What is the default implementation of `hashCode`? [duplicate]

Ask Question

This question already has an answer here:
What is an object's hash code if hashCode() is not overridden? *11 answers*

If one does not override the `hashCode` method, what is the default implementation of `hashCode` ?

java hashcode

asked Feb 28 '13 at 8:27
John Threepwood
4,883 17 62 115

marked as duplicate by Ingo, Jan Zyka, John Threepwood, MrSmith42, zessx Feb 28 '13 at 10:27

This question has been asked before and already has an answer. If those answers do not fully address your question, please ask a new question.

About default implementation you can read this – CAMOBAP Feb 28 '13 at 8:39

1 default implementation is JVM specific, but in general it returns `return Objects.hash(this.field1, this.field2, this.field3, etc.);` – NoName Jun 24 '17 at 6:19

3 Answers

Then this class inherits `hashCode` from one of its ancestors. If non of them overrides it, then Object.hashCode is used.

From the docs:

> As much as is reasonably practical, the hashCode method defined by class Object does return distinct integers for distinct objects. (This is typically implemented by converting the internal address of the object into an integer, but this implementation technique is not required by the JavaTM programming language.)

So default implementation is JVM-specific

edited Apr 2 '17 at 2:19
Ry- ♦
160k 33 323 344

answered Feb 28 '13 at 8:28
default locale
8,810 11 38 51

Thank you, yes. What does the implementation of `Object.hashCode() look` like? – John Threepwood Feb 28 '13 at 8:29

1 @JohnThreepwood it's an implementation detail, you need to address your JVM docs for that. – default locale Feb 28 '13 at 8:32

2 stackoverflow.com/a/32454673/6785908 – so-random-dude Mar 11 '17 at 23:27

By default, methods that are not overriden are inherited from `Object` .

If you look at that method's documentation, the return values are " `[...] distinct integers for distinct objects. (This is typically implemented by converting the internal address of the object into an integer [...])` ". The method in `java.lang.Object` is declared as native, which means the implementation is provided by the JVM and may vary depending on your runtime environment.

A small example:

```
Object o1 = new Object();
Object o2 = new Object();
System.out.println(o1.hashCode());
System.out.println(o2.hashCode());
```

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS

+ Create Team

prints (using my jdk6):

```
1660187542
516992923
```

A Hex representation of the `hashCode()` value is used in the default implementation of `toString()` by the way: Running `System.out.println(o1)` prints something like

```
java.lang.Object@7a5e1077
```

answered Feb 28 '13 at 8:35
f1sh
6,958 3 17 41

Object.hashcode() is a native method.

```
public native int hashCode();
```

That means it's implemented in platform specific code and is exposed as a native method.

code for the same will be a compiled code and not available withing JDK

this existing question might provide more info.

edited May 23 '17 at 12:10
Community ♦
1 1

answered Feb 28 '13 at 8:34
TheWhiteRabbit
10.5k 3 26 56