

QA
Light

Python

Михаил

Функция `super()`

позволяет сослаться на родителя
экземпляра класса

```
class Foo(object):  
    def __init__(self):  
        self.bar = 42  
class Baz(Foo):  
    def __init__(self):  
        super(Baz, self).__init__()  
        self.param = "test"
```

```
b = Baz()  
print b.bar, b.param  
>>> 42 test
```

«Новые» и «классические» классы

В версиях до 2.2 было невозможно наследовать встроенные классы и классы из модулей расширения.

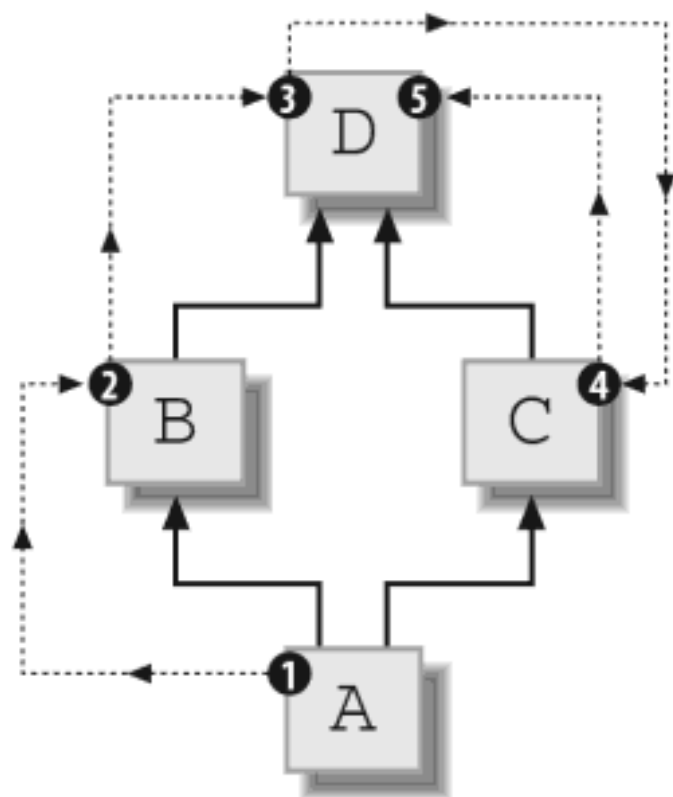
```
class OldStyleClass:  
    pass # класс "старого" типа
```

```
class NewStyleClass(object):  
    pass # и "нового"
```

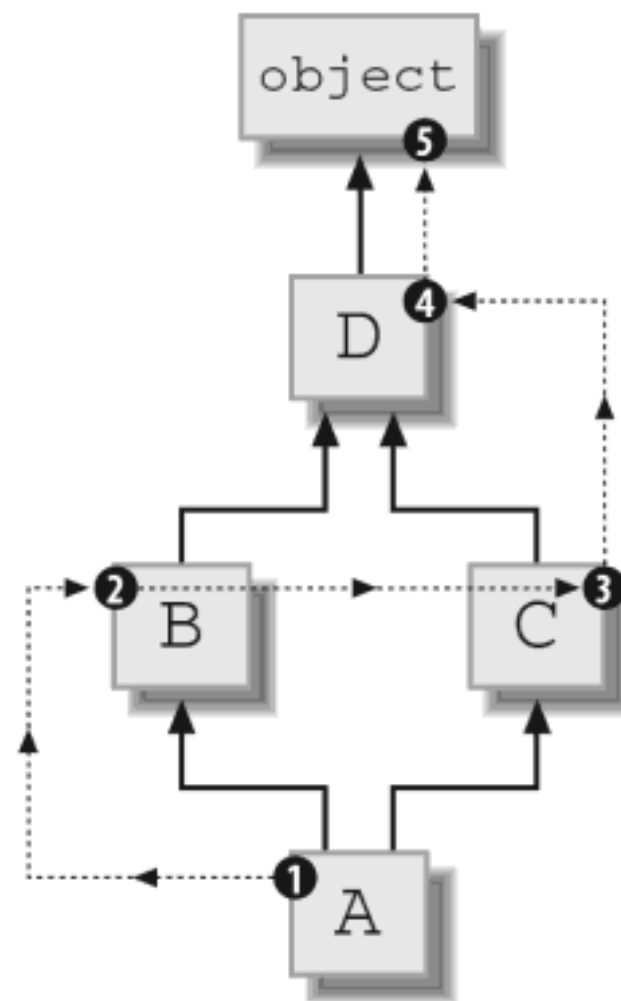
Множественное наследование

```
>>> class A(object): pass
...
>>> A.__mro__
(<class '__main__.A'>, <type 'object'>)
>>> class B(A): pass
...
>>> B.__mro__
(<class '__main__.B'>, <class '__main__.A'>,
<type 'object'>)
>>> class C(A): pass
...
>>> C.__mro__
(<class '__main__.C'>, <class '__main__.A'>,
<type 'object'>)
```

Алгоритм MRO C3



Classic method resolution order



New-style method resolution order

Алгоритм MRO C3

Старый стиль (C, A, object, B)

Новый стиль (C, A, B, object)

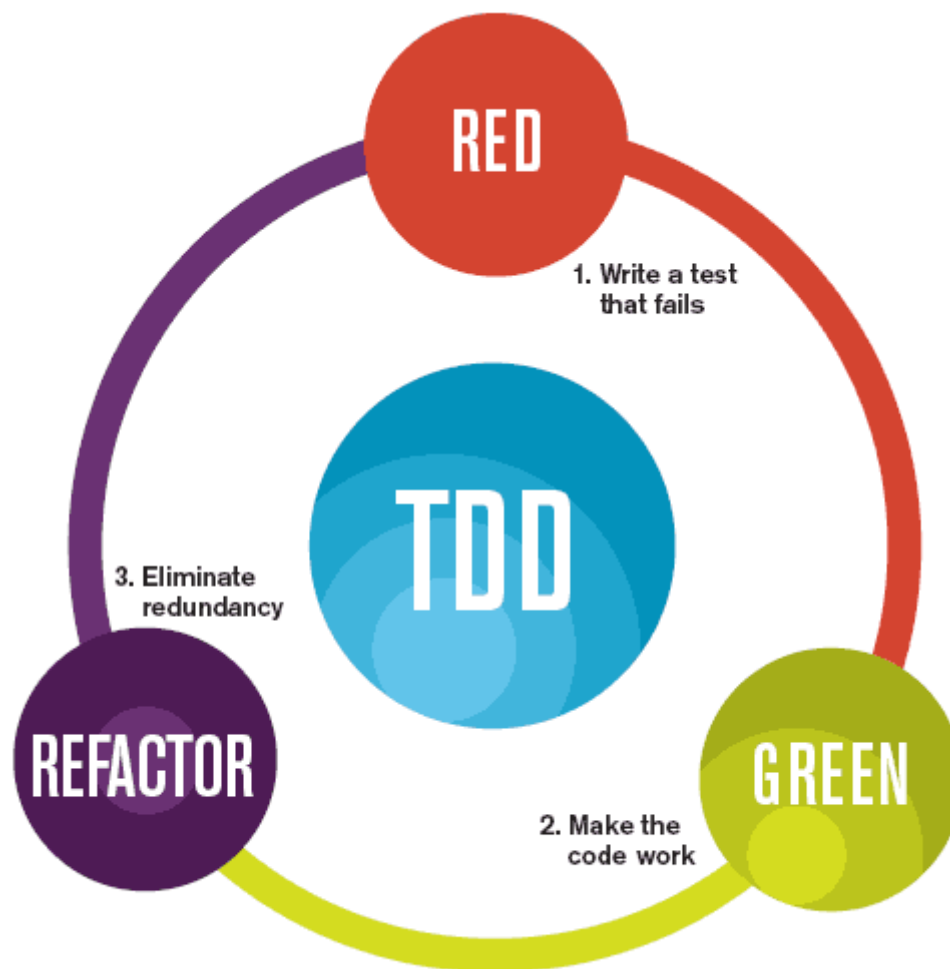
```
>>> class C(A, B): pass
```

```
...
```

```
>>> C.__mro__
```

```
(<class '__main__.C'>, <class '__main__.A'>,  
<class '__main__.B'>, <type 'object'>)
```

TDD



The mantra of Test-Driven Development (TDD) is “red, green, refactor.”

+38 (063) 78-010-78

+38 (097) 78-010-78

+38 (099) 78-010-78

info@qalight.com.ua

qalight.com.ua

TDD nose

```
from unnecessary_math import multiply
```

```
def test_numbers_3_4():  
    assert multiply(3,4) == 12
```

```
def test_strings_a_3():  
    assert multiply('a',3) == 'aaa'
```

```
> nosetests -v test_um_nose.py  
simple_example.test_um_nose.test_numbers_3_4 ...  
ok  
simple_example.test_um_nose.test_strings_a_3 ... ok
```

```
-----  
Ran 2 tests in 0.000s
```

```
OK
```


pytest

```
def test_function():  
    assert f() == 4
```

```
===== FAILURES =====  
_____ test_function _____
```

```
def test_function():  
>     assert f() == 4  
E       assert 3 == 4  
E       + where 3 = f()
```

Python

Киев
ул. Космонавта Комарова 1
НАУ, корп.11

+38 (097) 78 - 010 - 78

+38 (099) 78 - 010 - 78

+38 (063) 78 - 010 - 78

info@qalight.com.ua