

ООП: Наследование

Практикум по программированию

Наследование

```
class Things:
```

```
    def __init__(self,n,t):
```

```
        self.namething = n
```

```
        self.total = t
```

```
th1 = Things("table", 5)
```

```
th2 = Things("computer", 7)
```

```
print (th1.namething,th1.total) # Вывод: table 5
```

```
print (th2.namething,th2.total) # Вывод: computer 7
```

```
th1.color = "green" # новое свойство объекта th1
```

```
print (th1.color) # Вывод: green
```

```
print (th2.color) # ОШИБКА: у объекта th2 нет свойства color!
```

Наследование

```
class Table:
```

```
    def __init__(self,l,w,h):
```

```
        self.long = l
```

```
        self.width = w
```

```
        self.height = h
```

```
    def outing(self):
```

```
        print (self.long,self.width,self.height)
```

```
class Kitchen(Table):
```

```
    def howplaces(self,n):
```

```
        if n < 2:
```

```
            print ("It is not kitchen table")
```

```
        else:
```

```
            self.places = n
```

```
    def outplases(self):
```

```
        print (self.places)
```

```
t_room1 = Kitchen(2,1,0.5)
```

```
t_room1.outing()
```

```
t_room1.howplaces(5)
```

```
t_room1.outplases()
```

```
t_2 = Table(1,3,0.7)
```

```
t_2.outing()
```

```
t_2.howplaces(8) # ОШИБКА
```

```
class Tree(object):
    def __init__(self, kind, height):
        self.kind = kind
        self.age = 0
        self.height = height

    def info(self):
        """ Метод вывода информации о дереве """
        print ("{} years old {}. {} meters
high.".format(self.age, self.kind, self.height))

    def grow(self):
        """ Метод роста """
        self.age += 1
        self.height += 0.5
```

Надкласс (суперкласс)

Подкласс

Создание экземпляра

```
class FruitTree(Tree):
    def __init__(self, kind, height):
        # Необходимо вызвать метод
инициализации родителя.
        # В Python 3.x это делается при помощи
функции super()
        super().__init__(kind, height)

    def give_fruits(self):
        print ("Collected 20kg of
{s}".format(self.kind))
```

```
tree_2 = FruitTree("apple", 0.7)
# у нас есть доступ к методам родителя
tree_2.info()
tree_2.grow()
# Мы можем использовать свой метод
tree_2.give_fruits()
# А для родительского экземпляра метод
give_fruits() недоступен
tree_1.give_fruits() # Вызовет ошибку
```