

# Программирование на Python

## Урок №12

### План урока:

1. Функции и блок-схемы
2. Вызов функции внутри функции
3. Задания

### Результат:

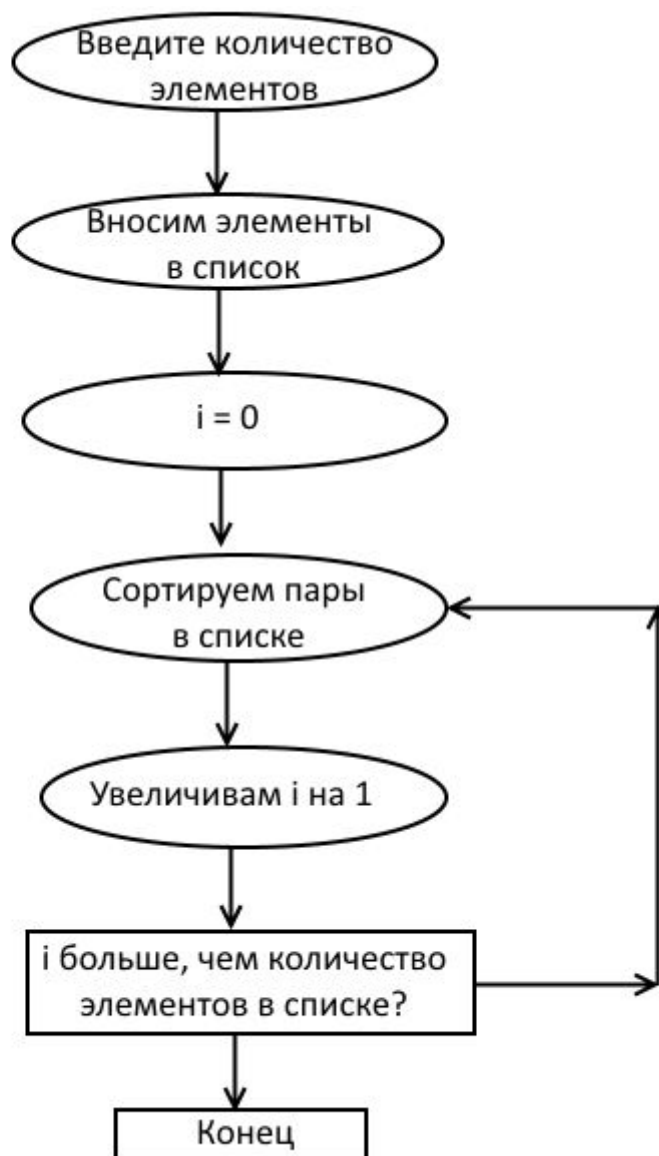
```
def sortirovka(new_spisok):  
    i = 0  
    while i < len(new_spisok) - 1:  
        if new_spisok[i] > new_spisok[i+1]:  
            swap = new_spisok[i]  
            new_spisok[i] = new_spisok[i+1]  
            new_spisok[i+1] = swap  
        i = i + 1  
  
def srt(spisok):  
    i = 0  
    while i < len(spisok):  
        sortirovka(spisok)  
        i = i + 1  
  
spisok = []  
chis_elem = int(input())  
i = 0  
while i < chis_elem:  
    new_elem = int(input())  
    spisok.append(new_elem)  
    i = i + 1  
srt(spisok)  
print(spisok)
```

## Функции и блок-схемы:

Функции очень удобно располагать в блок-схемах. Реализация функций не входит в описание блок-схем, но в этом и заключается основной их смысл — скрыть реализацию от пользователя, предложив ему лишь описание работы функции. Пусть у нас есть функция, которая меняет местами пары в списке для сортировки по возрастанию. Выглядит она следующим образом.

```
def sortirovka(new_spisok):  
    i = 0  
    while i < len(new_spisok) - 1:  
        if new_spisok[i] > new_spisok[i+1]:  
            swap = new_spisok[i]  
            new_spisok[i] = new_spisok[i+1]  
            new_spisok[i+1] = swap  
        i = i + 1
```

Составим блок-схему, принимая во внимание, что у нас есть такая функция.



Так блок-схема становится намного более читабельной.

### **Вызов функции внутри функции:**

Мы уже вызывали функции внутри функций. К примеру, когда использовали `print()` или `input()`. Вызов функций, которые создали мы сами, ничем не отличается от вызова встроенных функций. К примеру, имея функцию из пункта выше, мы можем создать функцию сортировки списка, которая будет использовать нашу функцию `sortirovka`. Принцип ее работы мы уже полностью изобразили на блок-схеме из прошлого пункта, с единственным

отличием, что нам сразу передается готовый список и нам не надо его создавать и заполнять.

```
def sortirovka(new_spisok):
    i = 0
    while i < len(new_spisok) - 1:
        if new_spisok[i] > new_spisok[i+1]:
            swap = new_spisok[i]
            new_spisok[i] = new_spisok[i+1]
            new_spisok[i+1] = swap
        i = i + 1

def srt(spisok):
    i = 0
    while i < len(spisok):
        sortirovka(spisok)
        i = i + 1

spisok = []
chis_elem = int(input())
i = 0
while i < chis_elem:
    new_elem = int(input())
    spisok.append(new_elem)
    i = i + 1
srt(spisok)
print(spisok)
```

Теперь наша программа выглядит более понятной, а так же мы можем сортировать любой список простой командой srt().

### Задания:

1. Создать еще одну функцию, которая будет использовать уже готовую функцию sortirovka() и сортировать список в порядке убывания.
2. Создать функцию, которая принимает список и выводит его элементы в обратном порядке
3. Создать функцию, которая выводит элементы списка в порядке возрастания.
4. Создать функцию, которая выводит максимальный и минимальный элемент в списке.

## Тайминг:

Тема	Время с начала занятия, мин
Функции и блок-схемы	15
Вызов функции внутри функции	20
Задания	85
Контрольные вопросы	90

## Контрольные вопросы:

1. В чем заключается основная идея делегации логических частей кода в функцию?
2. Какие в этом плюсы?
3. Как строить блок-схему с функциями?