

# Объектно- Ориентированное Программирование

---

Для вопросов по курсу:

Иванов Дмитрий Владимирович,

[dmitry.ivanov@moevm.info](mailto:dmitry.ivanov@moevm.info)

Префикс в теме письма [CS\_23XX]



# super()

- Иногда в процессе написания метода в классе-наследнике может понадобиться вызвать метод суперкласса. Это можно сделать через имя суперкласса или через функцию `super()`.
- `class C(B) :`
  - `def method(self, arg) :`
    - `super().method(arg)`    *# То же самое, что:*
    - `super(C, self).method(arg)`



# Вызов конструктора суперкласса

```
class A:  
    def __init__(self, name):  
        self.name = name  
  
class B(A):  
    def __init__(self, name, age):  
        super().__init__(name)  
        self.age = age
```

```
b = B('Olya', 10)  
  
print(b.name, b.age)
```



# Полиморфизм

- Способность функции обрабатывать разные типы данных, если эти данные могут поддерживать соответствующий интерфейс.
- Возможность обрабатывать объекты разных типов одинаковым образом, не задумываясь о типе каждого объекта.



# Перегрузка операторов

- В Python существует возможность переопределения не только методов класса, но и операторов выражений.
- Вы можете создать свой тип данных и определить для его экземпляров операции сложения/сравнения/извлечения среза и т.д.

см. М. Лутц “Изучаем Python”, 4-е издание

Как выглядят операторы в классах:

<https://pythonworld.ru/osnovy/peregruzka-operatorov.html>



## Поле класса **не**изменяемого типа

```
class Counter:  
    counter = 0 # Неизменяемое поле класса
```

**Counter.counter** *# Доступ к полю класса без создания объекта*

**Counter().counter** *# Доступ к полю класса через объект*



## Поле класса **не**изменяемого типа (продолжение)

Объект может присвоить полю другое значение:

```
class Counter:
```

```
    counter = 0 # Неизменяемое поле класса
```

```
a = Counter()
```

```
print(a.counter, Counter.counter) # 0 0
```

```
a.counter = 10
```

```
print(a.counter, Counter.counter) # 10 0
```



## Поле класса **изменяемого** типа

```
class Student:  
    marks = []      # Изменяемое поле класса
```

**Student.marks** *# Доступ к полю класса без создания объекта*

**Student().marks** *# Доступ к полю класса через объект*





# Поле класса изменяемого типа (продолжение)

Объект может изменить поле класса:

```
class Student:  
    marks = [] # Изменяемое поле класса
```

```
Alex = Student()  
print(Alex.marks, Student.marks) # [] []  
Alex.marks.append(5)  
print(Alex.marks, Student.marks) # [5] [5]
```



# Поле класса изменяемого типа (продолжение 2)

Объект может изменить поле класса:

```
class Student:
```

```
    marks = [] # Изменяемое поле класса
```

```
Alex = Student()
```

```
Mary = Student()
```

```
print(Alex.marks, Mary.marks, Student.marks) # [] [] []
```

```
Alex.marks = [] # создаем ссылку на новый объект
```

```
Mary.marks.append(5)
```

```
print(Alex.marks, Mary.marks, Student.marks) # [] [5] [5]
```



# Поля класса при наследовании

```
class A:  
    field = 10
```

```
class B(A):  
    pass
```

# без создания объекта

```
print(B.field) # 10
```

# с созданием объекта

```
b = B()
```

```
print(b.field) # 10
```



# Резюме

- Представьте, что вы пишете свой собственный класс для определенной задачи (не важно, какой именно).
- Что надо сделать, чтобы объекты этого класса **obj** можно было бы передавать в функцию **print(obj)**?
- Что надо сделать, чтобы можно было использовать функцию **len(obj)** с объектами вашего класса?
- Что надо сделать, чтобы объекты вашего класса можно было сравнивать друг с другом и выполнять математические операции?

# DEMO

---



# Источники и полезные ссылки

- Классы в Python, документация  
<https://docs.python.org/3/tutorial/classes.html>
- Операторы языка python, которые вы можете переопределить:  
<https://docs.python.org/3/reference/datamodel.html>
- Курс “Python: основы и применение”  
<https://stepik.org/course/512/syllabus>
- М. Лутц “Изучаем Python”, 4-е издание

Вопросы по курсу можно  
задавать:

---

Иванов Дмитрий Владимирович  
[dmitry.ivanov@moevm.info](mailto:dmitry.ivanov@moevm.info)