

Урок по информатике на тему:

Программирование разветвляющихся алгоритмов

УСЛОВНЫЙ ОПЕРАТОР

```
1 x = int(input())
2 if x > 0:
3     print(x)
4 else:
5     print(-x)
```

if <условие>:
 <блок инструкций 1>
else:
 <блок инструкций 2>

Поиск

Стек Данных

Оболочка Python

Команды выполняются без отладки. Ис

```
3.7.2 (tags/v3.7.2:9a3ffc0492,
Python Type "help", "copyright"
>>> [анализируем 23.py]
-5
5
>>> |
```

Для выделения блока инструкций, в языке Питон используются отступы. Все инструкции, которые относятся к одному блоку, должны иметь равную величину отступа, то есть одинаковое число пробелов. Рекомендуется использовать отступ в 4 пробела

НЕПОЛНЫЙ УСЛОВНЫЙ ОПЕРАТОР

23.py *

```
1 x = int(input())
2 if x < 0:
3     x = -x
4 print(x)
```

if <условие>:
 <блок инструкций>

Поиск

Стек Данных

Оболочка Python

Команды выполняются без отладки. Исг   Опции

```
3.7.2 (tags/v3.7.2:9a3fffc0492, Dec 23 2018, 2
Python Type "help", "copyright", "credits" or
>>> [анализируем 23.py]
-6
6
>>>
```

ВЛОЖЕННЫЙ УСЛОВНЫЙ ОПЕРАТОР

```
1 x = int(input())
2 y = int(input())
3 if x > 0:
4     if y > 0:
5         print("Первая четверть")
6     else:
7         print("Четвертая четверть")
8 else:
9     if y > 0:
10        print("Вторая четверть")
11    else:
12        print("Третья четверть")
```

Поиск Стек Данных Оболочка Python

Команды выполняются без отладки. Иск   Опции

```
3.7.2 (tags/v3.7.2:9a3fffc0492, Dec 23 2018, 2
Python Type "help", "copyright", "credits" or
>>> [анализируем 23.py]
2
-3
Четвертая четверть
```

ОПЕРАТОРЫ СРАВНЕНИЯ И СЛОЖНЫЕ УСЛОВИЯ

<

>

<=

>=

==

!=

меньше

больше

меньше
или равнобольше или
равно

равенство

не равно

Операторы сравнения в Питоне можно объединять в цепочки (в отличие от большинства других языков программирования, где для этого нужно использовать логические связки), например, **a == b == c** или **1 <= x <= 10**.

Условие принимает два логических значения **True** или **False**

В Питоне существуют стандартные логические операторы:

логическое И **and**

логическое ИЛИ **or**

логическое отрицание **not**

КАСКАДНЫЙ УСЛОВНЫЙ ОПЕРАТОР

```
1 x = int(input())
2 y = int(input())
3 if x > 0 and y > 0:
4     print("Первая четверть")
5 elif x > 0 and y < 0:
6     print("Четвертая четверть")
7 elif y > 0:
8     print("Вторая четверть")
9 else:
10    print("Третья четверть")
```

if <условие>:
 <блок инструкций 1>
elif <условие>:
 <блок инструкций 2>
else
 <блок инструкций 3>

Поиск Стек Данных Оболочка Python

Команды выполняются без отладки. Исг  

```
3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23
Python Type "help", "copyright", "cre
>>> [анализируем 23.py]
3
-4
Четвертая четверть
```

В такой конструкции условия проверяются по очереди, выполняется блок, соответствующий первому из истинных условий. Если все проверяемые условия ложны, то выполняется блок **else**, если он присутствует.

ДОМАШНЕЕ ЗАДАНИЕ

- §3.4
- №6-16, решить на Python!

РЕШИТЬ

• УРОК 3 УСЛОВНАЯ ИНСТРУКЦИЯ (5 ЗАДАЧ)

Задача А. Максимум двух чисел

Задача В. Какое число больше?

Задача С. Знак числа

Задача D. Високосный год

Задача Е. Максимум трех чисел

Задача F. Существует ли треугольник?

Задача G. Сколько совпадает чисел

Задача H. Тестирующая система

Задача I. Ход ладьи

Задача J. Ход короля

Задача K. Ход слона

Задача L. Ход ферзя

Задача M. Ход коня

Задача N. Шахматная доска

Задача O. Шоколадка

Задача P. Линейное уравнение

Задача Q. Коровы

Задача R. Яша плавает в бассейне

Задача S. Тип треугольника

Задача T. Билеты на метро - 1

Задача U. Билеты на метро -2

Задача V. Узник замка Иф

Задача W. Коробки

Задача X. Складирование ноутбуков

Задача Y. Аншлаг, аншлаг!

Задача Z. Котлеты

Задача AA. Упорядочить три числа