



Программирование, лекция 8. Подпрограммы, функции



Кафедра ИУ7 МГТУ им. Н. Э. Баумана,
2022 год

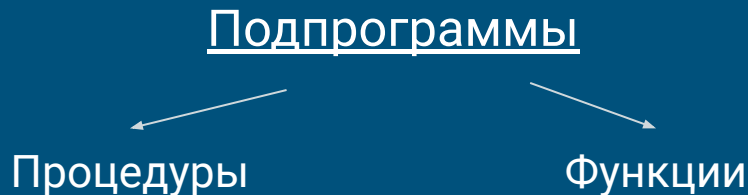


Оператор сравнения с шаблоном (3.10+)

```
match subject:
  case <pattern_1>:
    <action_1>
  case <pattern_2>:
    <action_2>
  case <pattern_3>:
    <action_3>
  case _:
    <action_wildcard>
```

Подпрограммы

Подпрограмма - поименованная или иным образом идентифицированная отдельная функционально независимая часть компьютерной программы.



Параметры подпрограммы - переменные, которые вызывающая программа передаёт подпрограмме

Формальные параметры - те, которые объявлены при описании подпрограммы

Фактические параметры (аргументы) - те, которые передаются в подпрограмму при её вызове

Функции в языке Python

Оператор `def` создаёт новый объект и присваивает его имени

`def имя_функции(список формальных параметров):`

операторы

`def times(a, b):` # `a, b` - формальные параметры

`return a * b`

`s = times(2, 3)` # `2, 3` - фактические параметры

Параметры функций

1. Присваивание новых значений аргументам внутри функций не затрагивает вызывающий код.
2. Модификация аргумента внутри функции:
 - неизменяемого - создаст копию (не повлияет на вызывающий код)
 - изменяемого - повлияет на вызывающий код (изменит значение в нём)

Виды параметров в Python:

1. Позиционные аргументы
2. Именованные аргументы

Определение и передача аргументов

```
def func1([posonly1, posonly2, /,] pos_or_keyword1, pos_or_keyword2[, *,  
kw_only1, kw_only2]): # позиционные, смешанные, именованные
```

```
    pass
```

```
def func2(*args, **kwargs): # прочие позиционные и именованные
```

```
    pass
```

```
def func3(a, b=2): # значения по умолчанию
```

```
    pass
```

Пространства имён, области видимости

Пространство имён - множество уникальных идентификаторов (имён).

Область видимости - часть программы, в пределах которой идентификатор остаётся связан с сущностью, которой он был назначен при объявлении.

В Python пространство имён определяется по местоположению присваивания этому имени какого-либо значения.

Основы областей видимости в Python

- имена, присвоенные внутри `def`, “видны” только в коде внутри этого оператора; ссылаться на них извне функции нельзя;
- имена внутри `def` не конфликтуют с переменными за пределами `def`.

Области видимости:

1. Глобальная - если переменная объявлена за пределами всех `def`, то она является “глобальной” в целом файле.
2. Локальная - переменная, объявленная внутри `def`, будет локальной в своей функции.
3. Нелокальная - переменная, объявленная внутри `def`, включающем другие `def`.
4. Встроенная (built-in).

Изменение областей видимости

Оператор **global** делает имя внутри функции глобальным.

Оператор **nonlocal** делает имя внутри функции нелокальным.

Распознавание имён. Правило LEGB

Поиск имени выполняется последовательно в:

1. local
2. enclosing (объемлющих) функциях
3. global
4. built-in

Рекурсия. Рекурсивные функции

Рекурсия - вызов подпрограммы из неё же самой:

- непосредственно - простая рекурсия;
- через другие подпрограммы - косвенная рекурсия.

Тело рекурсивной подпрограммы должно иметь не меньше двух альтернативных (условных) ветвей, хотя бы одна из которых должна быть **терминальной**.

Виды рекурсивных вызовов

По количеству вызовов:

- линейная - в теле функции присутствует только один вызов самой себя;
- нелинейная - в теле присутствует несколько вызовов.

По месту расположения рекурсивного вызова:

- головная - рекурсивный вызов расположен ближе к началу тела функции;
- хвостовая (концевая) - рекурсивный вызов является последним оператором функции.

Функции высшего порядка

Функция первого порядка - та, которая принимает только значения “простых” (не функциональных) типов и возвращает значения таких же типов в качестве результата.

Функция высшего порядка - та, которая принимает в качестве аргументов или возвращает другие функции.