**Модуль 1. Объекты программы на Python. Основные алгоритмические конструкции**

Выполнил Мелега Алексей Валерьевич

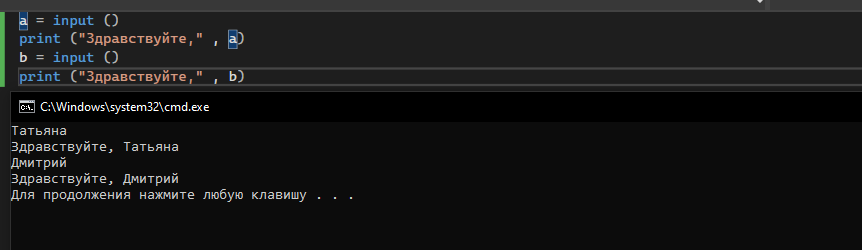
**Урок 3-5. Структура программы на Python,   
базовые типы данных, операторы ввода и вывода данных**

**Задание 6 (\*)**

До изучения методов Искусственного Интеллекта нам, конечно, ещё далеко, но давайте вдохнём жизнь в эти бездушные строчки программного кода. Напишите программу, которая по введённому имени поприветствует пользователя в форме:

*Здравствуйте,* [имя]

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| Татьяна | Здравствуйте, Татьяна |
| Дмитрий | Здравствуйте, Дмитрий |



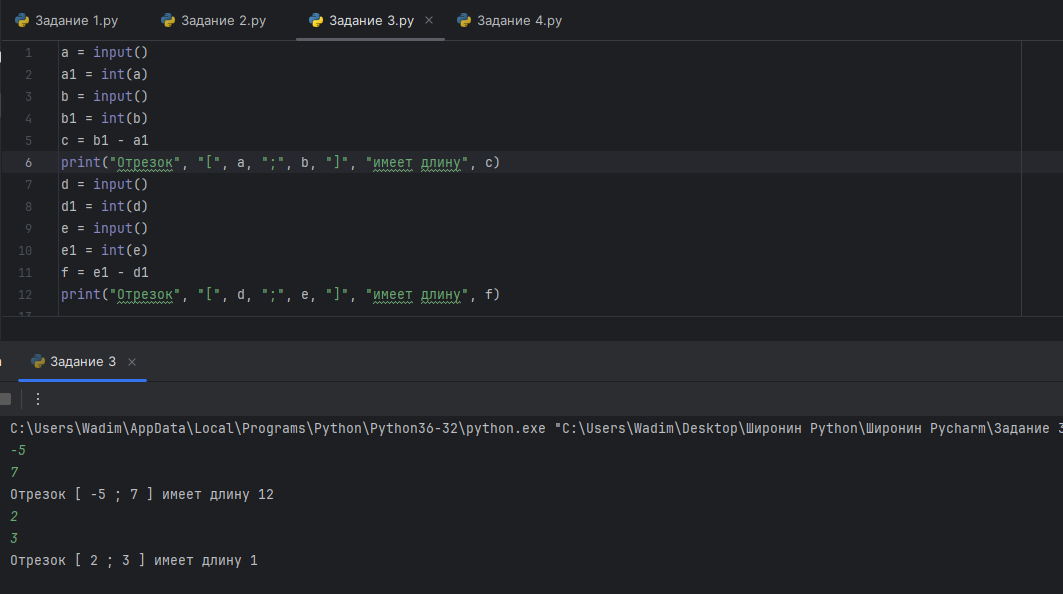
**Урок 6. Знакомство с IDE PyCharm.   
Установка и настройка среды программирования**

**Задание 3**

Напишите программу, которая для заданного отрезка вернёт информацию в следующей форме:

*Отрезок* [левая граница] *;* [правая граница] *] имеет длину* [длина]

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| -5  7 | Отрезок [ -5 ; 7 ] имеет длину 12 |
| 2  3 | Отрезок [ 2 ; 3 ] имеет длину 1 |

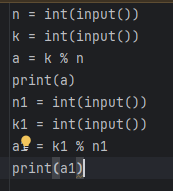


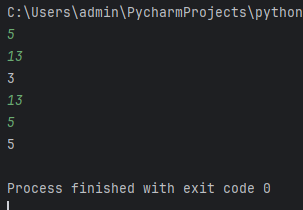
## Урок 11–13. Оператор присваивания. Решение вычислительных задач

**Задание 3**

Компания из *n* грибников собрала *k* грибов и после возвращения домой решила поделить поровну найденные грибы. При условии, что один гриб нельзя разделять на части, грибники договорились сложить оставшиеся грибы обратно в корзину. Ответьте на вопрос, сколько грибов останется в корзине? Сначала вводится число *n*, затем – число *k*.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 5  13 | 3 |
| 13  5 | 5 |

****

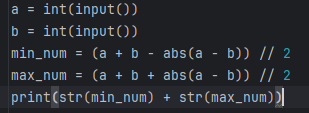
****

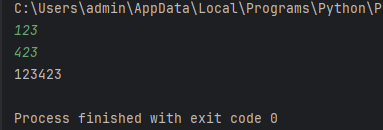
## Задания к уроку 15-16. Линейная программа. Регулирование вывода через sep и end

**Задание 5 (\*)**

*Без использования условных операторов, операторов отношения, функций min и max*, Вам необходимо написать программу, которая принимает в качестве входных данных два числа, после чего последовательно выводит наименьшее из низ, затем — наибольшее. Вывод происходит в одну строку без пробелов.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 123  423 | 123423 |
| 265472 321 | 321265472 |



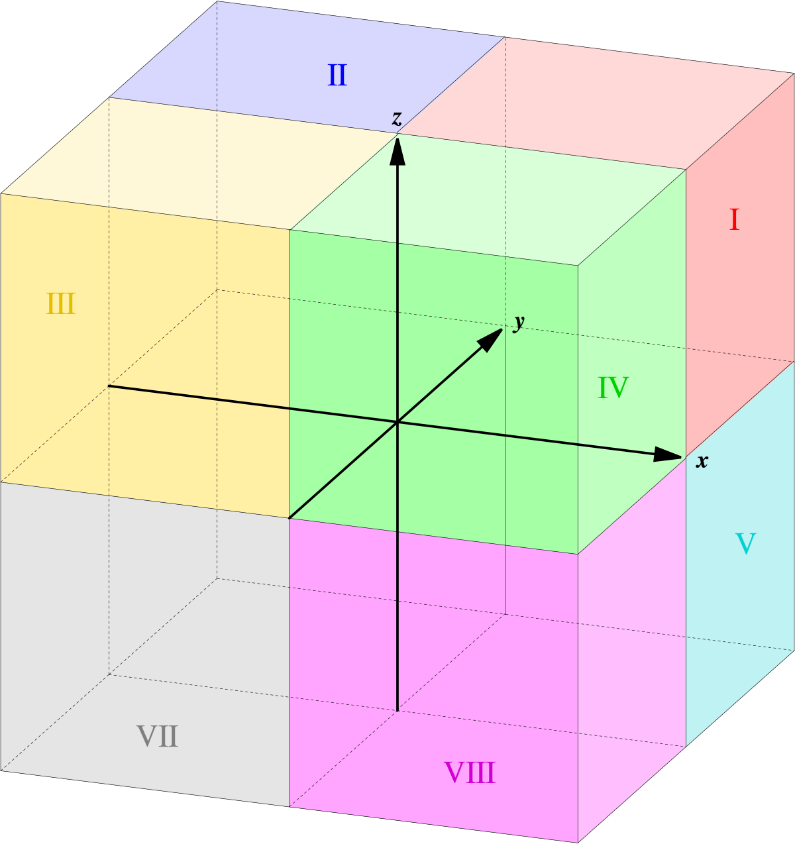


## Урок 20-21. Логические операции в сложных условиях. Программирование ветвления, отладка кода

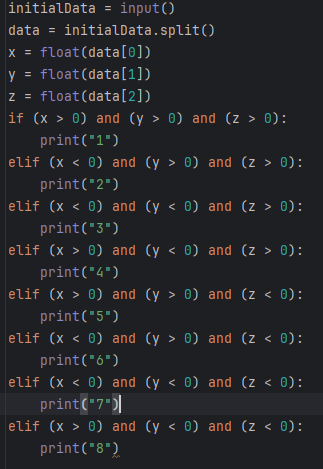
**Задание 5**

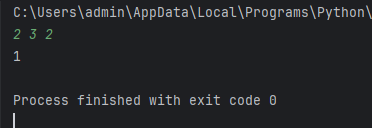
Вам необходимо написать программу, которая для координат конкретной точки в трёхмерном пространстве, утверждает, в каком октанте она находится.

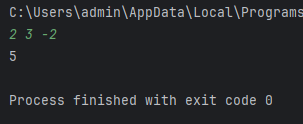
Расположение октантов можно посмотреть на картинке ниже:



|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 2 3 2 | 1 |
| 2 3 -2 | 5 |





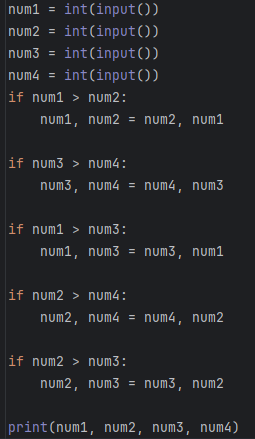


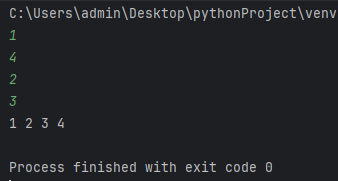
## Урок 22-23. Вложенность в структурах ветвления. Применение оператора выбора

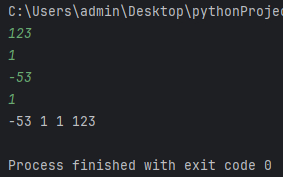
**Задание 5 (\*)**

Используя только условный оператор и оператор присваивания (не считая ввода и вывода), напишите программу, которая для четырёх введённых целых чисел, выводит их в порядке неубывания.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 1 4 2 3 | 1 2 3 4 |
| 123 1 -53 1 | -53 1 1 123 |







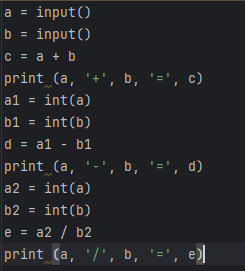
## Урок 7. Линейная программа. Построение диалога компьютер-пользователь

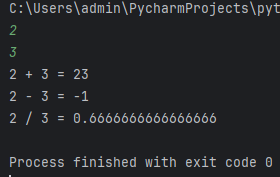
**Задание 1**

Напишите программу, которая принимает в качестве входных данных два целых числа, после чего выводит три следующих равенства:

[число 1] + [число 2] = [сумма]  
[число 1] - [число 2] = [разность]  
[число 1] / [число 2] = [отношение]

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 2  3 | 2 + 3 = 5  2 - 3 = -1  2 / 3 = 0.6666667 |





## Урок 33-34. Циклы. Классические алгоритмы: алгоритм Евклида, выделение и обработка цифр числа, вычисление факториала, получение делителей натурального числа

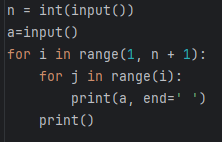
**Задание 4**

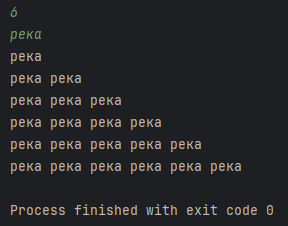
Для создания образов поэты порой обращаются к такому приему, как повтор -- данный прием основывается, как это не сложно догадаться, на повторении слова или сочетания слов, что должно увеличить вызываемую у читателя эмоцию, расширить смысл передаваемых образов.

Вам необходимо написать программу, которая выводила бы стихотворение, построенное по следующему принципу. Основой стихотворения является одно слово (вводится с клавиатуры). Первая поэтическая строка состоит только из основного слова, вторая строка - из двух основных слов, и каждая следующая строка будет отличаться от предыдущей количеством слов – в ней будет на одно слово больше.

В первой строке программа принимает на вход целое число n (n – количество строк в стихотворении). Во второй строке программа принимает на вход одно слово, являющееся основным. Программа выводит n строк, составленных по описанному выше правилу.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 6  река | река  река река  река река река  река река река река  река река река река река  река река река река река река |





## Урок 27-28. Циклические схемы в программах. Ввод данных в цикле

**Задание 7**

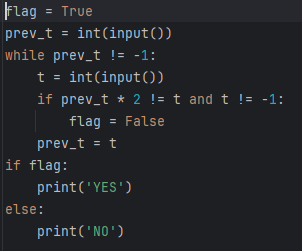
Тот же режиссер имеет много концепций для снятия картин. Так, еще одна его идея заключается в том, что каждая следующая сцена в фильме должна идти в два раза дольше предыдущей. Это могут быть не обязательно степени двойки, например если первая сцена идет 3 минуты, то следующая будет идти 6 минут, сцена за ней – 12 минут и так далее...

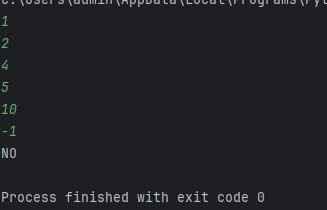
Напишите программу, которая проверяла бы, соответствует ли фильм данному критерию.

Программа принимает последовательность чисел, которые по порядку характеризуют длительность сцен в фильме. Последовательность оканчивается числом -1.

Если фильм соответствует критерию концепции режиссера, то программа должна вывести слово "YES", иначе – "NO".

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 1  2  4  5  10  -1 | NO |





## Урок 26. Организация циклов. Виды циклов в Python. Принципы работы вложенных циклов

**Задание 1**

Один из самых простых рядов чисел – ряд натуральных чисел, но давайте несколько его усложним и будем рассматривать не все числа, а только некоторый отрезок этого ряда.

Напишите программу, которая принимает два числа – a и b – границы рассматриваемого отрезка. Будем считать, что в случае, если a < b, то программа должна вывести все числа из указанного отрезка в порядке возрастания, иначе – в порядке убывания.

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 5  9 | 5 6 7 8 9 |
| 8  3 | 8 7 6 5 4 3 |

