



Учебный центр при МГТУ им. Н.Э. Баумана

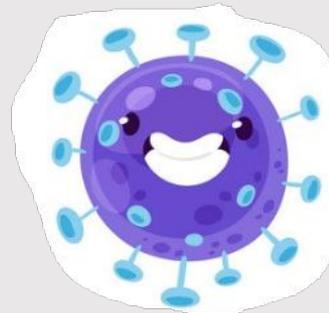
**специалист.ru**

# Программирование на языке Python. Уровень 2. Продвинутый курс

Модуль 1. Понятие класса. Инкапсуляция

# Пара технических моментов на этот курс

- Пожалуйста, указывайте своё имя и фамилию на вебинаре, т.к. у нас в группе есть и тёзки, и однофамильцы :)
- Если подключаетесь с 2 компьютеров, тоже это как-то пометьте (например, *"Иванов Иван Иванович 2й компьютер"*)
- Микрофон и веб-камеру лучше включать только когда хотите что-то сказать, остальное время ставьте на mute
- Обмен файлами – через личный кабинет. Можем дополнительно создать чат для обмена файлами.



# Обо мне



- Леонов Даниил Игоревич
- Окончил МГТУ им. Н.Э.Баумана по специальности «Радиоэлектронные системы»
- Выпускник курса по программированию ЦКО Специалист 2004 года
- Программирую на Python больше 10 лет до управления механикой и роботами
- Использую Python для моделирования обработки сигналов в связке с C++
- Веду спецкурс по созданию новых продуктов в магистратуре «Инновационный менеджмент» в МГУ им. М. В. Ломоносова



# Давайте ПОЗНАКОМИМСЯ

# С чем работаем?

- Python версии 3.8
- Notepad++  
(и это всё)



<https://www.python.org/downloads/>  
<https://notepad-plus-plus.org/downloads/>

# Классы

- С помощью класса мы можем описать сущность, её состояние и поведение в зависимости от состояния
- В языке python класс объявляется следующим образом:

```
class имя(перечень, родительских, классов):  
    тело класса (docstring, методы, атрибуты)
```

например,

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

# Экземпляры класса

- Как только мы определили класс, мы можем создавать экземпляры (объекты, instance) этого класса. У одного класса (типа) может быть сколько угодно объектов того класса (экземпляров)

a = A()

b = A()

- A – это класс, a и b – объекты этого класса

# Методы и атрибуты класса

- Атрибут класса – это переменная, принадлежащая этому классу. Ей может быть и переменная одного из примитивных (встроенных) типов, так и объект какого-то класса.
- Метод класса – это функция, принадлежащая этому классу. В python у метода есть как минимум один аргумент, в который попадает ссылка на использующий его экземпляр. Он всегда идёт на первом месте и его принято обозначать как self.

```
class A():  
    # Это атрибут  
    atr = 0  
    # Это метод  
    def met(self):  
        print('atr =', self.atr)
```

```
a = A()
```

```
b = A()
```

```
a.atr = 1 # Изменит атрибут только для a
```

```
A.atr = 5 # Изменит атрибут для всех объектов A
```

```
a.met()
```

```
b.met()
```

```
A.met(a) # Требуется указать экземпляр как аргумент
```

# Конструктор класса

- Конструктор (constructor, ctor) – это метод, который вызывается при создании объекта.
- Конструкторы бывают с параметрами и без параметров (конструктор по умолчанию, default constructor)
- Конструктор вызывается каждый раз когда создаётся объект класса. Его удобно использовать для инициализации объекта и передачи начальных параметров.
- Для создания конструктора нужно создать метод с зарезервированным названием `__init__`

```
class Duck():
    def __init__(self, name, can_fly = True):
        print("Я родился")
        self.name = name
        self.can_fly = can_fly
    def speak(self):
        print("Я -", self.name)
    def fly(self):
        if self.can_fly == True:
            print("плавёт")
        else:
            print("не плавёт")
a = Duck("Валера", True)
b = Duck("Серёжа", False)
```

# Деструктор

- Аналогично конструктору, деструктор вызывается перед уничтожением объекта
- Обозначается зарезервированным словом `__del__`
- Всегда принимает только один параметр (`self`)
- В силу особенностей автоматической сборки мусора в `python`, на практике используется не так часто.

```
def __del__(self):  
    print("пока")  
del a  
del b
```

# Выбирайте Центр «Специалист» – крупнейший учебный центр России!

info@specialist.

RU (495)

232-32-16