

ДЕТСКАЯ ОНЛАЙН-ШКОЛА ПРОГРАММИРОВАНИЯ

# HELLO WORLD



#ШПАРГАЛОЧКИ



# ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON

## Базовый уровень

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити в Telegram: [https://t.me/hw\\_school](https://t.me/hw_school)





# Flappy bird





**Полиморфизм** в объектно-ориентированном программировании позволяет переопределять родительские методы в дочерних классах. Например, если в дочернем классе не определен `__init__`, то при создании объекта будет вызван конструктор родителя. Но если метод определен в дочернем классе, то он словно “перекрывает” собой соответствующий метод родителя.



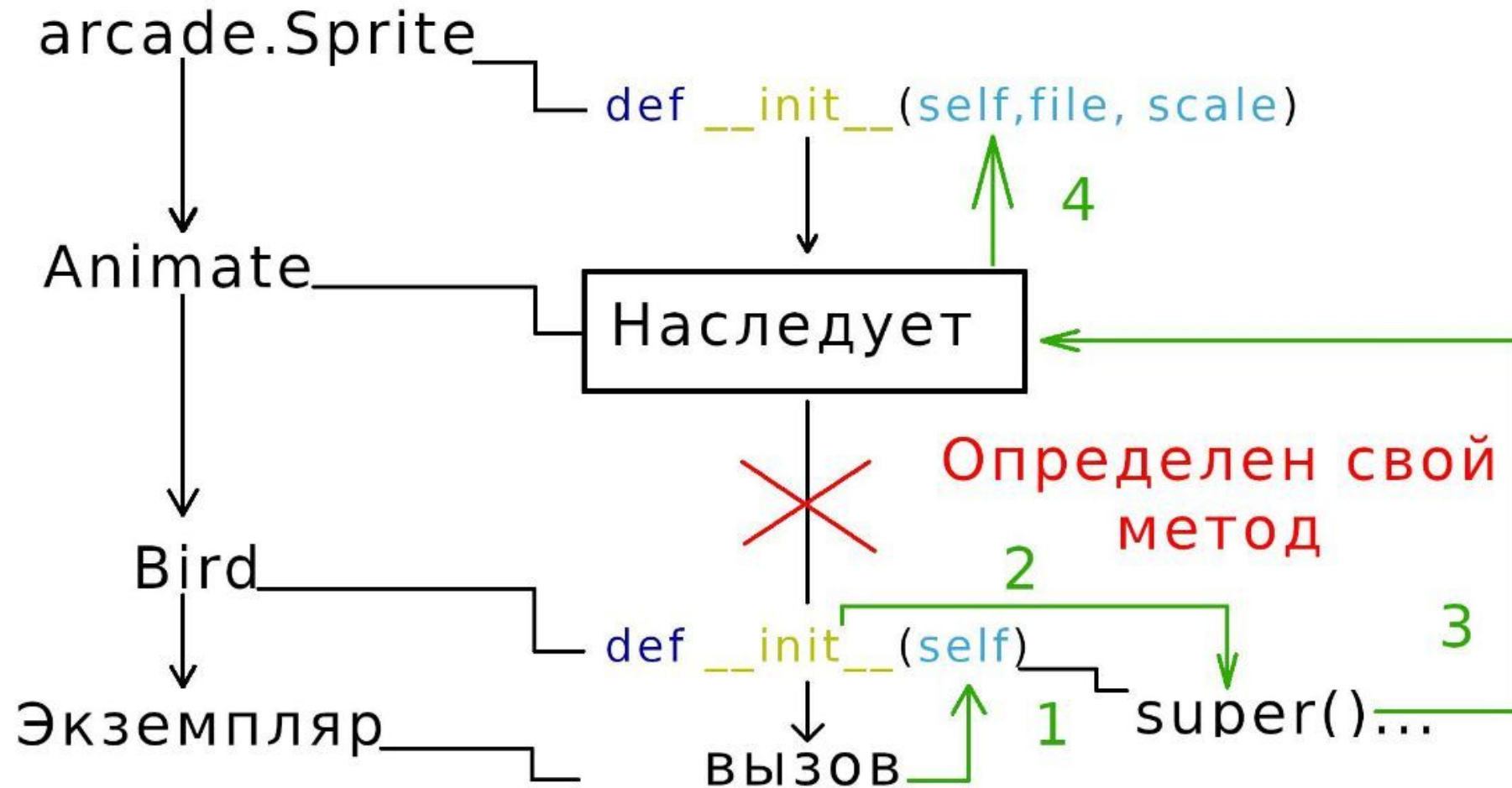
Если в дочернем классе необходимо переопределить метод, но вместе с тем сохранить поведение родительского метода, его нужно вызвать в методе класса-потомка, используя специальную функцию **super**:

```
class Bird(Animate):
```

```
    def __init__(self):
```

```
        super().__init__("bird.png", 0.1) # вызываем
```

```
        конструктор родителя (Animate)
```





Атрибут **angle** класса **Sprite** отвечает за наклон спрайта

**self.angle = 25**

Если же угол наклона меняется и его нужно ограничить, понадобится условие, почти как при ограничении координат спрайта:

**self.angle += 0.1**

**if self.angle >= 45:**

**self.angle = 45** # угол наклона не будет больше 45





**arcade.SpriteList** - класс, позволяющий работать со списком спрайтов (обновлять его и отрисовывать) как с одним спрайтом:

```
pipe_list = arcade.SpriteList()
```

```
pipe_list.append(pipe_bottom)
```

...

```
pipe_list.draw()
```



**flipped\_vertically** - параметр конструктора класса **Sprite**, отвечает за отражение картинки по вертикали:

```
super().__init__("pipe.png", 0.2, flipped_vertically=True) # картинка  
перевернется
```

**arcade.check\_for\_collision\_with\_list** - функция, проверяющая столкновение одного спрайта со списком спрайтов. Принимает спрайт и спрайтлист. Возвращает список элементов из **SpriteList**, с которыми столкнулся спрайт. Если столкновения не было, возвращает пустой список.

```
hits = arcade.check_for_collision_with_list(self.bird, self.pipe_list)
```





**arcade.load\_sound** - функция, загружающая в программу звуковой файл.

```
self.sound = arcade.load_sound("audio/wings.wav")
```

**arcade.play\_sound** - функция, проигрывающая звук. Принимает сам звук и громкость (число от 0 до 1, где 1 - 100%).

```
arcade.play_sound(self.sound, 0.3)
```