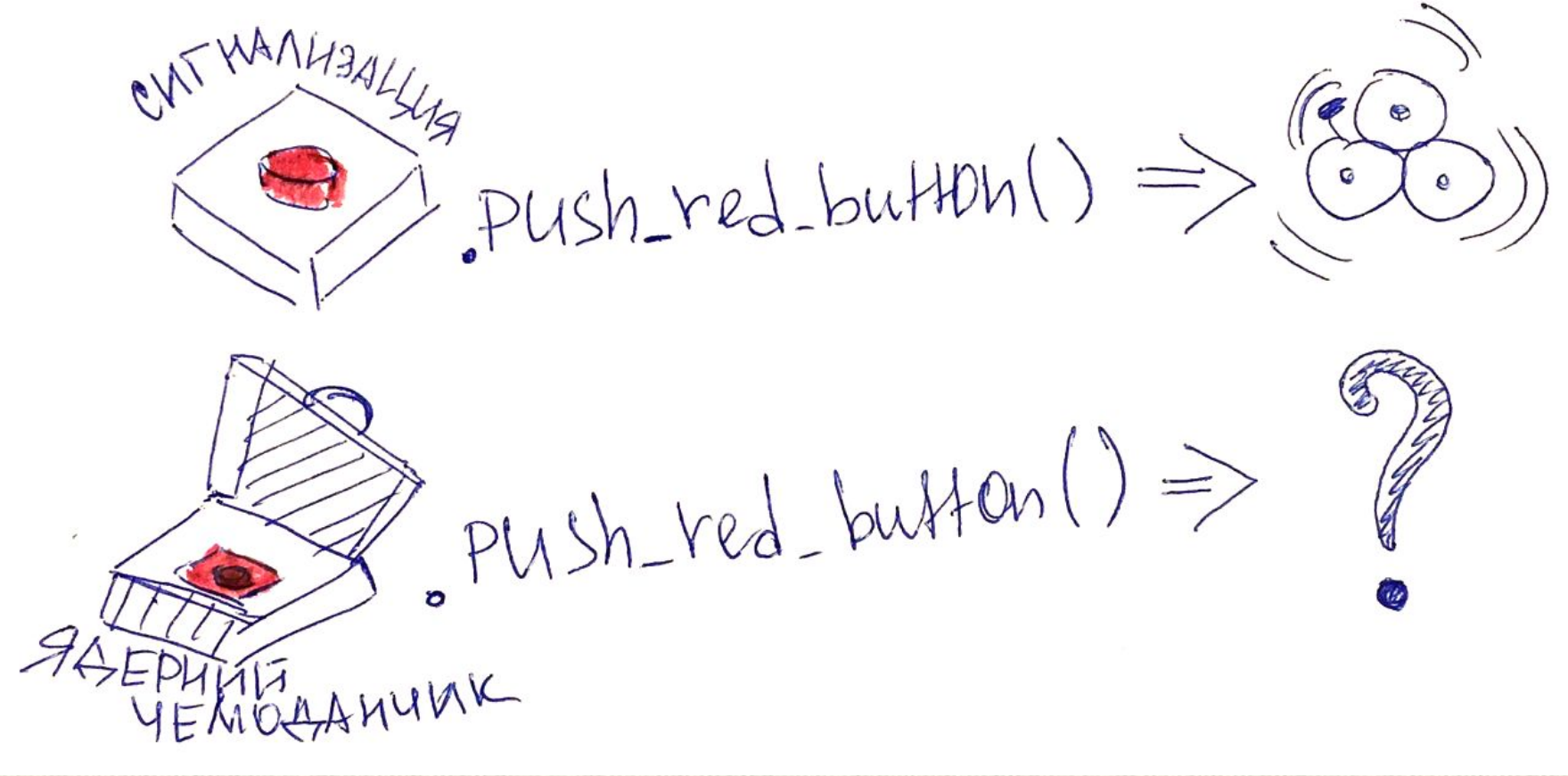
Полиморфизм

У каждого из нас своя реализация,
Главное - кем быть, можно по всякому называться,
Полиморфизм: Человек.сказать(слово)
Каждый может по-своему и это так клево.

#НТР -- Полиморфизм

Полиморфизм

Полиморфизм (греч. «имеющий многие формы») - возможность метода с одним и тем же именем выполнять различные действия в различных классах.



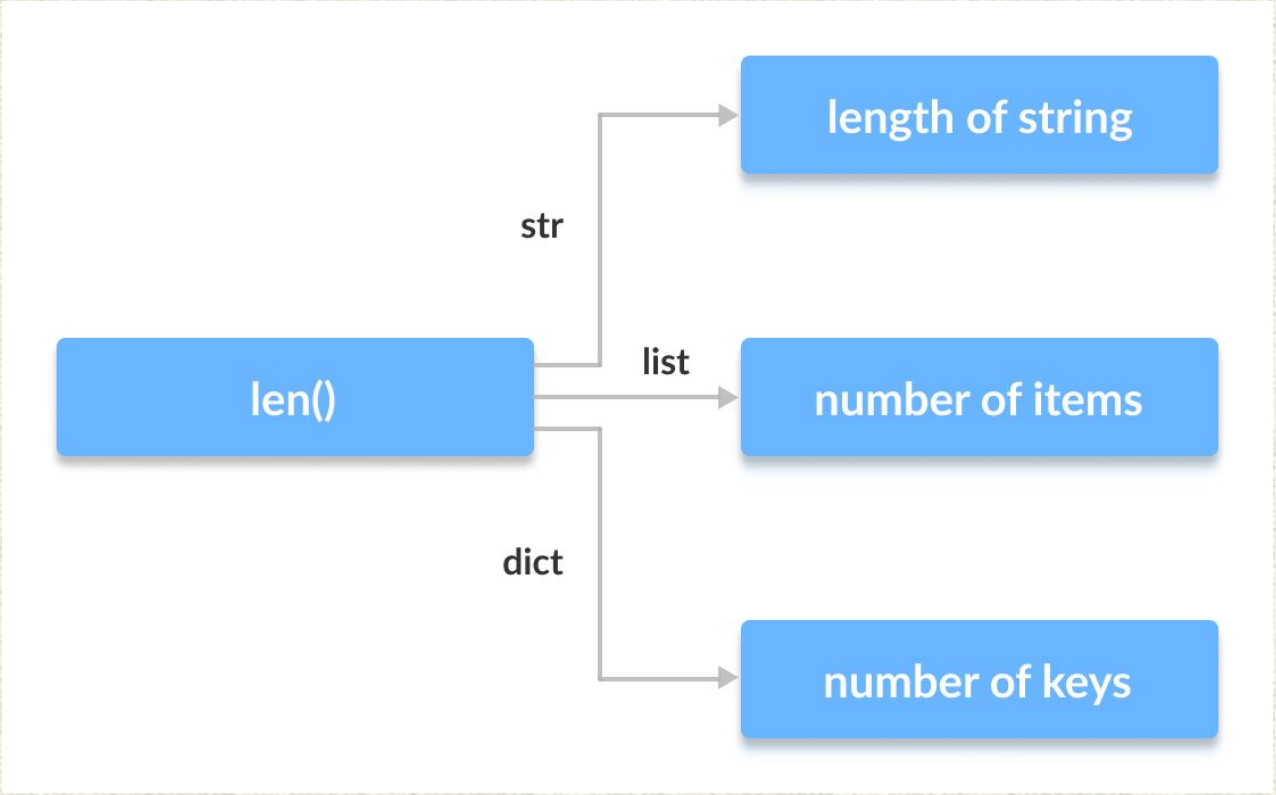
Полиморфизм

```
class T1:
    n=10
    def total(self, N):
      self.total = int(self.n) + int(N)
class T2:
    def total(self,s):
      self.total = len(str(s))

t1 = T1()
t2 = T2()
t1.total(45)
t2.total(45)
print(t1.total) # Вывод: 55
print(t2.total) # Вывод: 2
```

Полиморфизм

print(123 + 456)
print('Вася' + 'Петя')



Полиморфизм

Полиморфизм дает возможность реализовывать так называемые единые интерфейсы для объектов различных классов. Например, разные классы могут предусматривать различный способ вывода той или иной информации объектов. Однако одинаковое название метода вывода позволит не запутать программу, сделать код более ясным.

Полиморфизм

```
class Animal:
    def __init__(self, name): # Конструктор
        self.name = name
    def talk(self): # Абстрактный метод
        pass

class Cat(Animal):
    def talk(self):
        return 'Meow!'

class Dog(Animal):
    def talk(self):
        return 'Woof! Woof!'

animals = [Cat('Missy'), Cat('Mr. Mistoffelees'), Dog('Lassie')]
for ani in animals:
    print ani.name + ': ' + ani.talk()
```

Полиморфизм

В Python полиморфизм также используется для перегрузки стандартных операторов.

__str__() -- для print()
__add__() -- для +
__sub__() -- для -

__lt__() -- для <
__le__() -- для <=
__ne__() -- для !=
__eq__() -- для ==
__ge__() -- для >=
__gt__() -- для >

Полный перечень методов класса object можно почитать здесь.

Атрибуты и методы

```
class Negr:
  color = ‘Black’
  count = 0

   def __init__(self, name, job):
     self.name = name
     self.job = job
    Negr.count += 1

  def info(self):
     print('Имя: ', self.name)
     print( ‘Профессия: ', self.job)
     print( 'Всего: ', Negr.count)

negr_1 = Negr(‘Lui', ‘jazzman')
negr_1.info()
```

```
class Negr:
  color = ‘Black’
  count = 0

   def __init__(self, name, job):
     self.name = name
     self.job = job
    Negr.count += 1

  def __str__(self):
  s='Имя:  ‘+  str(self.name)  +
‘Профессия:  ‘  +  str(self.job)  +
'Всего: ‘ + str(Negr.count)
return s

negr_1 = Negr(‘Lui', ‘jazzman')
print(negr_1)
```