

# Прекрасный мир Java

Часть 6.

# Методы класса Object

**public boolean equals(Object obj)**

Метод equals() служит для проверки, является ли содержимое двух разных объектов одним и тем же.

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

# Методы класса Object

**public boolean equals(Object obj)**

```
String str1 = new String("Test");
```

```
String str2 = new String("Test");
```

```
boolean test1 = (str1 == str2);
```

```
boolean test2 = (str1.equals(str2));
```

```
boolean test3 = (str2.equals(str1));
```

# Методы класса Object

**public boolean equals(Object obj)**

- Операция equals() должна быть симметричной
- Результат вызова obj.equals(null) всегда должен быть false

# Методы класса Object

**public int hashCode()**

Метод hashCode () должен просто вернуть некоторое целое число, согласованное с состоянием объекта.

# Методы класса Object

**public int hashCode()**

Если `a.equals(b)` возвращает `true`, то `a.hashCode()` и `b.hashCode()` обязаны совпадать.

Из равенства хэш-кодов не следует равенство объектов.

Но из неравенства хэш-кодов следует неравенство объектов.

# Основные понятия ООП

- Инкапсуляция
- **Абстракция**
  - **Интерфейс**
- Наследование
- Полиморфизм

# Интерфейс



Интерфейс

Класс

Интерфейс

Абстрактный класс

Класс

Наличие виртуальных методов  
позволяет использовать абстрактные  
классы.

```
abstract class Parent {  
    public printMsg() {  
        System.out.println(getMessage());  
    }  
    abstract protected String getMessage();  
}
```

```
class Child extends Parent {  
    protected String getMessage() {  
        return "It is a child class!";  
    }  
}
```

Есть по крайней мере три роли, которые может выполнять человек, находящийся в университете: студент, преподаватель, сотрудник.



Применение делегирования позволяет динамически изменять поведение объекта, при использовании наследования поведение объекта наследуется жестко.

Наследование – частный (неявный) случай делегирования, когда в роли делегата выступает класс предок.

- Абстракции – основа для построения полиморфного кода
- Интерфейсы – способ описания абстракций
- Наследование – механизм повторного использования кода и способ создания устойчивых поведенческих и структурных отношений между классами

- Полиморфизм – механизм работы с объектами через абстракции, позволяющий создавать модульный расширяемый код
- Абстрактные классы – способ частичной фиксации модели поведения
- Делегирование – способ реализации изменчивой модели поведения