

Plane

```
class Plane:  
    def start_engine(self):  
        engine_on = True # К сожалению, не сработает  
    def fly_to(self, city):  
        if engine_on: # Ошибка NameError  
            print("Летим в город {}".format(city))  
        else: print("Самолет не заправлен, никуда не летим")  
c = Plane()  
c.start_engine()  
c.fly_to('Владивосток')
```

```
class Plane :  
    def start_engine(self):  
        self.engine_on = True # сработает  
    def drive_to(self, city):  
        if self.engine_on: # Ошибка NameError  
            print(" Летим в город {}".format(city))  
        else: print("Самолет не заправлен, никуда не летим")  
c = Plane()  
c.start_engine()  
c.fly_to('Владивосток') # Летим в город Владивосток.  
plane2 = Plane()  
plane2.fly_to('Лиссабон')
```

```
class Plane :  
    def __init__(self):  
        self.engine_on = False  
    def start_engine(self):  
        self.engine_on = True # сработает  
    def fly_to(self, city):  
        if self.engine_on: # Ошибка NameError  
            print("{} Самолет летит в город {}".format(self.color, city))  
        else: print("Самолет не заправлен, никуда не летим")  
plane1 = Plane()  
plane1.start_engine()  
plane1.fly_to('Владивосток') # Летим в город Владивосток.  
plane2 = Plane()  
plane2.fly_to('Лиссабон') # Самолет не заправлен, никуда не летим.
```

```
class Plane :  
    def __init__(self):  
        self.engine_on = False  
        self.color = color  
    def start_engine(self, color ):  
        self.engine_on = True  
    def fly_to(self, city):  
        if self.engine_on:  
            print("{} Самолет летит в город {}".format(self.color, city))  
        else: print("{} Самолет не заправлен, никуда не летим {}".format(self.color))
```

```
plane1 = Plane("RUS") # Создали российский самолет
plane2 = Plane("uk") # И еще один английский
plane1.start_engine() # Обратите внимание, что мы заправили только один
# тот, на который ссылается plane1 ("RUS").
# plane2 -- это другой объект, он не изменится.
plane1.fly_to("Владивосток") # российский самолет летит в город
# Владивосток.
plane2.fly_to("") # английский самолет не заправлен, никуда не летит.
```

Инкапсуляция

Такая технология сокрытия информации о внутреннем устройстве объекта за внешним интерфейсом из методов называется **инкапсуляцией**. Надо стараться делать интерфейс методов достаточно полным. Тогда вы, как и другие программисты, будете пользоваться этими методами, а изменения в атрибутах не будут расползаться по коду, использующему ваш класс. Кроме того, инкапсуляция позволяет шире использовать такое понятие, как полиморфизм. О нем мы поговорим на следующем уроке.