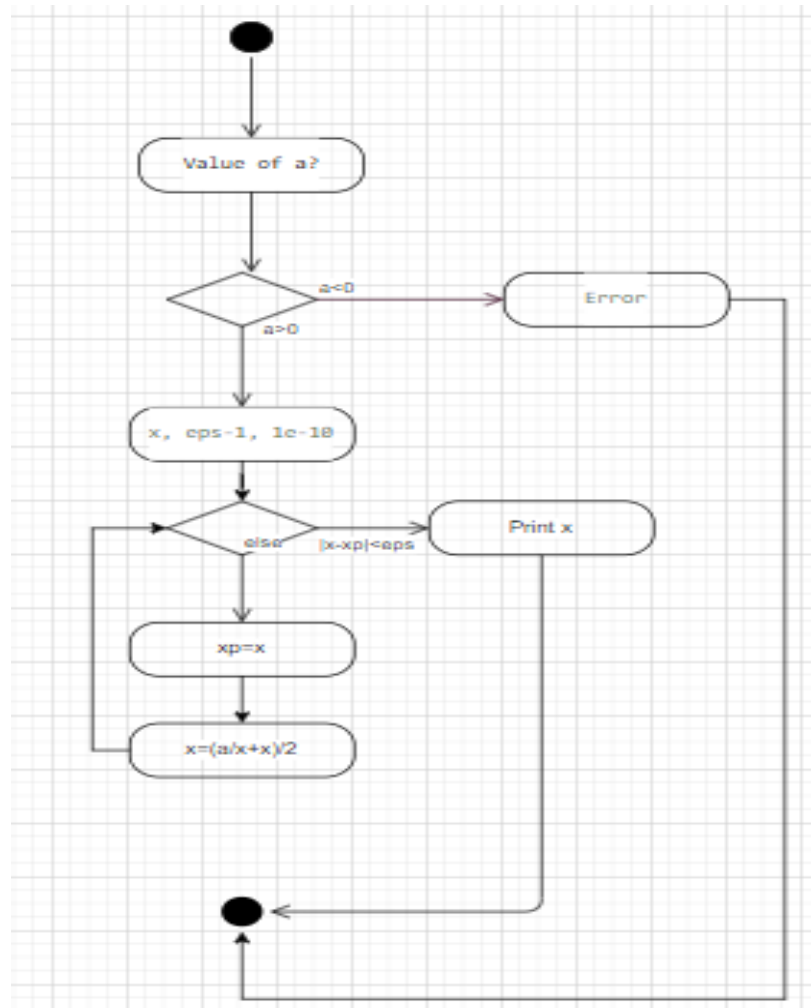


Рисунок 11. UML-диаграмма на «Пример 4»

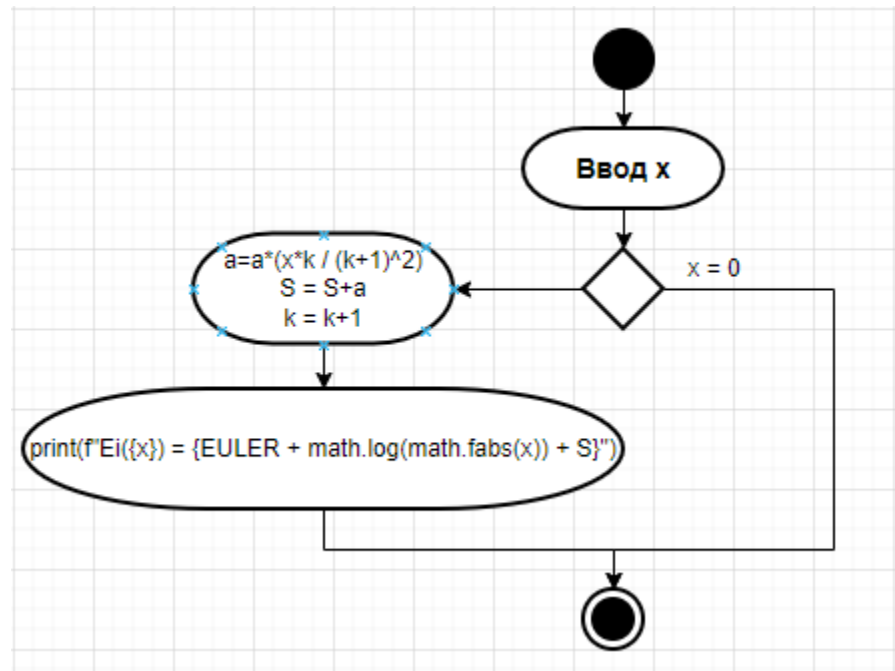


Рисунок 12. UML-диаграмма на «Пример 5»

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

Контрольные вопросы

1. Для чего нужны диаграммы деятельности UML?

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

UML можно визуализировать, специфицировать, конструировать и документировать артефакты программных систем.

2. Что такое состояние действия и состояние деятельности?

Также, например, можно выполнить операцию над объектом, послать ему сигнал или даже создать его или уничтожить. Все эти выполняемые атомарные вычисления называются состояниями действия.

Состояния деятельности не являются атомарными, то есть могут быть прерваны. Можно считать, что состояние действия - это частный вид

состояния деятельности, а конкретнее – такое состояние, которое не может быть подвергнуто дальнейшей декомпозиции.

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

В UML переход представляется простой линией со стрелкой, 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.

5. Чем отличается разветвляющийся алгоритм от линейного?

В программе разветвляющейся структуры имеется один или несколько условных операторов.

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

Условный оператор ветвления if

Оператор ветвления if позволяет выполнить определенный набор инструкций в зависимости от некоторого условия.

- 1) Конструкция if;
- 2) Конструкция if – else;
- 3) Конструкция if – elif – else;

7. Какие операторы сравнения используются в Python?

В языках программирования используются специальные знаки, подобные тем, которые используются в математике: > (больше), < (меньше), >= (больше или равно), <= (меньше или равно), == (равно), != (не равно).

8. Что называется простым условием? Приведите примеры.

Логические выражения типа (kByte >= 1023) являются простыми, так как в них выполняется только одна логическая операция.

9. Что такое составное условие? Приведите примеры.

Простые инструкции описываются одной строкой когда, составные же – содержат вложенные инструкции.

```
if x <= 0:
```

```
y = 2 * x * x + math.cos(x)
```

```
elif x < 5:
```

```
y = x + 1
```

10. Какие логические операторы допускаются при составлении сложных условий?

В таких случаях используются специальные операторы, объединяющие два и более простых логических выражения. Широко используются два оператора – так называемые логические И (and) и ИЛИ (or).

12. Какой алгоритм является алгоритмом циклической структуры?

Алгоритм циклической структуры - это алгоритм, в котором происходит многократное повторение одного и того же участка программы. Такие повторяемые участки вычислительного процесса называются циклами.

13. Типы циклов в языке Python.

Оператор цикла while

Операторы break и continue

Оператор цикла for

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.

step - с каким шагом растут числа. По умолчанию - 1

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?

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

```
a = 0
```

```
while a >= 0:
```

```
if a == 7:  
    break  
a += 1  
print("A")
```

В приведенном выше коде, выход из цикла произойдет при достижении переменной `a` значения 7. Если бы не было этого условия, то цикл выполнялся бы бесконечно.

18. Для чего нужен оператор `break` ?

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

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

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

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

В операционной системе по умолчанию присутствуют стандартных потока вывода на консоль: буферизованный поток `stdout` для вывода данных и информационных сообщений, а также не буферизованный поток `stderr` для вывода сообщений об ошибках.

21. Как в Python организовать вывод в стандартный поток `stderr`?

Для того, чтобы использовать поток `stderr` необходимо передать его в параметре `file` функции `print`.

22. Каково назначение функции `exit` ?

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