



Курс Java-разработчик



ООП



Конструктор

- Экземплярный конструктор – это экземплярный метод, предназначенный для инициализации только что созданного объекта.

Инициализация объекта главным образом заключается в заполнении информации о классе, которому принадлежит объект.

В Java операция создания объекта (`new`) объединена с вызовом конструктора.



Конструктор

В языке Java экземплярный конструктор представляет собой метод, имя которого совпадает с именем класса, а возвращаемое значение отсутствует.

```
1 package com.example;
2
3 public class Application {
4
5     public static void main(String[] args) {
6         var cat = new Cat("Мурзик", 2);
7         cat.say();
8     }
9
10 }
```

Создание объекта с вызовом конструктора

```
1 package com.example;
2
3 public class Cat {
4
5     private String name;
6     private Integer age;
7
8     public Cat(String name, Integer age) {
9         this.name = name;
10        this.age = age;
11    }
12
13    public Integer getAge() {
14        return this.age;
15    }
16
17    public String getName() {
18        return this.name;
19    }
20
21    public void setName(String name) {
22        this.name = name;
23    }
24
25    public void setAge(Integer age) {
26        this.age = age;
27    }
28
29    public void say() {
30        System.out.println("Мяу");
31    }
32
33 }
```



Наследование

- Тело любого конструктора производного класса должно начинаться с вызова одного из конструкторов базового класса (**super**).



Наследование

```
1 package com.example;
2
3 public class Cat extends Animal {
4
5     public Cat(String name, Integer age) {
6         super(name, age);
7     }
8
9     public void say() {
10         System.out.println("Мяу");
11     }
12 }
```

```
1 package com.example;
2
3 class Animal {
4     private String name;
5     private Integer age;
6
7     public Animal(String name, Integer age) {
8         this.name = name;
9         this.age = age;
10    }
11
12    public Integer getAge() {
13        return this.age;
14    }
15
16    public String getName() {
17        return this.name;
18    }
19
20    public void setName(String name) {
21        this.name = name;
22    }
23
24    public void setAge(Integer age) {
25        this.age = age;
26    }
27
28    public void say() {
29        System.out.println("Я - животное");
30    }
31 }
```

Вызов конструктора базового класса



Конструктор по умолчанию

- Конструктор по умолчанию (default constructor) – это экземплярный конструктор с пустой сигнатурой.

В Java конструктор по умолчанию автоматически создаётся компилятором, если для класса не определён ни один конструктор.

```
1 package com.example;
2
3 public class Cat extends Animal {
4
5     private String name;
6     private Integer age;
7
8     public Cat() {
9
10    }
11
12    public Cat(String name, Integer age) {
13        super(name, age);
14    }
15
16    public void say() {
17        System.out.println("Мяя");
18    }
19
20 }
```



Статические поля

Статическое поле, принадлежащее некоторому классу – это поле, разделяемое всеми объектами этого класса.

В языке Java статические поля объявляются с модификатором **static**.

```
1 package com.example;
2
3 public class Cat extends Animal {
4
5     private String name;
6     private Integer age;
7
8     public static Integer count;
9
10 }
```

```
1 package com.example;
2
3 public class Application {
4
5     public static void main(String[] args) {
6         var cat = new Cat();
7         Cat.count = 1;
8     }
9
10 }
```



Статические методы

- Статический метод, объявленный в некотором классе – это метод, не имеющий доступа к внутреннему состоянию объектов этого класса.

```
1 package com.example;
2
3 public class Application {
4
5     public static void main(String[] args) {
6         var cat = new Cat();
7         Cat.count = 1;
8     }
9
10 }
```



Константы

- Константа – фиксированное значение переменной, которое запрещено к изменению программой.

Для того, чтобы переменную сделать константой, необходимо использовать ключевое слово **final**.

Константы в программе принято писать большими буквами, разделяя каждое слово символом нижнего подчеркивания (Snake case).

```
1 package com.example;
2
3 public class Application {
4
5     private static final int MAX_VALUE = 3;
6
7     public static void main(String[] args) {
8         System.out.println(MAX_VALUE);
9     }
10 }
```



Домашнее задание (часть 1)

Добавить конструкторы для животных (default, по всем полям).

