Лекция 3

Функции. Рекурсия. Модель памяти.

Функции.

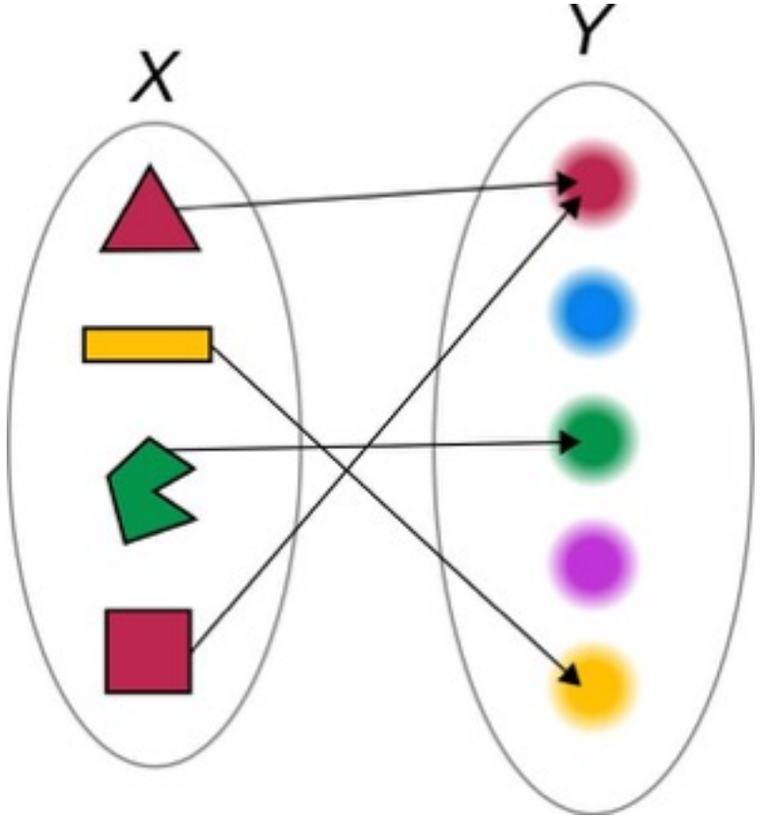
Определение функции

В математике:

Пусть X, Y - множества. Тогда **f : X -> Y** - правило, по которому каждому элементу X ставится в соответствие ровно 1 элемент из Y.

X - область определения
(входные данные)

Y - область значения
(выходные данные)



Определение функции

В программировании:

Подпрограмма, к которой можно обратиться из другого фрагмента программы.

- Может иметь несколько или даже переменное число входных данных - аргументов.
- Возвращаемое значение всегда одно.
- В некоторых языках программирования выделяют процедуры - функции, которые ничего не возвращают.

Определение функции

Парадигма черного ящика (black box):

При вызове функции нас не интересует, как она устроена внутри. Важен лишь результат ее работы.

Функции

- Ключевое слово **def** для объявления и **return** – для возвращения значения.
- После **return** происходит выход из функции, и все дальнейшие строки игнорируются.

```
def max(a, b):
    if a > b:
        return a
    else:
        return b

print(max(1, 3))
print(max(1.3, -2))
```

Функции

- Если **return** не указан, функция вернет **None** по умолчанию.
- Можно объявлять функции внутри функций.

```
def hello():
    print('Hello, world!')

hello()
print(hello())
```

Функции

- Можно указывать значения по умолчанию, но они должны идти в конце.
- Можно обращаться к аргументам по ключу, а не по позиции.

```
def list_range(start, stop=None, step=1):
    if stop == None:
        start, stop = 0, start
    res = []
    while start < stop:
        res.append(start)
        start += step
    return res

print(list_range(5 ,10))
print(list_range(10, step=2))
```

Функции

- Можно указывать переменное число параметров (VarArgs)
- Внутри функции параметры распаковываются в список.

```
def max(*a):
    res = a[0]
    for val in a[1:]:
        if val > res:
            res = val
    return res

print(max(3, 5, 4))
```

Локальные и глобальные переменные

Функции имеют локальную область видимости:

- Переменные, определенные внутри функции, не видны снаружи (так же, как в циклах).
- Переменные, определенные вне функции, можно использовать в самой функции.

```
def f(x):
    a = x * x
    return a + b

b = 1
print(f(1))
print(b)
#print(a)
```

Локальные и глобальные переменные

- Если попытаться изменить значение внешней переменной внутри функции, переменная становится локальной.
- Благодаря этому, нельзя случайно изменить значение, вызвав функцию.

```
def f(x):
    #print(b)
    b = x
    return b

b = 0
print(f(1))
print(b)
```

Локальные и глобальные переменные

- Чтобы поменять значение внешней переменной внутри функции, нужно использовать инструкцию **global**.
- Обычно, это не самое удачное решение.

```
def f(x):
    global b
    b = x
    return x

b = 0
print(f(1))
print(b)
```

Рекурсия.

Рекурсия

В математике:

Рекуррентное соотношение - способ задания функции, при котором значение в одних точках выражается через значения в других.

Факториал:

- “Нерекурсивное” определение:
  n! = Fact(n) = 1 * 2 * … * n

- Рекурсивное определение:
  Fact(n) = n * Fact(n-1)
  Fact(1) = 1

Рекурсия

Иногда формула в явном виде выглядит сложно или вообще не существует.

Числа Фибоначчи:

- Рекурсивное определение:

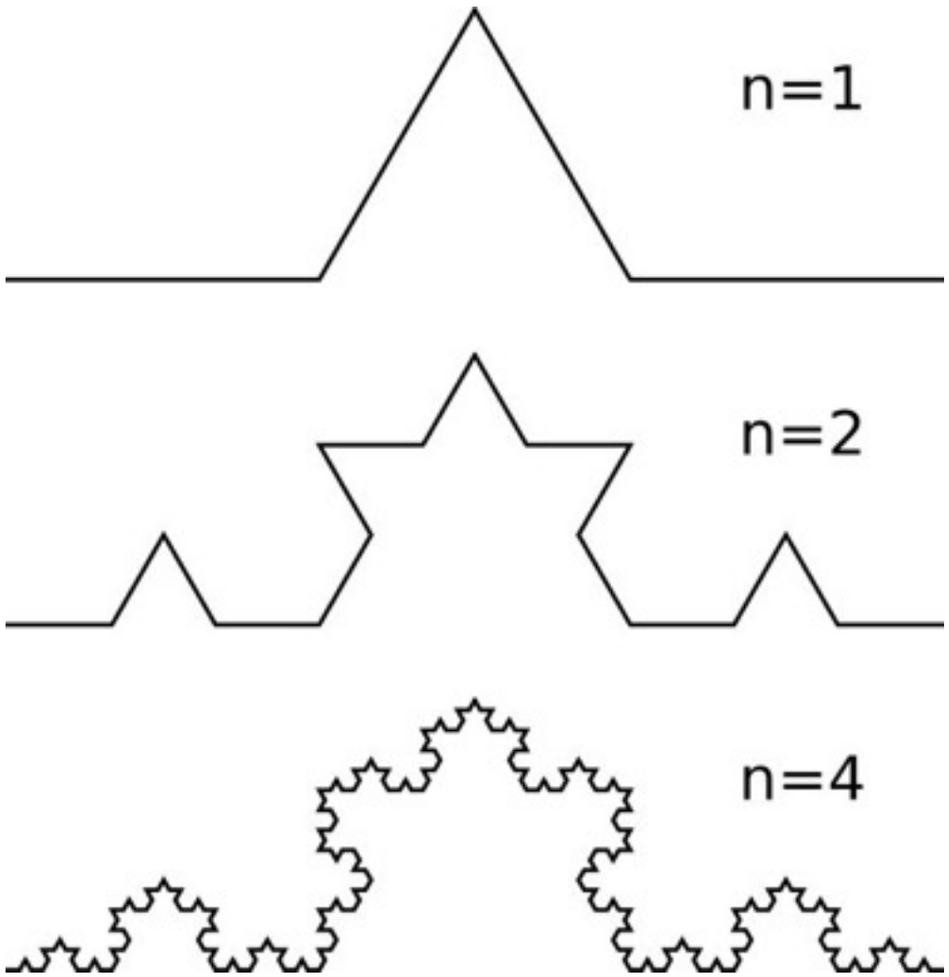
  F(n) = F(n-1) + F(n-2)
  F(1) = F(2) = 1

- “Нерекурсивное” определение:

$$F_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}} = \frac{\varphi^n - (-\varphi)^{-n}}{\varphi - (-\varphi)^{-1}} = \frac{\varphi^n - (-\varphi)^{-n}}{2\varphi - 1},$$

Рекурсия

В геометрии и искусстве:



Построение кривой Коха

M.K. Эшер “Меньше и меньше”

Рекурсия

В программировании:

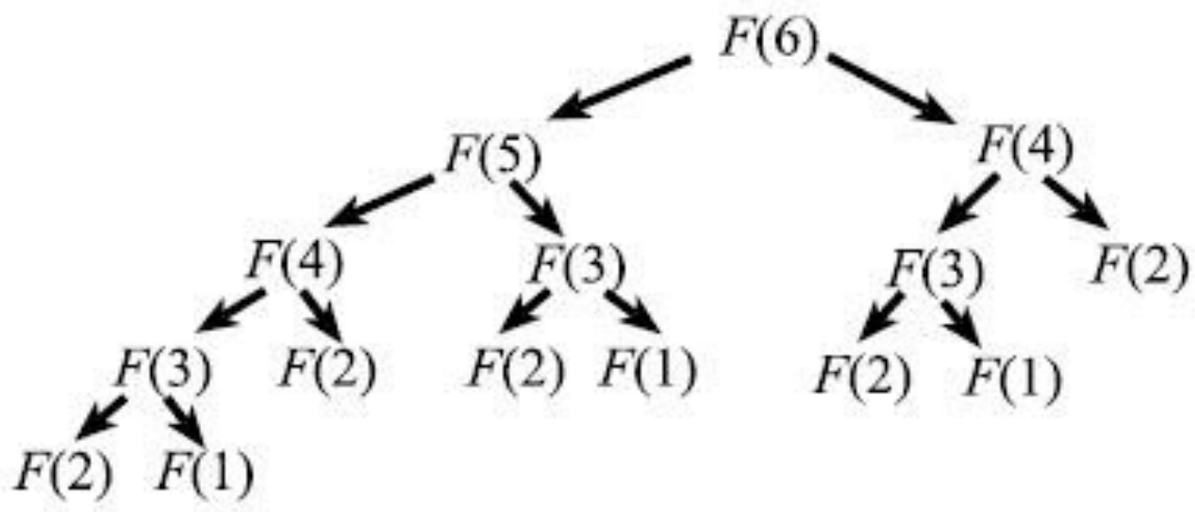
- Рекурсивная функция – в своем теле вызывает себя.
- Обязательно должен быть выход из рекурсии: проверочное условие или точка остановки.

```
def gcd(a, b):
    '''Returns greatest common divisor
of natural numbers. '''
    if b == 0:
        return a
    else:
        return gcd(b, a % b)


print(gcd(2, 1))
print(gcd(8, 12))
```

Рекурсия

- Во время выполнения рекурсивной функции создается стек рекурсии, запоминающий аргументы при каждом вызове. Если глубина рекурсии слишком велика, может произойти переполнение памяти.
- Многие рекуррентные формулы вычисляются за экспоненциальное время, что очень долго. Поэтому важно уметь оценивать время работы рекурсивных функций.



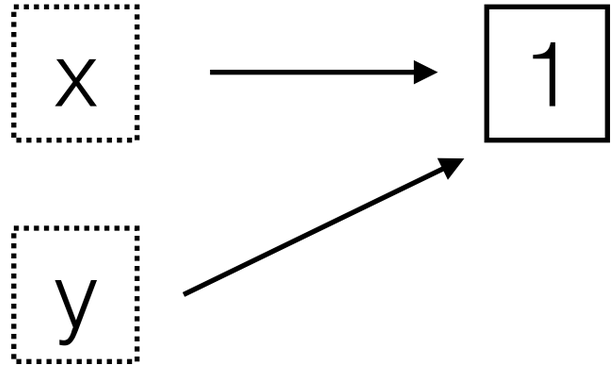
Дерево рекурсии для чисел Фибоначчи

Модель данных.

Имена и ссылки

Название в коде — ссылка → Объект в памяти

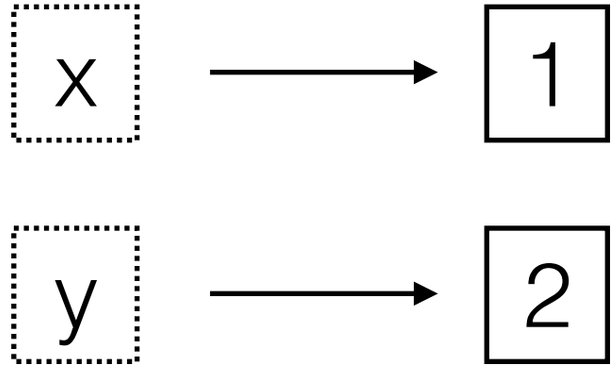
```
x = 1
y = x
```



Имена и ссылки

Название в коде — ссылка → Объект в памяти

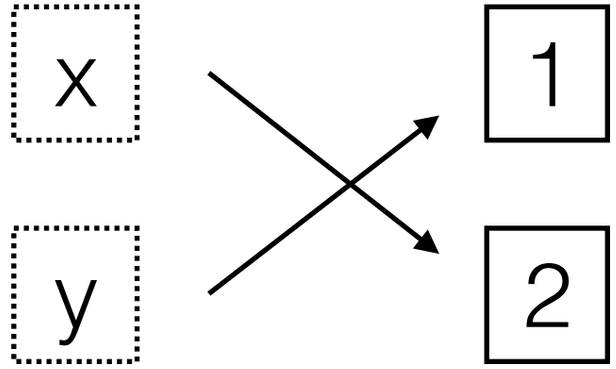
```
x = 1
y = x
x = 2
```



Имена и ссылки

Название в коде — ссылка → Объект в памяти

```
x = 1
y = x
x = 2
x, y = y, x
```



Объекты

Все данные в программе - объекты:

- Числа, списки, строки
- Функции, классы.

Основные свойства объектов:

- Идентичность (identity)
- Тип (type)
- Значение (value)

Свойства объектов

Identity

- ~ адрес объекта в памяти.
- Нельзя изменить.
- Можно сравнить с помощью **is** и **is not**.

Тип

- Влияет на возможные операции и значения.
- Нельзя изменить.
- **type**()

Значение

- Изменяемость (mutability) зависит от типа.
- Можно сравнить с помощью **==**.