



**DS**  
**Программирование**  
**Python**

# Введение в ООП

---



# Введение



*В программе хранится список. Какой символ нужно поставить после названия списка, чтобы вызвать команду `append()`?*



*В программе хранится список. Какой символ нужно поставить после названия списка, чтобы вызвать команду `append()`?*

*Точку, например, `my_list.append(2)`*



- ◆ *В программе хранится список. Какой символ нужно поставить после названия списка, чтобы вызвать команду `append()`?*  
*Точку, например, `my_list.append(2)`*
- ◆ *Как называются команды, которые мы вызываем с помощью точки?*



◆ *В программе хранится список. Какой символ нужно поставить после названия списка, чтобы вызвать команду `append()`?*

*Точку, например, `my_list.append(2)`*

◆ *Как называются команды, которые мы вызываем с помощью точки?*  
Методы



- ◆ **В программе хранится список. Какой символ нужно поставить после названия списка, чтобы вызвать команду `append()`?**  
Точку, например, `my_list.append(2)`
- ◆ **Как называются команды, которые мы вызываем с помощью точки?**  
Методы
- ◆ **Какая функция позволяет узнать тип значения, которое лежит в переменной?**



- ◆ **В программе хранится список. Какой символ нужно поставить после названия списка, чтобы вызвать команду `append()`?**  
Точку, например, `my_list.append(2)`
- ◆ **Как называются команды, которые мы вызываем с помощью точки?**  
Методы
- ◆ **Какая функция позволяет узнать тип значения, которое лежит в переменной?**  
Функция `type()`





- ◆ **В программе хранится список. Какой символ нужно поставить после названия списка, чтобы вызвать команду `append()`?**  
Точку, например, `my_list.append(2)`
- ◆ **Как называются команды, которые мы вызываем с помощью точки?**  
Методы
- ◆ **Какая функция позволяет узнать тип значения, которое лежит в переменной?**  
Функция `type()`
- ◆ **Какими основными элементами оперирует структурное программирование?**



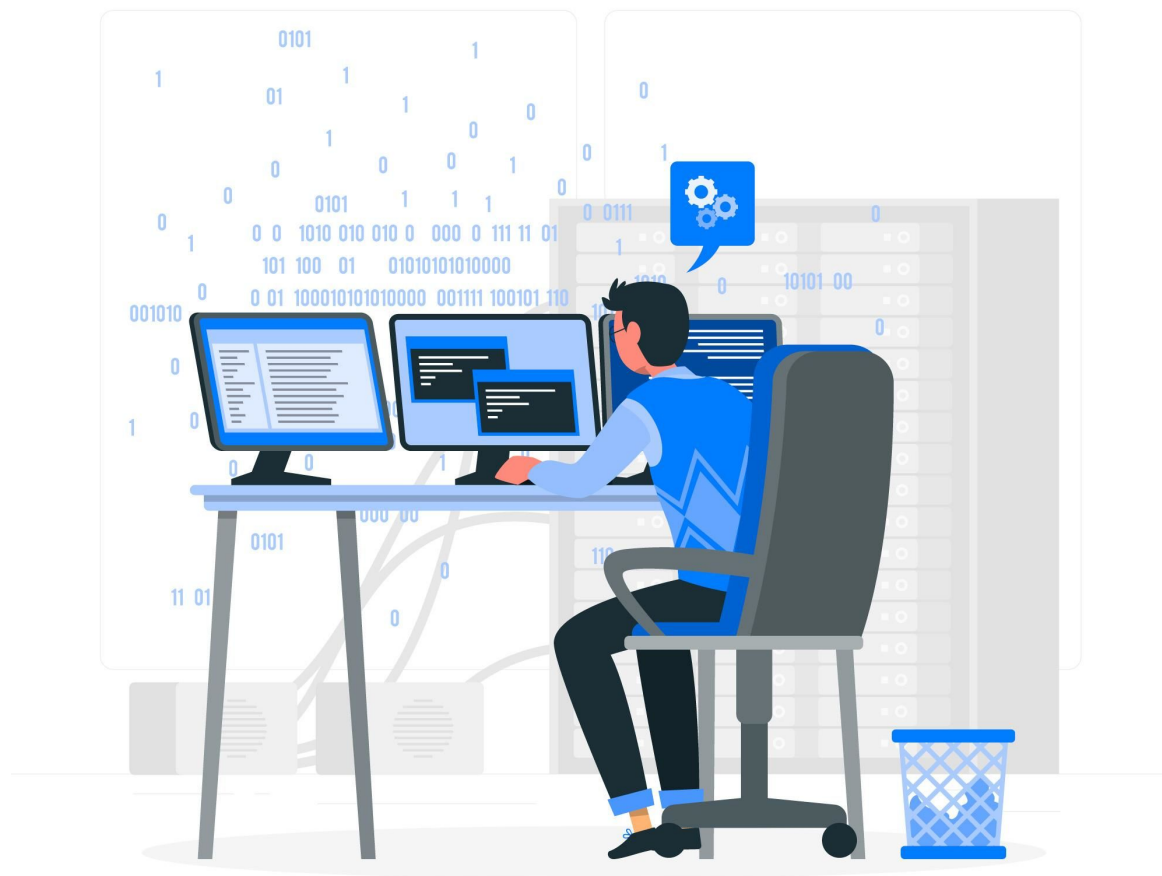
- ◆ **В программе хранится список. Какой символ нужно поставить после названия списка, чтобы вызвать команду `append()`?**  
Точка, например, `my_list.append(2)`
- ◆ **Как называются команды, которые мы вызываем с помощью точки?**  
Методы
- ◆ **Какая функция позволяет узнать тип значения, которое лежит в переменной?**  
Функция `type()`
- ◆ **Какими основными элементами оперирует структурное программирование?**  
Условные операторы, циклы, функции



- ◆ **В программе хранится список. Какой символ нужно поставить после названия списка, чтобы вызвать команду `append()`?**  
Точка, например, `my_list.append(2)`
- ◆ **Как называются команды, которые мы вызываем с помощью точки?**  
Методы
- ◆ **Какая функция позволяет узнать тип значения, которое лежит в переменной?**  
Функция `type()`
- ◆ **Какими основными элементами оперирует структурное программирование?**  
Условные операторы, циклы, функции
- ◆ **В виде чего представляется программа в функциональном программировании?**



- ◆ **В программе хранится список. Какой символ нужно поставить после названия списка, чтобы вызвать команду `append()`?**  
Точку, например, `my_list.append(2)`
- ◆ **Как называются команды, которые мы вызываем с помощью точки?**  
Методы
- ◆ **Какая функция позволяет узнать тип значения, которое лежит в переменной?**  
Функция `type()`
- ◆ **Какими основными элементами оперирует структурное программирование?**  
Условные операторы, циклы, функции
- ◆ **В виде чего представляется программа в функциональном программировании?**  
Программа представляется как процесс вычисления функций в их математическом понимании





- » Ввод-вывод данных
- » Условные операторы
- » Циклы
- » Базовые встроенные структуры данных
- » Функции
- » Работа с текстовыми файлами
- » Генераторные выражения
- » Функциональное программирование
- » Обработка ошибок и исключений



















## «Моя машина»

Марка – Toyota

Модель – Corolla

Цвет – синий

Текущая скорость – 0

Завести двигатель

Увеличить/уменьшить скорость

Повернуть



Клас

с

«Моя машина»

Марка – Toyota

Модель – Corolla

Цвет – синий

Текущая скорость – 0

Завести двигатель

Увеличить/уменьшить скорость

Повернуть





**Класс** – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные объекты («Автомобиль»)



Клас

с

Объект

«Моя машина»

Марка – Toyota

Модель – Corolla

Цвет – синий

Текущая скорость – 0

Завести двигатель

Увеличить/уменьшить скорость

Повернуть





**Класс** – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные объекты («Автомобиль»)

**Объект** – конкретный экземпляр, созданный на основании указанного класса («Моя машина»)







Клас

с

Объект

«Моя машина»

**Марка – Toyota**

**Модель – Corolla**

**Цвет – синий**

**Текущая скорость – 0**

Свойства

Завести двигатель

Увеличить/уменьшить скорость

Повернуть





**Свойства** – характеристики, имеющиеся у каждого объекта класса  
(марка, модель, цвет, текущая скорость)



Клас

с

Объект

«Моя машина»

Марка – Toyota

Модель – Corolla

Цвет – синий

Текущая скорость – 0

Свойства

**Завести двигатель**

**Увеличить/уменьшить  
скорость Повернуть**

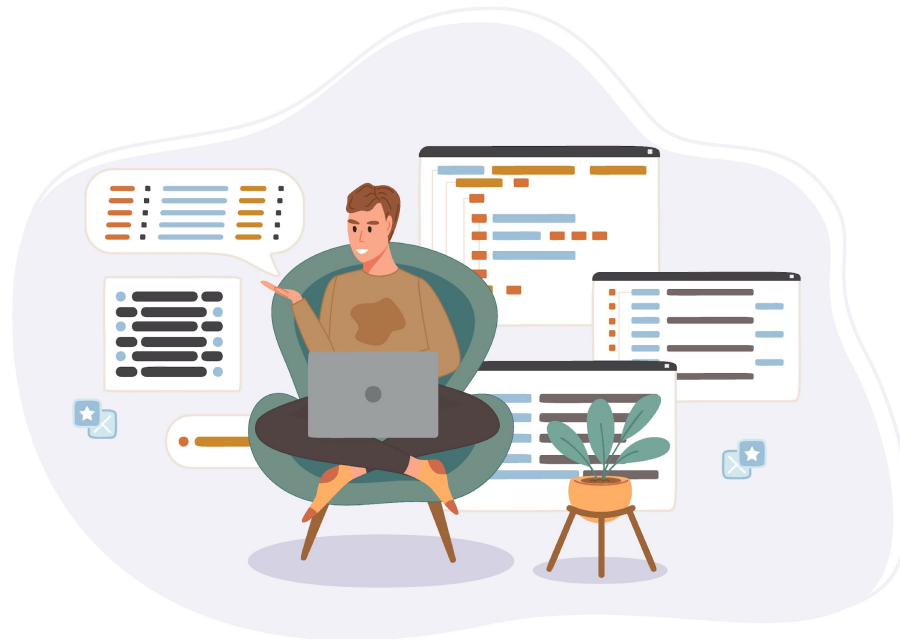
Методы





**Свойства** – характеристики, имеющиеся у каждого объекта класса (марка, модель, цвет, текущая скорость)

**Методы** – функции, позволяющие взаимодействовать со свойствами класса (завести двигатель, увеличить скорость, уменьшить скорость, повернуть)









## Свойства

Масса автомобиля  
Максимальная скорость  
Мощность двигателя

## Методы

Завести двигатель  
Переключить  
передачу



**Абстракция** – использование только тех характеристик, которые действительно нужны при проектировке конкретного класса.









```
class Car:
```

*# создаем класс "Автомобиль"*



```
class Car:
```

```
    # описываем свойства класса
```

*# создаем класс "Автомобиль"*



```
class Car:                                     # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                            # марка
```



```
class Car:                                     # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                           # марка  
    model = 'Corolla'                         # модель
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                         # цвет
```



```
class Car:                                     # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                           # марка  
    model = 'Corolla'                         # модель  
    color = 'blue'                            # цвет  
    speed = 0                                 # скорость
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                        # цвет  
    speed = 0                             # скорость  
my_car_1 = Car()                          # создаем экземпляр класса Car
```





```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                         # цвет  
    speed = 0                              # скорость  
my_car_1 = Car()                           # создаем экземпляр класса Car  
# выводим информацию об экземпляре  
print(my_car_1)
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                    # модель  
    color = 'blue'                        # цвет  
    speed = 0                             # скорость  
my_car_1 = Car()                          # создаем экземпляр класса Car  
# выводим информацию об экземпляре  
print(my_car_1)
```

```
>>> <__main__.Car object at 0x0000016C8BC6F810>
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                         # цвет  
    speed = 0                              # скорость  
my_car_1 = Car()                           # создаем экземпляр класса Car  
# выводим тип экземпляра класса  
print(type(my_car_1))
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                         # цвет  
    speed = 0                              # скорость  
my_car_1 = Car()                           # создаем экземпляр класса Car  
# выводим тип экземпляра класса  
print(type(my_car_1))
```

```
>>> <class '__main__.Car'>
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                        # цвет  
    speed = 0                             # скорость  
my_car_1 = Car()                          # создаем экземпляр класса Car  
# выводим значение свойства model  
print(f'Модель автомобиля:{my_car_1.model}')
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                        # цвет  
    speed = 0                             # скорость  
my_car_1 = Car()                          # создаем экземпляр класса Car  
# выводим значение свойства model  
print(f'Модель автомобиля:{my_car_1.model}')
```

```
>>> Модель автомобиля:Corolla
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                        # цвет  
    speed = 0                             # скорость  
my_car_1 = Car()                          # создаем экземпляр класса Car  
# выводим на экран все свойства объекта my_car_1  
print(f'Марка автомобиля:{my_car_1.mark}')  
print(f'Модель автомобиля:{my_car_1.model}')  
print(f'Цвет автомобиля:{my_car_1.color}')  
print(f'Текущая скорость:{my_car_1.speed}')
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                        # цвет  
    speed = 0                             # скорость  
my_car_1 = Car()                          # создаем экземпляр класса Car  
# выводим на экран все свойства объекта my_car_1  
print(f'Марка автомобиля:{my_car_1.mark}')  
print(f'Модель автомобиля:{my_car_1.model}')  
print(f'Цвет автомобиля:{my_car_1.color}')  
print(f'Текущая скорость:{my_car_1.speed}')
```

```
>>> Марка автомобиля:Toyota  
      Модель автомобиля:Corolla  
      Цвет автомобиля:blue  
      Текущая скорость:0
```





```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                         # цвет  
    speed = 0                              # скорость  
my_car_1 = Car()                           # создаем экземпляр класса Car  
# выводим на экран все свойства объекта my_car_1  
print(f'Марка автомобиля:{my_car_1.mark}')  
print(f'Модель автомобиля:{my_car_1.model}')  
print(f'Цвет автомобиля:{my_car_1.color}')  
print(f'Текущая скорость:{my_car_1.speed}')  
my_car_1.color = 'red'                     # меняем значение свойства color
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                         # цвет  
    speed = 0                             # скорость  
my_car_1 = Car()                          # создаем экземпляр класса Car  
# выводим на экран все свойства объекта my_car_1  
print(f'Марка автомобиля:{my_car_1.mark}')  
print(f'Модель автомобиля:{my_car_1.model}')  
print(f'Цвет автомобиля:{my_car_1.color}')  
print(f'Текущая скорость:{my_car_1.speed}')  
my_car_1.color = 'red'                    # меняем значение свойства color  
my_car_2 = Car()                          # создаем еще один экземпляр класса Car
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                        # цвет  
    speed = 0                             # скорость  
my_car_1 = Car()                          # создаем экземпляр класса Car  
# выводим на экран все свойства объекта my_car_1  
print(f'Марка автомобиля:{my_car_1.mark}')  
print(f'Модель автомобиля:{my_car_1.model}')  
print(f'Цвет автомобиля:{my_car_1.color}')  
print(f'Текущая скорость:{my_car_1.speed}')  
my_car_1.color = 'red'                    # меняем значение свойства color  
my_car_2 = Car()                          # создаем еще один экземпляр класса Car  
# выводим на экран все свойства объекта my_car_2  
print(f'Марка автомобиля:{my_car_2.mark}')
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                         # цвет  
    speed = 0                              # скорость  
my_car_1 = Car()                           # создаем экземпляр класса Car  
# выводим на экран все свойства объекта my_car_1  
print(f'Марка автомобиля:{my_car_1.mark}')  
print(f'Модель автомобиля:{my_car_1.model}')  
print(f'Цвет автомобиля:{my_car_1.color}')  
print(f'Текущая скорость:{my_car_1.speed}')  
my_car_1.color = 'red'                     # меняем значение свойства color  
my_car_2 = Car()                           # создаем еще один экземпляр класса Car  
# выводим на экран все свойства объекта my_car_2  
print(f'Марка автомобиля:{my_car_2.mark}')  
print(f'Модель автомобиля:{my_car_2.model}')
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                         # цвет  
    speed = 0                              # скорость  
my_car_1 = Car()                           # создаем экземпляр класса Car  
# выводим на экран все свойства объекта my_car_1  
print(f'Марка автомобиля:{my_car_1.mark}')  
print(f'Модель автомобиля:{my_car_1.model}')  
print(f'Цвет автомобиля:{my_car_1.color}')  
print(f'Текущая скорость:{my_car_1.speed}')  
my_car_1.color = 'red'                     # меняем значение свойства color  
my_car_2 = Car()                           # создаем еще один экземпляр класса Car  
# выводим на экран все свойства объекта my_car_2  
print(f'Марка автомобиля:{my_car_2.mark}')  
print(f'Модель автомобиля:{my_car_2.model}')  
print(f'Цвет автомобиля:{my_car_2.color}')
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                         # цвет  
    speed = 0                              # скорость  
my_car_1 = Car()                           # создаем экземпляр класса Car  
# выводим на экран все свойства объекта my_car_1  
print(f'Марка автомобиля:{my_car_1.mark}')  
print(f'Модель автомобиля:{my_car_1.model}')  
print(f'Цвет автомобиля:{my_car_1.color}')  
print(f'Текущая скорость:{my_car_1.speed}')  
my_car_1.color = 'red'                     # меняем значение свойства color  
my_car_2 = Car()                           # создаем еще один экземпляр класса Car  
# выводим на экран все свойства объекта my_car_2  
print(f'Марка автомобиля:{my_car_2.mark}')  
print(f'Модель автомобиля:{my_car_2.model}')  
print(f'Цвет автомобиля:{my_car_2.color}')  
print(f'Текущая скорость:{my_car_2.speed}')
```



```
class Car:                                # создаем класс "Автомобиль"  
    # описываем свойства класса  
    mark = 'Toyota'                       # марка  
    model = 'Corolla'                     # модель  
    color = 'blue'                         # цвет  
    speed = 0                             # скорость  
my_car_1 = Car()                          # создаем экземпляр класса Car  
# выводим на экран все свойства объекта my_car_1  
print(f'Марка автомобиля:{my_car_1.mark}')  
print(f'Модель автомобиля:{my_car_1.model}')  
print(f'Цвет автомобиля:{my_car_1.color}')  
print(f'Текущая скорость:{my_car_1.speed}')  
my_car_1.color = 'red'                    # меняем значение свойства color  
my_car_2 = Car()                          # создаем еще один экземпляр класса Car  
# выводим на экран все свойства объекта my_car_2  
print(f'Марка автомобиля:{my_car_2.mark}')  
print(f'Модель автомобиля:{my_car_2.model}')  
print(f'Цвет автомобиля:{my_car_2.color}')  
print(f'Текущая скорость:{my_car_2.speed}')  
my_car_2.model = 'Tundra'                 # меняем значение свойства model
```



## «Автопарк спортивных машин»

Создайте класс **SportCar** и реализуйте внутри следующие свойства:

- марка автомобиля (строка)
- модель автомобиля (строка)
- мощность двигателя (целое число)
- максимальная скорость (целое число)

Создайте 3 экземпляра класса **SportCar**, для каждого из них укажите соответствующие свойства. Значения свойств вводятся с клавиатуры. Сохраните все созданные объекты в список `cars`, а затем выведете на экран для каждого объекта значения всех свойств, каждое с новой строки.





# Задание № 1

```
class SportCar:
```

```
# создаем класс SportCar
```



```
class SportCar:                                # создаем класс SportCar
    mark = ''                                   # марка
    model = ''                                  # модель
    engine_power = 0                            # мощность двигателя
    max_speed = 0                               # максимальная скорость
cars = []
```



```
class SportCar:
    mark = ''
    model = ''
    engine_power = 0
    max_speed = 0
cars = []
```

*# создаем класс SportCar*  
*# марка*  
*# модель*  
*# мощность двигателя*  
*# максимальная скорость cars = []*  
*# создаем список cars для хранения объектов класс SportCar*



```
class SportCar:
    mark = ''
    model = ''
    engine_power = 0
    max_speed = 0
cars = []
for _ in range(3):
```

*# создаем класс SportCar*  
*# марка*  
*# модель*  
*# мощность двигателя*  
*# максимальная скорость cars = []*  
*# создаем список cars для хранения объектов класса SportCar*  
*# цикл для создания трех экземпляров класса Car*



```
class SportCar:
    mark = ''
    model = ''
    engine_power = 0
    max_speed = 0
cars = []
for _ in range(3):
    car = SportCar()
```

*# создаем класс SportCar*  
*# марка*  
*# модель*  
*# мощность двигателя*  
*# максимальная скорость cars = []*  
*# создаем список cars для хранения объектов класса SportCar*  
*# цикл для создания трех экземпляров класса Car*  
*# создаем экземпляр класса Car*



```
class SportCar:                                # создаем класс SportCar
    mark = ''                                   # марка
    model = ''                                  # модель
    engine_power = 0                           # мощность двигателя
    max_speed = 0                               # максимальная скорость
cars = []                                       # создаем список cars для хранения объектов класса SportCar
for _ in range(3):                             # цикл для создания трех экземпляров класса Car
    car = SportCar()                           # создаем экземпляр класса Car
    # задаем значения свойств объекта
    car.mark = input('Введите марку:')
    car.model = input('Введите модель:')
    car.engine_power = int(input('Введите мощность двигателя:'))
    car.max_speed = int(input('Введите макс. скорость:'))
```



```
class SportCar:                                # создаем класс SportCar
    mark = ''                                  # марка
    model = ''                                 # модель
    engine_power = 0                           # мощность двигателя
    max_speed = 0                              # максимальная скорость
cars = []                                     # создаем список cars для хранения объектов класса SportCar
for _ in range(3):                             # цикл для создания трех экземпляров класса Car
    car = SportCar()                           # создаем экземпляр класса Car
    # задаем значения свойств объекта
    car.mark = input('Введите марку:')
    car.model = input('Введите модель:')
    car.engine_power = int(input('Введите мощность двигателя:'))
    car.max_speed = int(input('Введите макс. скорость:'))
    cars.append(car)                            # добавляем экземпляр класса Car в список
print('-----')
```



```
class SportCar:                                # создаем класс SportCar
    mark = ''                                   # марка
    model = ''                                  # модель
    engine_power = 0                            # мощность двигателя
    max_speed = 0                               # максимальная скорость
cars = []                                       # создаем список cars для хранения объектов класса SportCar
for _ in range(3):                             # цикл для создания трех экземпляров класса Car
    car = SportCar()                           # создаем экземпляр класса Car
    # задаем значения свойств объекта
    car.mark = input('Введите марку:')
    car.model = input('Введите модель:')
    car.engine_power = int(input('Введите мощность двигателя:'))
    car.max_speed = int(input('Введите макс. скорость:'))
    cars.append(car)                            # добавляем экземпляр класса Car в список
    print('-----')
print('Машины в автопарке:') # выводим на экран информацию о свойствах всех машин
```





```
class SportCar:                                # создаем класс SportCar
    mark = ''                                   # марка
    model = ''                                  # модель
    engine_power = 0                            # мощность двигателя
    max_speed = 0                               # максимальная скорость
cars = []                                       # создаем список cars для хранения объектов класса SportCar
for _ in range(3):                             # цикл для создания трех экземпляров класса Car
    car = SportCar()                           # создаем экземпляр класса Car
    # задаем значения свойств объекта
    car.mark = input('Введите марку:')
    car.model = input('Введите модель:')
    car.engine_power = int(input('Введите мощность двигателя:'))
    car.max_speed = int(input('Введите макс. скорость:'))
    cars.append(car)                            # добавляем экземпляр класса Car в список
    print('-----')
print('Машины в автопарке:') # выводим на экран информацию о свойствах всех машин
for car in cars:
```



```
class SportCar:                                # создаем класс SportCar
    mark = ''                                  # марка
    model = ''                                  # модель
    engine_power = 0                            # мощность двигателя
    max_speed = 0                               # максимальная скорость
cars = []                                       # создаем список cars для хранения объектов класса SportCar
for _ in range(3):                             # цикл для создания трех экземпляров класса Car
    car = SportCar()                            # создаем экземпляр класса Car
    # задаем значения свойств объекта
    car.mark = input('Введите марку:')
    car.model = input('Введите модель:')
    car.engine_power = int(input('Введите мощность двигателя:'))
    car.max_speed = int(input('Введите макс. скорость:'))
    cars.append(car)                            # добавляем экземпляр класса Car в список
    print('-----')
print('Машины в автопарке:') # выводим на экран информацию о свойствах всех машин
for car in cars:
    print(f'Марка: {car.mark}')
    print(f'Модель: {car.model}')
    print(f'Мощность двигателя: {car.engine_power}')
    print(f'Макс. скорость: {car.max_speed}')
    print('-----')
```



```
class SportCar:                                # создаем класс SportCar
    mark = ''                                   # марка
    model = ''                                  # модель
    engine_power = 0                           # мощность двигателя
    max_speed = 0                              # максимальная скорость
cars = []                                       # создаем список cars для хранения объектов класса SportCar
for _ in range(3):                             # цикл для создания трех экземпляров класса Car
    car = SportCar()                           # создаем экземпляр класса Car
    # задаем значения свойств объекта
    car.mark = input('Введите марку:')
    car.model = input('Введите модель:')
    car.engine_power = int(input('Введите мощность двигателя:'))
    car.max_speed = int(input('Введите макс. скорость:'))
    cars.append(car)                           # добавляем экземпляр класса Car в список
    print('-----')
print('Машины в автопарке:') # выводим на экран информацию о свойствах всех машин
for car in cars:
    print(f'Марка: {car.mark}')
    print(f'Модель: {car.model}')
    print(f'Мощность двигателя: {car.engine_power}')
    print(f'Макс. скорость: {car.max_speed}')
    print('-----')
```



ИТОГ



## *Что такое класс в контексте ООП?*



### **Что такое класс в контексте ООП?**

*Класс – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные объекты*



## Что такое класс в контексте ООП?

*Класс – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные*

*объекты.*

---

Что такое объект класса?



### Что такое класс в контексте ООП?

Класс – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные

### объекты. Что такое объект класса?

Объект класса – конкретный экземпляр, созданный на основании указанного класса





## Что такое класс в контексте ООП?

Класс – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные

## объекты. Что такое объект класса?

Объект класса – конкретный экземпляр, созданный на основании указанного класса

## Что такое свойство класса?



### Что такое класс в контексте ООП?

Класс – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные

### объекты. Что такое объект класса?

Объект класса – конкретный экземпляр, созданный на основании указанного класса

### Что такое свойство класса?

Свойства класса – характеристики, имеющиеся у каждого объекта класса



## **Что такое класс в контексте ООП?**

*Класс – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные*

## **Что такое объект класса?**

*Объект класса – конкретный экземпляр, созданный на основании указанного класса*

## **Что такое свойство класса?**

*Свойства класса – характеристики, имеющиеся у каждого объекта класса*

## **Что такое методы класса?**



### **Что такое класс в контексте ООП?**

*Класс – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные*

### **Что такое объект класса?**

*Объект класса – конкретный экземпляр, созданный на основании указанного класса*

### **Что такое свойство класса?**

*Свойства класса – характеристики, имеющиеся у каждого объекта класса*

### **Что такое методы класса?**

*Методы класса – функции, позволяющие взаимодействовать со свойствами класса*



### **Что такое класс в контексте ООП?**

*Класс – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные*

### **Что такое объект класса?**

*Объект класса – конкретный экземпляр, созданный на основании указанного класса*

### **Что такое свойство класса?**

*Свойства класса – характеристики, имеющиеся у каждого объекта класса*

### **Что такое методы класса?**

*Методы класса – функции, позволяющие взаимодействовать со свойствами класса*

### **Что такое абстракция в контексте ООП?**



### Что такое класс в контексте ООП?

Класс – шаблон с описанными внутри характеристиками и возможностями, на основании которого можно создавать конкретные

### Что такое объект класса?

Объект класса – конкретный экземпляр, созданный на основании указанного класса

### Что такое свойство класса?

Свойства класса – характеристики, имеющиеся у каждого объекта класса

### Что такое методы класса?

Методы класса – функции, позволяющие взаимодействовать со свойствами класса

### Что такое абстракция в контексте ООП?

Абстракция – один из принципов ООП, подразумевающий использование только тех характеристик, которые действительно нужны при проектировке конкретного класса



**DS**  
**Программирование**  
**Python**

**Спасибо за внимание!**

---