

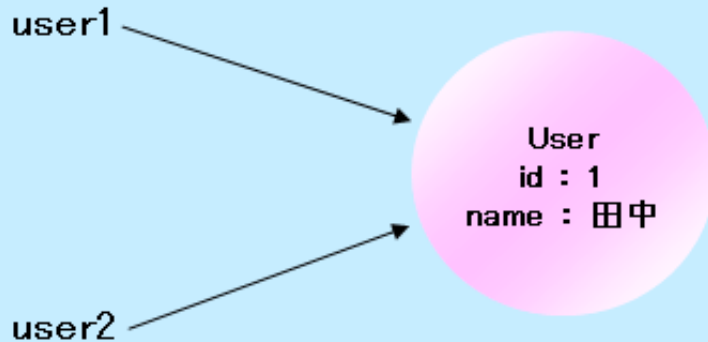
[トップ](#)

[Java tipsトップ](#)

オブジェクトの"同一性"とは？

1. [前提条件](#)
2. 同一性と同値性
3. [equalsメソッドの実装](#)
4. [hashCodeとは](#)
5. [equalsメソッドとhashCodeメソッドの正体](#)

user1とuser2は同じ(同一の)オブジェクトを参照している。



```
User user1 = new User(1, "田中");  
User user2 = user1;
```

user1とuser2は同じオブジェクトの参照を保持しています。従って、user1とuser2が参照しているオブジェクトは"同一"です。

オブジェクトの"同値性(等価性)"とは？

```
User user3 = new User(1, "田中");  
User user4 = new User(2, "鈴木");  
User user5 = new User(1, "鈴木");
```

user3とuser5はIDが等しい。user4とuser5は名前が等しい。

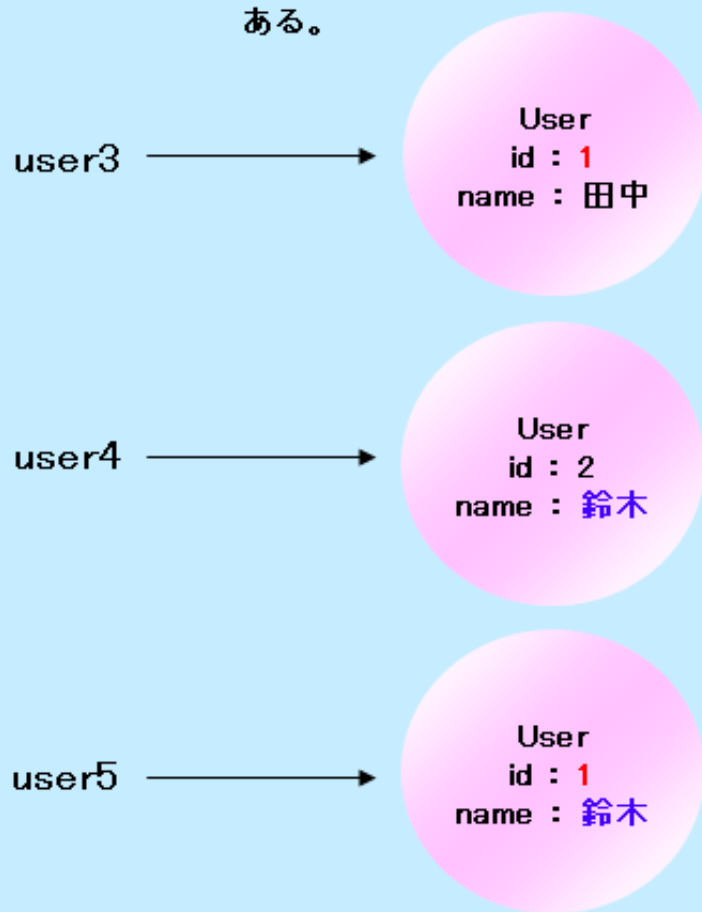
従って、user3とuser5はIDの値において"同値"のオブジェクトです。user4とuser5は名前の値において"同値"のオブジェクトです。

オブジェクトの同値性はプログラマ自身がビジネスのルールに従い実装する必要があります。

オブジェクトの同値性は、該当するクラスのequalsメソッドに定義します。

もし、UserクラスのオブジェクトをIDの値において同値性を判定するルールがあるならば、equalsメソ

user3、user4、user5はそれぞれ異なる(同一ではない)オブジェクトを参照している。
ただし、内部状態を比較した場合、user3とuser5はIDの値において同値であり、user4とuser1はnameの値において同値である。



ッドに、IDの値を比較する処理を記述します。

もし、Userクラスのオブジェクトを名前の値において同値性を判定するルールがあるならば、equalsメソッドに、名前の値を比較する処理を記述します。

オブジェクトの同一性の判定方法

オブジェクトの同一性は比較演算子"=="で判定します。上記user1とuser2は同一のオブジェクトなので

```
user1 == user2 // => true
```

の結果は"true"となります。

user3、user4、user5は全て異なるオブジェクトなので

```
user3 == user4 // => false
user3 == user5 // => false
user4 == user5 // => false
```

の結果は、全て"false"となります。

オブジェクトの同値性の判定方法

オブジェクトの同値性はそれぞれのクラスに実装されている"equals"メソッドで比較します。

UserクラスのequalsメソッドがIDを比較条件として実装されている場合、 equalsメソッドの結果は以下の通りとなります。

```
user3.equals(user4) // ⇒ false  
user3.equals(user5) // ⇒ true  
user4.equals(user5) // ⇒ false
```

Userクラスのequalsメソッドが名前を比較条件として実装されている場合、 equalsメソッドの結果は以下の通りとなります。

```
user3.equals(user4) // ⇒ false  
user3.equals(user5) // ⇒ false  
user4.equals(user5) // ⇒ true
```

