

Ссылка на урок: <https://coreapp.ai/app/player/lesson/67d50bf32b435e7049d17e2a>
(для учеников)

Занятие 11

Тема: Реализация циклических алгоритмов на ЯП Python. Цикл **while**. Часть 2

1. Теоретическая часть

На этом занятии мы рассмотрим ещё одну разновидность цикла – **цикл с известным условием окончания работы (с постусловием)**. Для его реализации также используется оператор **while**, с которым мы работали на прошлом занятии. Так как, в отличие от многих других ЯП (например, Pascal), в Питоне нет отдельного оператора цикла с постусловием.

Цикл с постусловием в Python реализуется с помощью бесконечного цикла с предусловием.

Полный формат данного цикла:

while True:

 <операторы>

if <условие>: **break**

Цикл **while True:**

 <операторы>

будет выполняться бесконечно, потому что условие **True** всегда истинно. Выйти из такого цикла можно только с помощью специального оператора **break** (в переводе с англ. — «прервать»), который выполняется, если условие окончания работы цикла становится истинным: **if** <условие>: **break**.

Рассмотрим задачу-игру «Угадай число», в которой пользователь должен угадать число, загаданное компьютером в определённом диапазоне, не более чем за 5 попыток.

Подсказка. При отгадывании числа используйте метод половинного деления (интервал всегда делится пополам).

Программа:

```
import random #загрузка библиотеки случайных чисел
number = random.randint(1, 25) #выбор компьютером случайного числа
k = 0 #создаём переменную k для подсчёта количества попыток
while k < 5: #организовываем цикл
    print('Угадай число между 1 and 25:') #предлагается пользо-
    вателю ввести число
    guess = input()
    guess = int(guess) #число должно быть целым
    k = k + 1 #с каждой попыткой счет увеличивается на 1
    if guess < number: #если число пользователя меньше загаданного
        print('Мое число больше твоего' )
    if guess > number: #если число пользователя больше загаданного
        print('Мое число меньше твоего' )
    if guess == number: #если число пользователя равно загаданному
        break #остановить программу
```

```

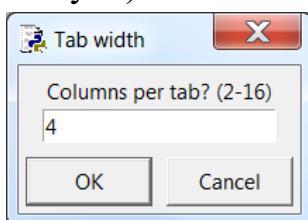
if guess == number:
    print('Молодец! Ты угадал число с ' + str(k) + ' попытки!' )
else:
    print('К сожалению, ты не угадал число. Я загадал ' +
str(number))

```

Возможная ошибка при наборе текста программы:

inconsistent indentation detected (обнаружен непоследовательный отступ). С виду вроде бы всё нормально, а системе не нравится. Делаем так:

1. Edit → Select All (Выделить всё).
2. Format → Untabify Region.
3. Устанавливаем стандартную ширину табуляции (4 пробела для каждого уровня отступа):



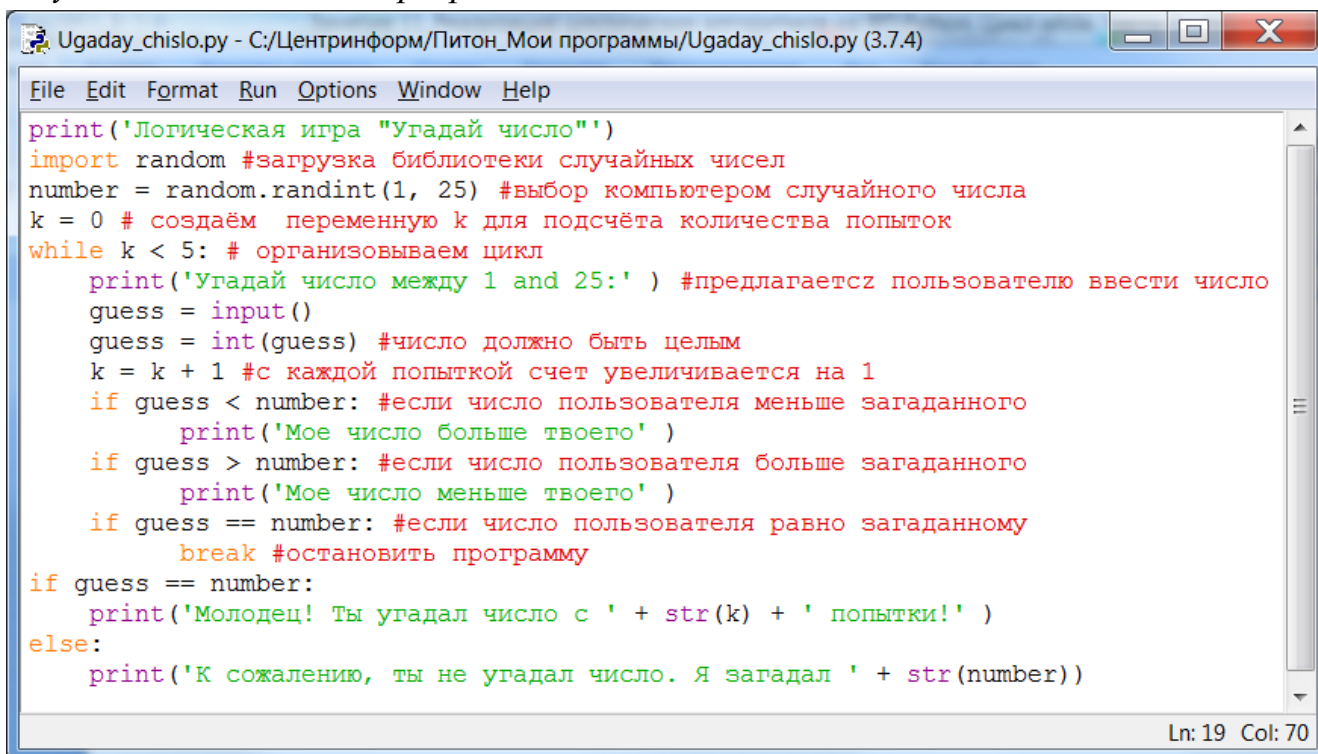
Ошибка *Untabify Region* (переводится как нетабуфицированная область) в программировании на Python возникает из-за несоответствия в использовании пробелов и табуляции для отступов.

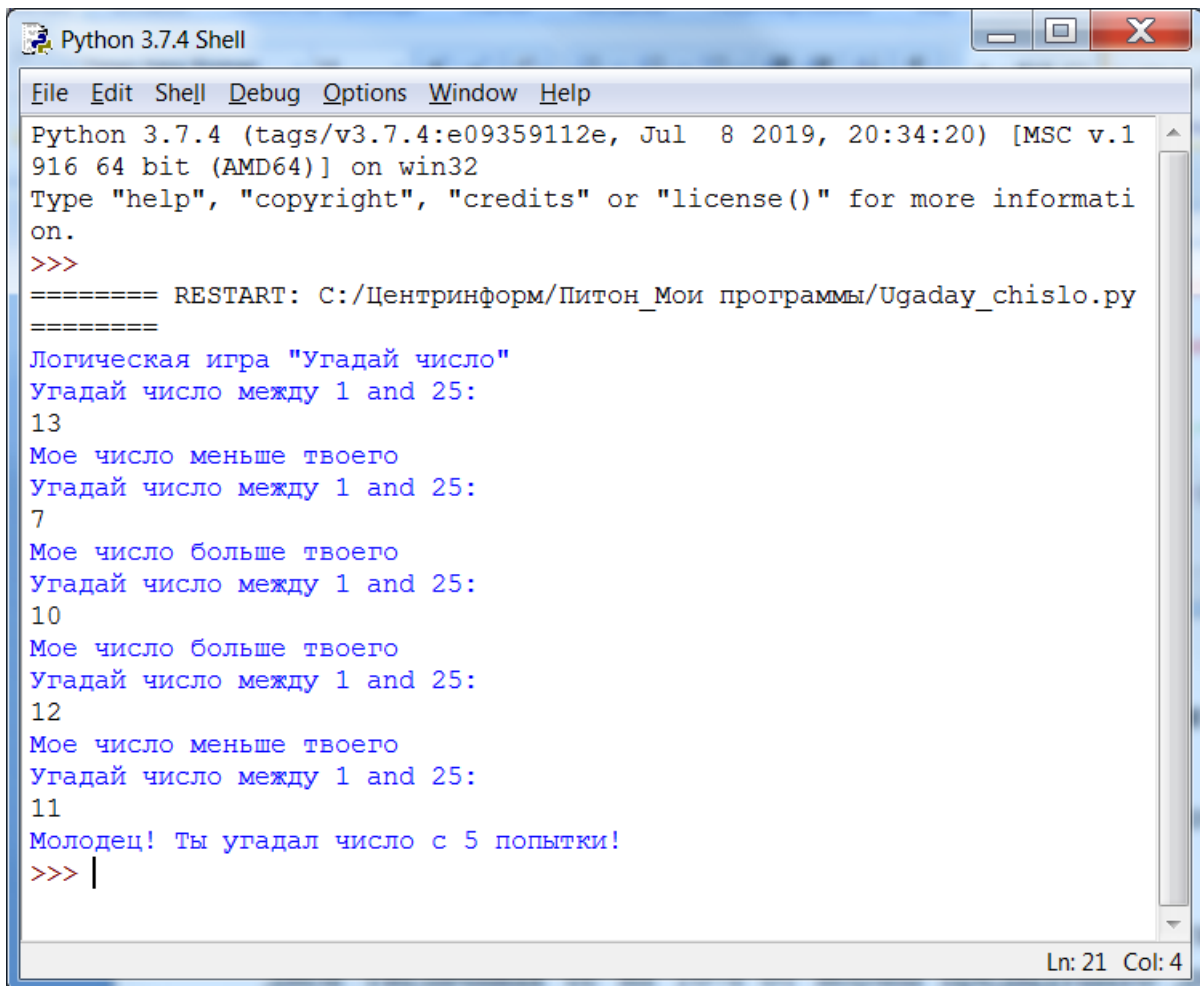
В Python 3.2 и более новых версиях рекомендуется использовать только пробелы для отступов. Если в программе смешивать пробелы и табуляцию, Python может выдать ошибку.

Функция Untabify Region в редакторе IDLE позволяет преобразовать отступы в уже существующих файлах. [1](#)

Чтобы исправить ошибку, нужно привести код к *единообразию*: использовать либо пробелы, либо табуляцию. В соответствии с рекомендациями стандарта оформления кода PEP 8, предпочтительно использовать 4 пробела для каждого уровня отступа.

Результат выполнения программы:



A screenshot of a Python 3.7.4 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following output:

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Центринформ/Питон_Мои программы/Ugaday_chislo.py
=====
Логическая игра "Угадай число"
Угадай число между 1 and 25:
13
Мое число меньше твоего
Угадай число между 1 and 25:
7
Мое число больше твоего
Угадай число между 1 and 25:
10
Мое число больше твоего
Угадай число между 1 and 25:
12
Мое число меньше твоего
Угадай число между 1 and 25:
11
Молодец! Ты угадал число с 5 попытки!
>>> |
```

The status bar at the bottom right indicates 'Ln: 21 Col: 4'.

2. Примеры программ с использованием оператора цикла while

Рассмотрим некоторые характерные примеры использования оператора цикла while в циклах с постусловием и с предусловием.

Задача 1

На тренировке спортсмен ежедневно пробегает некоторую дистанцию, с каждым днём увеличивая её на 10% от нормы предыдущего дня. Как только дневная норма достигнет или превысит 25 км, необходимо прекратить её увеличение и далее пробегать ежедневно ровно 25 км. Написать программу, определяющую, начиная с какого дня, спортсмен будет пробегать 25 км.

Алгоритм решения данной задачи представлен в виде блок-схемы на рисунке 1.

Пусть x – количество километров (дистанция), которое спортсмен пробежит в некоторый i -день. Тогда в следующий $(i + 1)$ -й день он пробежит $x + 0,1x$ километров ($0,1x$ – это 10% от x). Организуем цикл, в котором подсчёт количества дней и величины преодолённой спортсменом дистанции будет осуществляться до тех пор, пока количество километров не достигнет или превысит значение 25.

Как только условие $x \geq 25$ становится истинным, цикл прерывается и на экран выводится значение i .

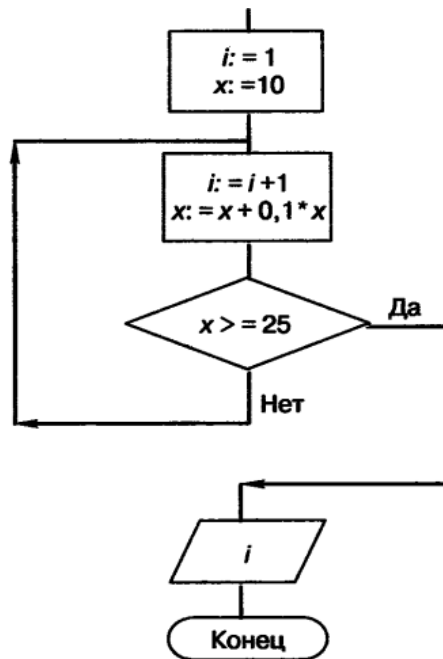


Рис. 1. Блок-схема задачи о графике тренировок спортсмена

Решение:

```
print(' График тренировок')
i = 1; x = 10
```

```
while True:
```

```
    i += 1
```

```
    x=x+0.1*x
```

```
    if x >= 25: break
```

```
print ('Начиная с ', i, ' -го дня спортсмен будет пробегать  
25 км')
```

Результат выполнения программы:

The screenshot shows a Python IDE window titled 'Trenirovki_beg_1.py'. The code in the editor is as follows:

```

print(' График тренировок')
i = 1; x = 10
while True:
    i += 1
    x = float(x+0.1*x)
    if x>= 25: break
print ('Начиная с ', i, ' -го дня спортсмен будет пробегать 25 км')
  
```

Below the editor is a 'Python 3.7.4 Shell' window showing the execution output:

```

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.
1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informat
ion.
>>>
==== RESTART: F:/Центринформ/Мои программы на Питоне/Trenirovki_beg_
1.py ====
 График тренировок
Начиная с  11  -го дня спортсмен будет пробегать 25 км
>>>
  
```

The status bar at the bottom right indicates 'Ln: 7 Col: 4'.

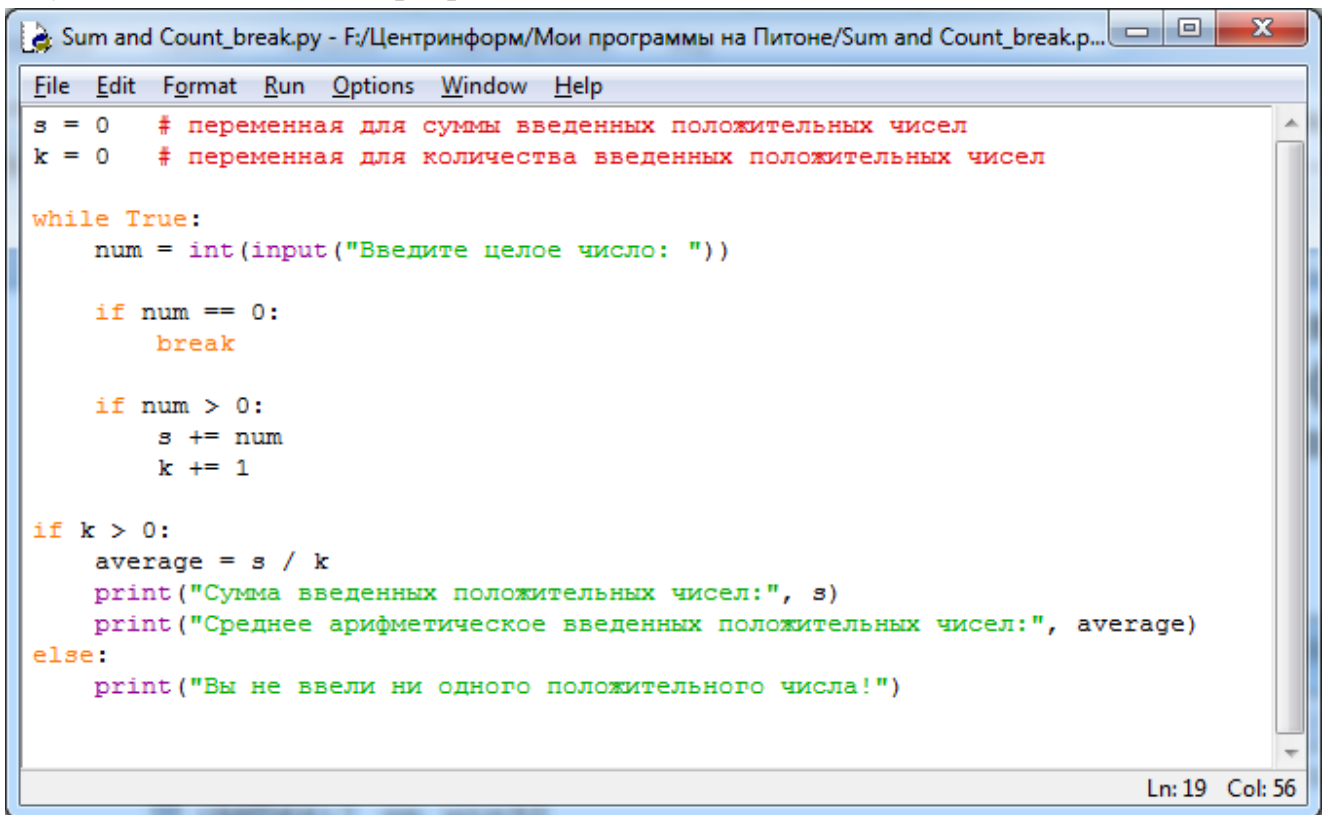
Задача 2

Написать программу, в которой осуществляется ввод целых чисел до тех пор, пока не будет введен ноль, и подсчёт суммы и среднего арифметического введенных положительных чисел.

Решение:

```
s = 0    #переменная для суммы введенных положительных чисел
k = 0    #переменная для количества введенных положительных
чисел
while True:
    num = int(input("Введите целое число: "))
    if num == 0:
        break
    if num > 0:
        s += num
        k += 1
if k > 0:
    average = s / k
    print("Сумма введенных положительных чисел:", s)
    print("Среднее арифметическое введенных положительных
чисел:", average)
else:
    print("Вы не ввели ни одного положительного числа!")
```

Результат выполнения программы:



```
File Edit Format Run Options Window Help
s = 0    # переменная для суммы введенных положительных чисел
k = 0    # переменная для количества введенных положительных чисел

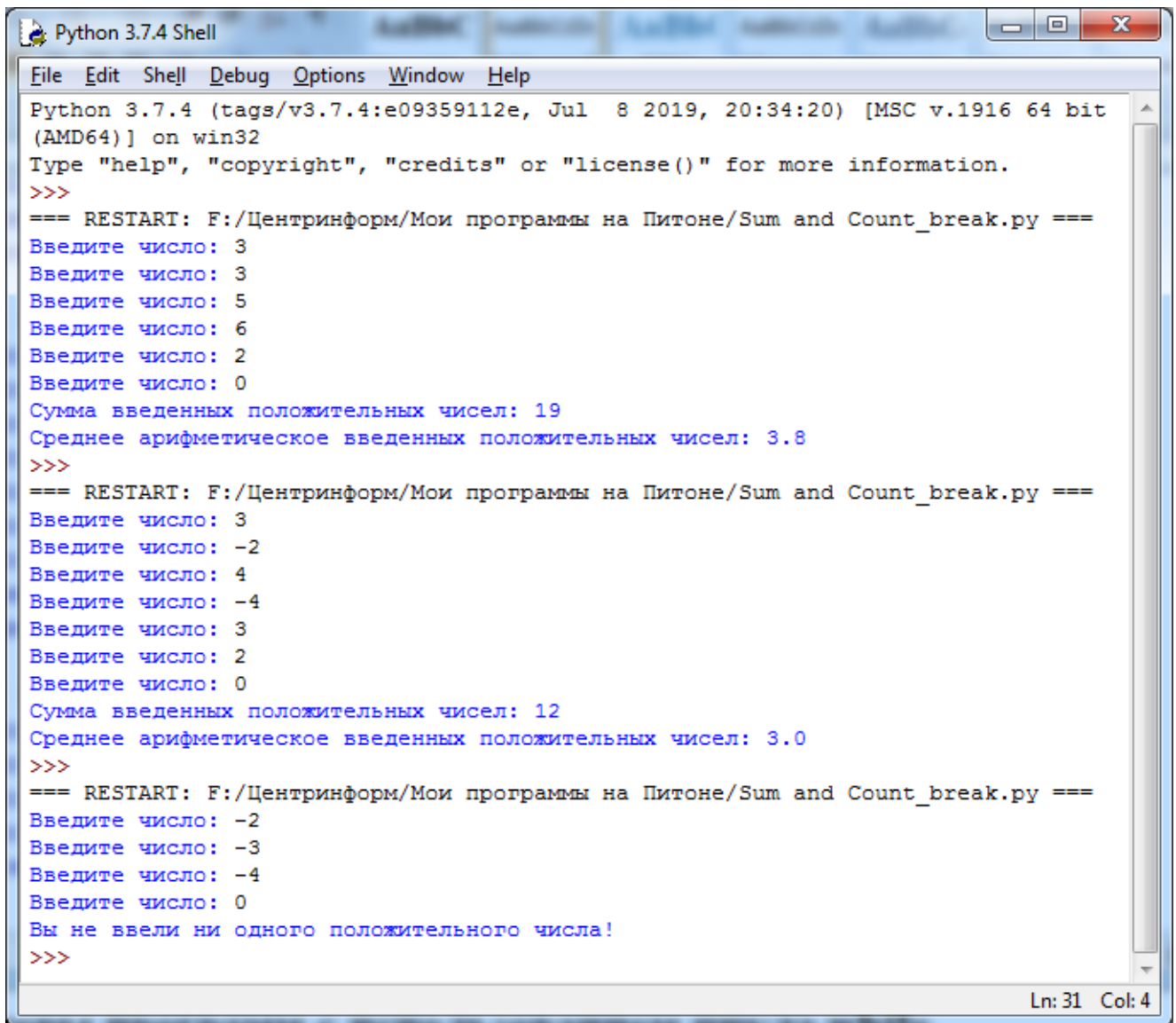
while True:
    num = int(input("Введите целое число: "))

    if num == 0:
        break

    if num > 0:
        s += num
        k += 1

if k > 0:
    average = s / k
    print("Сумма введенных положительных чисел:", s)
    print("Среднее арифметическое введенных положительных чисел:", average)
else:
    print("Вы не ввели ни одного положительного числа!")

Ln: 19 Col: 56
```



```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: F:/Центринформ/Мои программы на Питоне/Sum and Count_break.py ===
Введите число: 3
Введите число: 3
Введите число: 5
Введите число: 6
Введите число: 2
Введите число: 0
Сумма введенных положительных чисел: 19
Среднее арифметическое введенных положительных чисел: 3.8
>>>
=== RESTART: F:/Центринформ/Мои программы на Питоне/Sum and Count_break.py ===
Введите число: 3
Введите число: -2
Введите число: 4
Введите число: -4
Введите число: 3
Введите число: 2
Введите число: 0
Сумма введенных положительных чисел: 12
Среднее арифметическое введенных положительных чисел: 3.0
>>>
=== RESTART: F:/Центринформ/Мои программы на Питоне/Sum and Count_break.py ===
Введите число: -2
Введите число: -3
Введите число: -4
Введите число: 0
Вы не ввели ни одного положительного числа!
>>>
```

Задача 3

Написать программу, которая на вход получает строку и некий символ. Необходимо проверить наличие данного символа во введенной строке. Результат проверки каждого символа выводить на экран в виде текстового сообщения.

Указание. В данной задаче необходимо использовать функцию **len()**, которая считает количество символов в строке.

Решение:

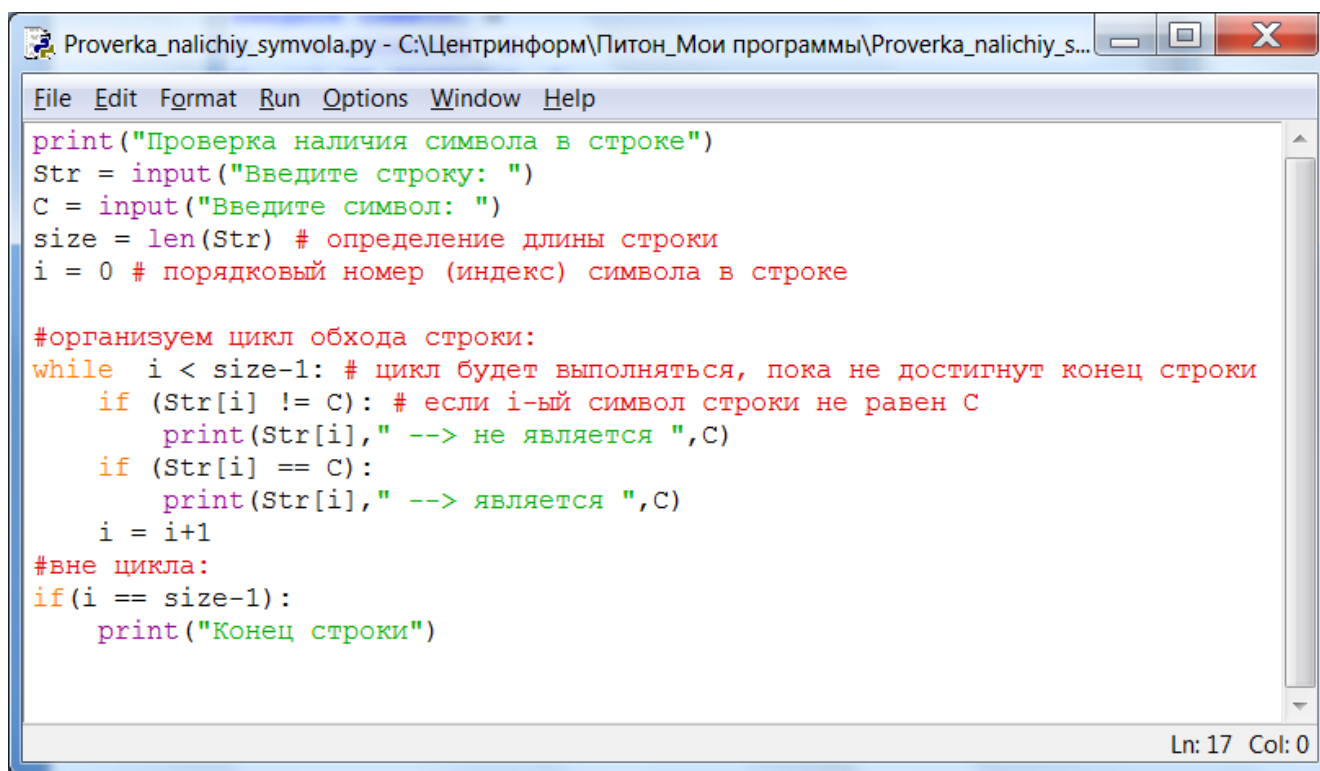
```
print("Проверка наличия символа в строке")
Str = input("Введите строку: ")
C = input("Введите символ: ")
size = len(Str) # определение длины строки
i = 0 # порядковый номер (индекс) символа в строке
#организуем цикл обхода строки:
while i < size-1: # цикл будет выполняться, пока не достигнут
конец строки
    if (Str[i] != C): # если i-ый символ строки не равен C
        print(Str[i], " --> не является ",C)
    if (Str[i] == C): # если i-ый символ строки равен C
        print(Str[i], " --> является ",C)
```

```

        i = i+1
#вне цикла:
if(i == size-1):
    print("Конец строки")

```

Результат выполнения программы:

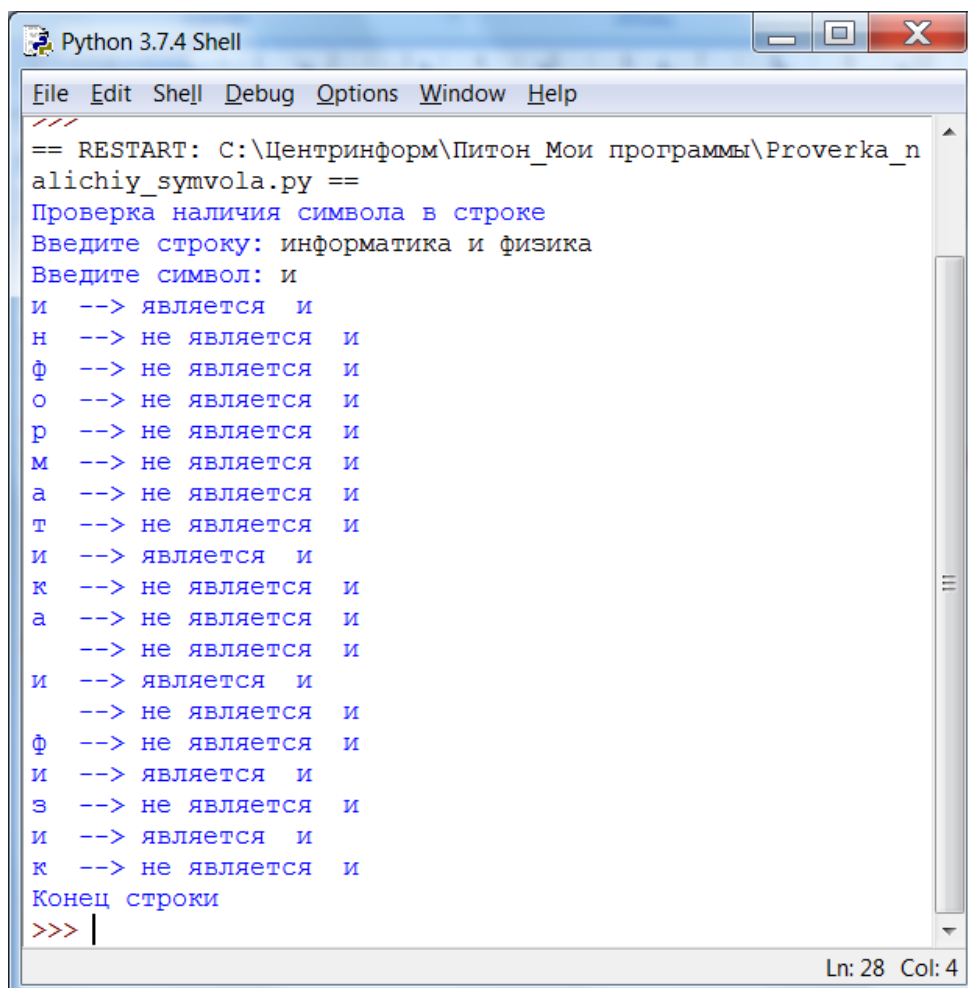


```

Proverka_nalichiy_symvola.py - C:\Центринформ\Питон_Мои программы\Proverka_nalichiy_s...
File Edit Format Run Options Window Help
print("Проверка наличия символа в строке")
Str = input("Введите строку: ")
C = input("Введите символ: ")
size = len(Str) # определение длины строки
i = 0 # порядковый номер (индекс) символа в строке

#организуем цикл обхода строки:
while i < size-1: # цикл будет выполняться, пока не достигнут конец строки
    if (Str[i] != C): # если i-ый символ строки не равен C
        print(Str[i], " --> не является ", C)
    if (Str[i] == C):
        print(Str[i], " --> является ", C)
    i = i+1
#вне цикла:
if(i == size-1):
    print("Конец строки")
Ln: 17 Col: 0

```



```

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
== RESTART: C:\Центринформ\Питон_Мои программы\Proverka_nalichiy_symvola.py ==
Проверка наличия символа в строке
Введите строку: информатика и физика
Введите символ: и
и --> является и
н --> не является и
ф --> не является и
о --> не является и
р --> не является и
м --> не является и
а --> не является и
т --> не является и
и --> является и
к --> не является и
а --> не является и
--> не является и
и --> является и
--> не является и
ф --> не является и
и --> является и
э --> не является и
и --> является и
к --> не является и
Конец строки
>>> |
Ln: 28 Col: 4

```


3. Просмотр учебного видеоролика «Изучаем циклы, цикл while» (с целью расширения знаний, наглядного представления и закрепления материала):

<https://youtu.be/fHj3RiQQlhE> (длина ролика: 5:58).

4. Закрепление

Задача 4

Написать программу, которая на вход получает строку и некий символ. Необходимо проверить наличие данного символа во введённой строке и посчитать, сколько раз он встречается в строке (может и ни разу не встечаться, это допустимая ситуация). На экран вывести результаты проверки каждого символа в строке и количество повторений заданного символа.

Указание. Данная задача решается на основе задачи 3, которую мы разобрали выше.

!!! Решение к задаче будет на следующей странице. Но сначала попробуйте решить её самостоятельно.

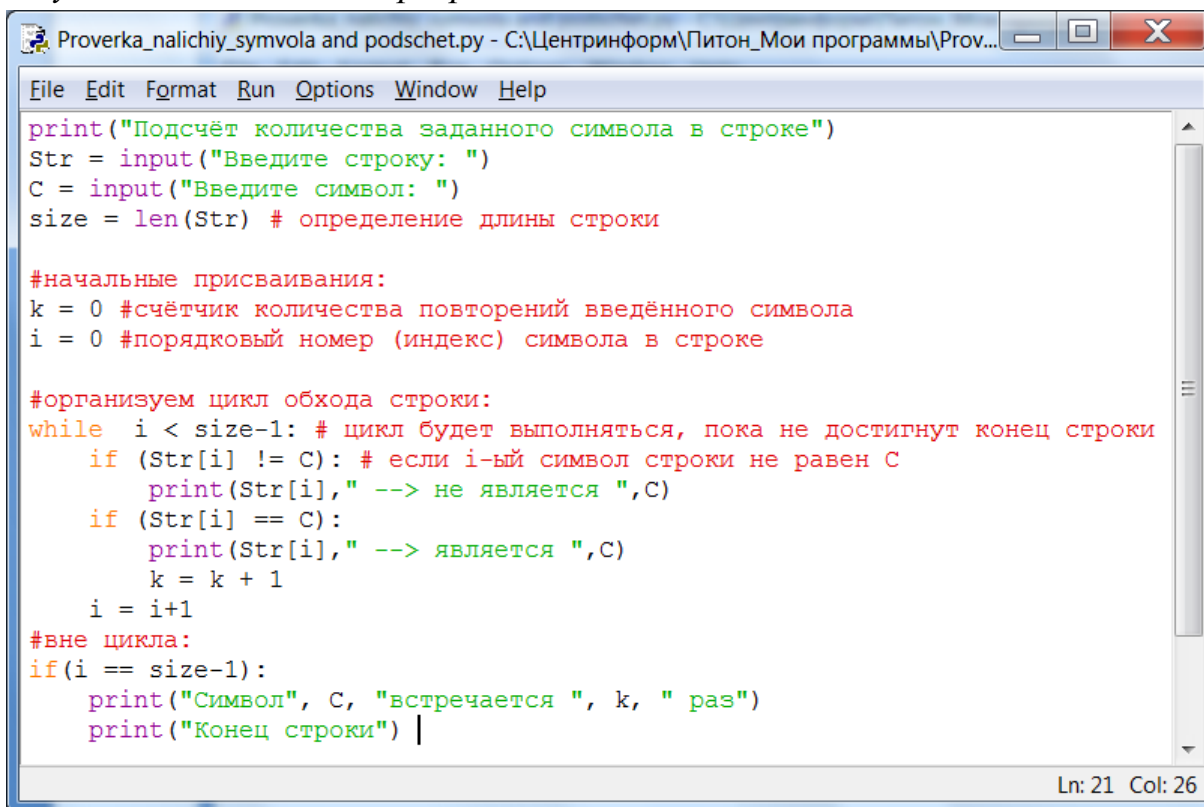
РЕШЕНИЕ

```
print("Подсчёт количества повторений заданного символа в строке")
Str = input("Введите строку: ")
C = input("Введите символ: ")
size = len(Str) # определение длины строки

#начальные присваивания:
k = 0 #счётчик количества повторений введённого символа
i = 0 #порядковый номер (индекс) символа в строке

#организуем цикл обхода строки:
while i < size-1: # цикл будет выполняться, пока не достигнут
    конец строки
    if (Str[i] != C): # если i-ый символ строки не равен C
        print(Str[i], " --> не является ", C)
    if (Str[i] == C):
        print(Str[i], " --> является ", C)
        k = k + 1
    i = i+1
#вне цикла:
if(i == size-1):
    print("Символ", C, "встречается ", k, " раз")
    print("Конец строки")
```


Результат выполнения программы:



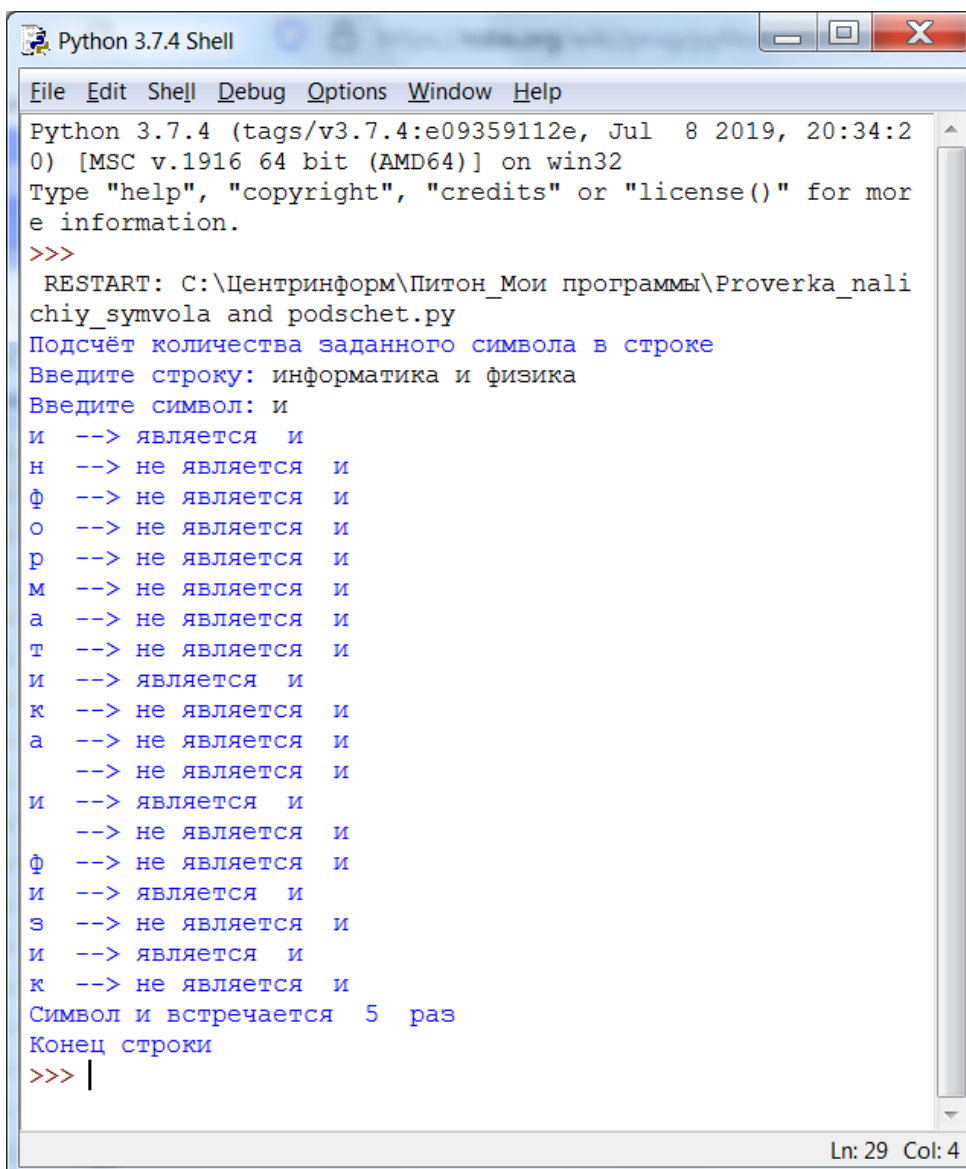
```
Proverka_nalichiy_symvola and podschet.py - C:\Центринформ\Питон_Мои программы\Prov...
File Edit Format Run Options Window Help

print("Подсчёт количества заданного символа в строке")
Str = input("Введите строку: ")
C = input("Введите символ: ")
size = len(Str) # определение длины строки

#начальные присваивания:
k = 0 #счётчик количества повторений введённого символа
i = 0 #порядковый номер (индекс) символа в строке

#организуем цикл обхода строки:
while i < size-1: # цикл будет выполняться, пока не достигнут конец строки
    if (Str[i] != C): # если i-ый символ строки не равен C
        print(Str[i], " --> не является ", C)
    if (Str[i] == C):
        print(Str[i], " --> является ", C)
        k = k + 1
    i = i+1
#вне цикла:
if(i == size-1):
    print("Символ", C, "встречается ", k, " раз")
    print("Конец строки") |
```

Ln: 21 Col: 26



```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Центринформ\Питон_Мои программы\Proverka_nalichiy_symvola and podschet.py
Подсчёт количества заданного символа в строке
Введите строку: информатика и физика
Введите символ: и
и --> является и
н --> не является и
ф --> не является и
о --> не является и
р --> не является и
м --> не является и
а --> не является и
т --> не является и
и --> является и
к --> не является и
а --> не является и
--> не является и
и --> является и
--> не является и
ф --> не является и
и --> является и
з --> не является и
и --> является и
к --> не является и
Символ и встречается 5 раз
Конец строки
>>> |
```

Ln: 29 Col: 4

Платформа Programiz

main.py	Run	Output
<pre>1 print("Подсчёт количества повторений заданного символа в строке") 2 Str = input("Введите строку: ") 3 C = input("Введите символ: ") 4 size = len(Str) 5 k = 0 6 i = 0 7 while i < size-1: 8 if (Str[i] != C): 9 print(Str[i], " --> не является ", C) 10 if (Str[i] == C): 11 print(Str[i], " --> является ", C) 12 k = k + 1 13 i = i+1 14 #вне цикла: 15 if(i == size-1): 16 print("Символ", C, " встречается ", k, " раз") 17 print("Конец строки") 18</pre>		<pre>Подсчёт количества повторений заданного символа в строке Введите строку: информатика и физика Введите символ: и и --> является и н --> не является и ф --> не является и о --> не является и р --> не является и м --> не является и а --> не является и т --> не является и и --> является и к --> не является и а --> не является и --> не является и и --> является и --> не является и ф --> не является и и --> является и з --> не является и и --> является и к --> не является и Символ и встречается 5 раз Конец строки</pre>