



Перевизначення методів equals та hashCode



Методи `equals` та `hashCode`

- Метод ***equals()*** – перевірка рівності елементів. Типове застосування – пошук в масивах, колекціях, картах.
- Метод ***hashCode()*** – повертає хеш-функцію об'єкта. Якщо цей метод перевизначено, елементи можна використовувати в **HashSet**, а також як ключі **HashMap**. **Якщо для цих типів перевизначається `equals()`, то повинен бути перевизначений і `hashCode()` .**



Що можна сказати про код

```
class MyClass {  
    int a;  
    @Override  
    public boolean equals(MyClass other) {  
        return this.a == other.a;  
    }  
}
```



Загальні вимоги до equals

- **Рефлексивність.** Для будь-якого посилання `ref` `ref.equals(ref)` повинно завжди давати `true`.
- **Симетричність.** `x.equals(y)` дорівнює `true` тоді і тільки тоді, коли `y.equals(x)`.
- **Транзитивність.** Якщо `x.equals(y)`, `y.equals(z)`, то `x.equals(z)`.
- **Несуперечливість.** Різні виклики `equals` для тих же об'єктів повинні давати той самий результат, якщо самі об'єкти не змінилися.
- **Порівняння з `null`.** Для кожного не-`null` посилання `ref.equals(null)` завжди `false`.

Часто перевіряється ще й відповідність типів.



Загальні вимоги до hashCode

- **Несуперечливість під час виконання** – різні виклики того самого об'єкта повинні давати той самий результат, якщо сам об'єкт не змінився.
- **Для рівних об'єктів значення хеш-функції повинні співпадати.**
- Для нерівних об'єктів неспівпадіння не обов'язкове, але дуже рекомендується.



Можлива реалізація equals

```
public boolean equals(Object obj) {  
    if (obj == null) { return false; }  
    if (getClass() != obj.getClass()) { return  
false; }  
    final KI other = (KI) obj;  
    if (this.a != other.a) { return false; }  
    if (this.b != other.b) { return false; }  
    if (this.c != other.c) { return false; }  
    return true;  
}
```



Можлива реалізація equals

```
public boolean equals(Object obj) {  
    if (obj == null) { return false; }  
    if (getClass() != obj.getClass()) { return  
false; }  
    final KI other = (KI) obj;  
    if (this.a != other.a) { return false; }  
    if (this.b != other.b) { return false; }  
    if (this.c != other.c) { return false; }  
    return true;  
}
```

Можлива реалізація

hashCode

```
public int hashCode() {  
    int hash = 3;  
    hash = 31 * hash + this.a;  
    hash = 31 * hash + this.b;  
    hash = 31 * hash + this.c;  
    return hash;  
}
```




A6o:

```
public int hashCode() {  
    return this.toString().hashCode();  
}
```



Методи `equals` та `compareTo`

- Природно, ці методи повинні бути узгоджені між собою.
- Якщо *`compareTo`* визначений належним чином, то *`equals`* може бути визначений, наприклад, так:
`return compareTo(other)==0;`