

# Class Object

№ урока: 9 Курс: Java Fundamentals

Средства обучения: Компьютер с установленной IntelliJ IDEA.

## Обзор, цель и назначение урока

Рассмотреть класс Object.  
Рассмотреть иерархию классов.  
Наследование класса Object.  
Методы класса Object.  
Методы equals, hashCode.

## Изучив материал данного занятия, учащийся сможет:

- Оперировать знаниями по классу Object.
- Работать с методами класса Object.
- Сравнивать по equals, hashCode;

## Содержание урока

1. Создание класса с явным наследованием от класса Object.
2. Создание класса без явного наследования от класса Object.
3. Методы класса Object.
4. equals and hashCode.

## Резюме

Класс **Object** - является корнем иерархии классов. Каждый класс является **subclass** для класса **Object**. Все объекты, включая массивы, реализовывают методы этого класса.

Класс **Object** наследуется в любом случае, явно (через **extends**) или неявно.

Методы класса Object:

**clone()** - Создает и возвращает копию объекта.

**equals()** - сравнивает объекты.

**finalize()** - объект дает сигнал сборщику мусора

**getClass()** – возвращает класс объекта

**hashCode()** – возвращает хэш-код объекта.

**toString()** – преобразовывает объект в строку и возвращает его

**notify, notifyAll, wait** – методы, которые используются при многопоточности.

**Перезапись(Override)** – метода **toString** позволит вам вернуть объект в нужном вам виде. Если в классе не переопределены **hashCode()** и **equals()**, то используется их реализация из класса **Object**.

```
public boolean equals(Object obj) {  
    return (this == obj);  
}
```

Метод **equals** класса **Object** реализует самые разные возможные отношения эквивалентности объектов; то есть, для любых ненулевых (**not null**) значений ссылок **X** и **Y**, этот метод возвращает истину, если, и только если **X** и **Y** относятся к одному объекту (**x == y** имеет значение **TRUE**, то есть ссылки на объект равны).

Правила сравнения:

1. Рефлексивность - для любого **not null** значения ссылки **X**, **x.equals(x)** должен возвращать **true**.

2. Симметричность - для любых **not null** значений ссылок X и Y, `x.equals(y)` должен возвращать **true**, если и только если `y.equals(x)` возвращает **true**.
3. Транзитивность: для любых **not null** значения ссылки X, Y, и Z, если `x.equals(y)` возвращает **true** и `y.equals(z)` возвращает **true**, `x.equals(z)` должно вернуть **true**.
4. Консистентность: для любых **not null** значений ссылок x и y, многочисленными вызовами `x.equals(y)` последовательно возвращает **true** или **false**.
5. Для любых **not null** значений X, `x.equals(null)` должно вернуть **false**.

Контракт `hashCode()`:

Когда вызывается один и тот же объект несколько раз в рамках выполнения приложения Java, то метод должен последовательно возвращать одно и тоже число.

Если два объекта равны в соответствии метода `equals(Object)`, а затем вызвать метод `hashCode` для каждого из объектов, то должны вернуться одинаковые целочисленные результаты. Этот метод реализован таким образом, что для одного и того-же входного объекта, хеш-код всегда будет одинаковым. Следует понимать, что множество возможных хеш-кодов ограничено примитивным типом `int`. Отсюда следует утверждение: "Множество объектов мощнее множества хеш-кодов". Из-за этого ограничения, вполне возможна ситуация, что хеш-коды разных объектов могут совпасть. Из этого следует:

- Если хеш-коды разные, то и входные объекты гарантированно разные.
- Если хеш-коды равны, то входные объекты не всегда равны.

Ситуация, когда у **разных объектов одинаковые хеш-коды** называется — **коллизией**. Вероятность возникновения коллизии зависит от используемого алгоритма генерации хеш-кода.

Связь `equals()` и `hashCode()`.

Эквивалентность и хеш-код тесно связаны между собой, поскольку хеш-код вычисляется на основании содержимого объекта (значения полей) и если у двух объектов одного и того же класса содержимое одинаковое, то и хеш-коды должны быть одинаковые

```
object1.equals(object2)// true
object1.hashCode() == object2.hashCode()//true
```

Алгоритм вычисления `hashCode`:

- 1) Объявляется **ненулевое** значение, любое, и записывается в переменную **result**.
- 2) Для каждого поля в вашем объекте сделать следующее(допустим поле "f"):
  - если поле `boolean`, выполнить `(f ? 1 : 0)`
  - если поле `byte`, `char`, `short`, `int`, выполнить `(int) f`;
  - если поле `long`, выполнить `(int) (f ^ (f >>> 32))`;
  - если поле `float`, выполнить `Float.floatToIntBits(f)`.
  - если поле `double`, выполнить `Double.doubleToLongBits(f)`.
- 3) Если поле является ссылкой на объект, то вызывайте метод `hashCode()` этого объекта.
- 4) Объединить полученные с п.2 – п.3 значения следующим образом: `37 * result + f`
- 5) Если поле является массивом, то п.4 для каждого элемента массива.
- 6) Проверить, что равные объекты возвращают одинаковый `hashCode`

### Закрепление материала

- Что такое класс **Object**?
- Какие методы есть в классе **Object**?
- Правила сравнения `equals`?
- Суть метода `hashCode`?
- Как нужно писать в `hashCode` для типа `Boolean`?

## Дополнительное задание

### Задание

Используя IntelliJ IDEA создать проект, пакет.

Создать класс Animal с именем String, возрастом int, хвостом Boolean. В классе переопределить метод toString, что бы вывод был следующим

«Имя: Васька, возраст: 45, хвост: нет».

В классе Animal переопределить методы equals & hashCode.

## Самостоятельная деятельность учащегося

### Задание 1

В любой из профильных книг (**Хорстман, Эккель**) найти соответствующие темы и закрепить материал.

Использование **YouTube, Quizful** приветствуется.

### Задание 2

Создать классы:

1) Основной класс Device (manufacturer(String), price(float), serialNumber(String));

2) Сабкласс Monitor (resolutionX(int), resolutionY(int)) and EthernetAdapter (speed (int), mac (String));

Добавить методы доступа. Конструктор.

### Задание 3

Смотреть задание 2.

В обоих классах переопределить метод toString, что бы вывод был следующим:

Device: manufacturer =Samsung, price=120, serialNumber=AB1234567CD

Monitor: manufacturer =Samsung, price=120, serialNumber=AB1234567CD, X=1280,Y=1024

### Задание 4

Смотреть задание 2.

Переопределить методы equals & hashCode в каждом классе.

Создать класс Main, в котором создать объекты классов и продемонстрировать переопределенные методы.

## Рекомендуемые ресурсы

<http://docs.oracle.com/javase/7/docs/api/java/lang/Object.html>

<https://docs.oracle.com/javase/tutorial/java/landl/objectclass.html>

<http://habrahabr.ru/post/168195/>

<http://www.seostella.com/ru/article/2012/10/04/kak-sgenerirovat-hashcode-v-java.html>