

# equals() メソッド

Seiichi Nukayama

2022 年 6 月 8 日

## 目次

1	equals() メソッドのオーバーライド	1
1.1	準備	1
1.2	equals() メソッドのオーバーロード	1
1.3	equals() メソッドのオーバーライド	2
2	hashCode() メソッドのオーバーライド	2
2.1	hashCode() をオーバーライドしない場合	2
2.2	hashCode() メソッドのオーバーライド	3

## 1 equals() メソッドのオーバーライド

### 1.1 準備

以下のような User クラスを考える。

リスト 1 User.java

```
1 public class User {
2     public String name;
3
4     public User(String name) {
5         this.name = name;
6     }
7 }
```

### 1.2 equals() メソッドのオーバーロード

これに、以下のような equals() メソッドをつくる。これは絶対にやってはいけないと言われているが、Object クラスの equals() メソッドをオーバーライドするのではなくて、オーバーロードしてみる。

リスト 2 User.java

```
1 ...( つづき )...
2
3 public boolean equals(User u) {
4     if (this.name.equals(u.name)) {
5         return true;
6     }
7     return false;
8 }
```

テストのために、以下のようなコードを試してみると.....。

リスト 3 Main.java

```
1 public class Main {
2     public static void main(String[] args) {
3         User u1 = new User('Sasuke');
4         User u2 = new User('Sasuke');
5         System.out.println(u1.equals(u2));
6     }
7 }
```

これは、true と表示される。では、これは？

リスト 4 Main.java

```
1 public class Main {
2     public static void main(String[] args) {
3         List<User> userList = new ArrayList<>();
4         User u1 = new User('Sasuke');
5         userList.add(u1);
6         u1 = new User('Sasuke');
```

```

7   userList.remove(u1);
8   for (User u : userList) {
9       System.out.println(u.name);
10  }
11  System.out.println(userList.size());
12  }
13 }

```

これは `Sasuke` / `1` と表示される。この場合、`userList` に `add()` した `h1` と、`userList` から `remove()` した `h1` を同じものとして見なしていないことがわかる。

`equals()` メソッドのオーバーロードはバグの元となるので、絶対やらないようにとされている。

### 1.3 `equals()` メソッドのオーバーライド

`User.java` 中の `equals()` メソッドを以下のように `Object` クラスの `equals()` メソッドをオーバーライドしたものにすると、結果はちがってくる。

リスト 5 `User.java`

```

1  ... ( つづき ) ...
2
3  @Override
4  public boolean equals(Object o) {
5      if (this == o) {
6          return true;
7      }
8      if (!(o instanceof User)) {
9          return false;
10     }
11     User u = (User) o;
12     if (this.name.equals(u.name)) {
13         return true;
14     }
15     return false;
16 }

```

このコードに変えてから `Main.java` を実行すると

`0` となる。

## 2 `hashCode()` メソッドのオーバーライド

`equals()` メソッドをオーバーライドしたら、`hashCode()` メソッドもオーバーライドしなければならない。

### 2.1 `hashCode()` をオーバーライドしない場合

先ほどの `equals()` メソッドをオーバーライドした `User` クラスを試してみる。

以下のようなコードを試す。

リスト 6 `Main.java`

```

1 import java.util.HashSet;
2 import java.util.Set;
3
4 public class Main {
5
6     public static void main(String[] args) {
7         Set<User> userSet = new HashSet<>();
8         User u = new User("Sasuke");
9         userSet.add(u);
10        showList(userSet);
11        u = new User("Sasuke");
12        userSet.remove(u);
13        showList(userSet);
14    }
15
16    public static void showList(Set<User> list) {
17        for (User u : list) {
18            System.out.println(u.name);
19        }
20        System.out.println("要素数:" + list.size());
21    }
22 }
23 }

```

このコードを実行すると、以下のように出力される。

```

Sasuke
要素数:1
Sasuke
要素数:1

```

途中で remove() されているにもかかわらず、remove() されていない。

## 2.2 hashCode() メソッドのオーバーライド

そこで、以下のように hashCode() メソッドをオーバーライドする。

リスト7 User.java

```

1 ... ( つづき ) ...
2
3 @Override
4 public int hashCode() {
5     int result = 37;
6     result = result * 31 + name.hashCode();
7     return result;
8 }
9 }

```

こののち、先ほどの Main クラスを実行すると、以下のように出力される。

```

Sasuke
要素数:1
要素数:0

```