



## Создание объекта. Порядок вызова конструкторов

Java Core (/quests/QUEST\_JAVA\_CORE)  
5 уровень (/quests/lectures/?quest=QUEST\_JAVA\_CORE&level=5), 5 лекция (/quests/lectures/questcore.level05.lecture05)

ОТКРЫТА

— Привет, Амиго! Сейчас я расскажу тебе о процессе создания объекта.

— А что там сложного, дядя Риша? Написал `new` имя класса, указал правильный конструктор и готово!

— Это так. Но что при этом происходит внутри объекта?

— А что там происходит?

— А вот что! Объект создается в несколько этапов.

- 1) Сначала выделяется память под все переменные — поля класса.
- 2) Затем идет инициализация базового класса.
- 3) Потом всем переменным присваиваются значения, если они указаны.
- 4) И наконец, вызывается конструктор.

— Выглядит не очень сложно: сначала переменные, затем конструктор.

— Давай посмотрим, как это будет работать на примере двух классов:

Код

Описание

```
1 class Pet
2 {
3     int x = 5, y = 5; ←-
4     int weight = 10; ←-
5
6     Pet(int x, int y)
7     {
8         this.x = x; ←-
9         this.y = y; ←-
10    }
11 }
12 class Cat extends Pet
13 {
14     int tailLength = 8; ←-
15     int age;
16     Cat(int x, int y, int age)
17     {
18         super(x, y); ←-
19         this.age = age; ←-
20    }
21 }
```

Мы создали два класса: Pet(животное) и Cat(кот).

В классе Cat мы видим явный вызов конструктора базового класса.

Он всегда должен быть в первой строке конструктора.

Вот что произойдет после выделения памяти:

18 – вызов конструктора базового класса

3, 4 – инициализация переменных в Pet

8, 9 – отработает код конструктора Pet

далее начнется процесс инициализации класса Cat

14 – инициализация переменных в Cat

19 – отработает код конструктора Cat

```
1 public static void main(String[] args)
2 {
3     Cat cat = new Cat (50, 50, 5);
4 }
```

— Что-то немного запутанно. Почему так сложно?

— На самом деле не сложно, если знать что на самом деле происходит:

Если у класса нет ни одного конструктора, он будет создан автоматически.

### Конструктор по умолчанию

```
1 class Cat
2 {
3     int x = 5;
4     int y = 5;
5 }
```

```
1 class Cat
2 {
3     int x = 5;
4     int y = 5;
5     public Cat()
6     {
7     }
8 }
```

Если не вызываешь конструктор базового класса, его вызов будет добавлен автоматически.

#### Вызов конструктора базового класса

```
1 class Pet
2 {
3     public String name;
4 }
```

```
1 class Pet extends Object
2 {
3     public String name;
4     public Pet()
5     {
6         super();
7     }
8 }
```

```
1 class Cat extends Pet
2 {
3     int x = 5;
4     int y = 5;
5 }
```

```
1 class Cat extends Pet
2 {
3     int x = 5;
4     int y = 5;
5     public Cat()
6     {
7         super();
8     }
9 }
```

Инициализация переменных класса происходит в конструкторе.

**Инициализация переменных класса**

```
1 class Cat
2 {
3     int x = 5;
4     int y = 5;
5 }
```

```
1 class Cat
2 {
3     int x;
4     int y;
5     public Cat()
6     {
7         super();
8         this.x = 5;
9         this.y = 5;
10    }
11 }
```

**Как все это происходит на самом деле**

```
1 class Pet
2 {
3     int x = 5, y = 5;
4     int weight = 10;
5     Pet(int x, int y)
6     {
7         this.x = x;
8         this.y = y;
9     }
10 }
11
12 class Cat extends Pet
13 {
14     int tailLength = 8;
15     int age;
16     Cat(int x, int y, int age)
17     {
18         super(x, y);
19         this.age = age;
20     }
21 }
```

```
1 class Pet extends Object
2 {
3     int x;
4     int y;
5     int weight;
6
7     Pet(int x, int y)
8     {
9         //вызов конструктора базового класса
10        super();
11        //инициализация переменных
12        this.x = 5;
13        this.y = 5;
14        this.weight = 10;
15        //вызов кода конструктора
16        this.x = x;
17        this.y = y;
18    }
19 }
20 class Cat extends Pet
21 {
22     int tailLength;
23     int age;
24     Cat(int x, int y, int age)
25     {
26         //вызов конструктора базового класса
27         super(x, y);
28         //инициализация переменных
29         this.tailLength = 8;
30         //вызов кода конструктора
31         this.age = age;
32     }
33 }
```

ЗАДАЧА **T** Java Core, 5 уровень, 5 лекция

ДОСТУПНА

★★★★☆

**Набираем код**

Иногда думать не надо, строчить надо! Как ни парадоксально звучит, порой пальцы «запоминают» лучше, чем сознание. И даже на втором квесте иногда полезно просто понабирать код. Вперёд!

×16

Открыть

— Теперь намного понятнее: сначала базовый класс, затем переменные вне конструктора, затем вызов кода конструктора.

— Молодец, Амиго, именно так!

&lt; (/quests/lectures/questcore.level05.lecture04)

×15 &gt; (/quests/lectures/questcore.level05.lecture06)

VK 0

Facebook 0

LinkedIn 0

Twitter 0

G+ 0

+26

Комментарии (19)

популярные

новые

старые

Никита

JavaRush.ru/), **G+** (<https://plus.google.com/114772402300089087607>), **Twitter** ([https://twitter.com/javarush\\_ru](https://twitter.com/javarush_ru)), **Instagram** ([https://www.instagram.com/javarush\\_ru/](https://www.instagram.com/javarush_ru/))



Программистами не рождаются  
© 2018