Cheat single inheritance in Java?

Asked 16 years, 3 months ago Modified 7 years, 2 months ago

Viewed 5k times

8

I have heard there is a way to cheat single inheritance and implement multiple inheritance in Java. Does anyone know how to implement this(with out using interface)?

Just out of curiosity ;-)

java oop inheritance

Share

Improve this question

Follow

edited May 13, 2012 at 2:48

community wiki
11 revs, 2 users 100%
Omnipotent

Its a very different thought. – Warrior Dec 4, 2008 at 8:34

13 Answers

Sorted by: Highest score (default)

Sure you can, but it's tricky and you should really consider if that's the way you want to go.

20

The idea is to use scope-based inheritance coupled with type-based one. Which is type-talk for saying that for internal purposes, inner classes "inherit" methods and fields of the outer class. It's a bit like mixins, where the outer class is mixed-in to the inner class, but not as safe, as you can change the state of the outer class as well as use its methods.

Gilad Bracha (one of the main java language designers) wrote a paper discussing that. So, suppose you want to share some methods for internal use between some unrelated classes (e.g, for string manipulation), you can create sub classes of them as inner classes of a class that has all the needed methods, and the sub classes could use methods both from their super classes and from the outer class.

Anyway, it's tricky for complex classes, and you could get most of the functionality using static imports (from java 5 on). Great question for job interviews and pub quizzes, though ;-)

Share Improve this answer

Follow

edited Dec 6, 2015 at 11:27

Tom Zych

13.6k ● 9 ● 37 ● 54

answered Sep 16, 2008 at 10:37

Michael Bar-Sinai

2,739 ● 21 ● 28

1 there's a minor typo here , I'd fix it but I don't have enough rep yet. 'create sub classes of THEN as' – TygerKrash Sep 21, 2010 at 9:15

@TygerKrash You don't need rep to edit posts: SE is meant to be like a Wiki. – Yatharth Agarwal Jan 9, 2015 at 14:20

5

~~Single~~Multiple inheritance is not supported by Java, instead it has got interfaces to serve the same purpose. In case you are adamant on using multiple inheritance it should be done in C++.

Share Improve this answer

Follow

edited May 12, 2009 at 20:47

community wiki
2 revs, 2 users 67%
Warrior

4

Use of composition instead of inheritance tends to be the way around this. This actually also helps a lot with testability, so it's good practice in general.

If you just want your type to "behave" like several other types, you can inherit from as many interfaces as you like, though; you can't "borrow" implementation details from these though, obviously.

Share Improve this answer

Follow

answered Sep 16, 2008 at 9:04

Calum
5,926 ● 2 ● 29 ● 23

3

I believe that the fundamental reason that Java doesn't support multiple inheritance is the same as C#; all objects are ultimately derived from Object and it's having multiple paths to the same base class is ambiguous for the compiler. Ambiguous == Bad, so the compiler doesn't allow it.

Instead, you can simulate multiple inheritance through delegation. See this article for an example.

Share Improve this answer

Follow

answered Sep 16, 2008 at 9:05

Steve Morgan

13.1k ● 2 ● 43 ● 49

> Actually sharing Object is not ambiguous for the compiler. It only gets ambiguous when methods are defined in different classes. So if one branch overrides hashcode, and the other doesn't. In those cases the programmer will have to make a choice. – Paul de Vrieze Sep 16, 2008 at 10:38

2

You can cheat it a little (and I stress a little) by using java.lang.reflect.Proxy instances.

This really just allows you to add extra interfaces and delegate their calls to another instance at runtime.

As someone who mentors and tutors new developers I would be horrified if somebody showed me code that did this. Reflection is one of those tools that you really need to understand and have a good understanding of Java before jumping in. I personally have only ever done this

once, and it was to make some code I didn't have control over implement some interfaces some other code I had no control over was expecting (it was a quick hack so I didn't have to write and maintain too much glue code).

Share Improve this answer

Follow

answered Sep 16, 2008 at 9:12

Aidos

Yeah, this is a "neat trick" but it's not something that anyone should really use in production code! – Calum Sep 16, 2008 at 9:17

but to use Proxy you need interfaces. OP: "...with out using interface..." – user85421 May 13, 2009 at 9:02

1

Use `interface`s. You can implement as many as you'd like. You can usually use some variant on the Composite Pattern (GoF) to be able to reuse implementation code if that's desirable.

Share Improve this answer

Follow

answered Sep 16, 2008 at 9:05

Hank Gay

71.8k • 36 • 161 • 222

1

You need to be careful to distinguish interface inheritance (essentially inheritance of a contract to provide particular facilities) from implementation inheritance (inheritance of implementation mechanisms).

Java provides interface inheritance by the *implements* mechanism and you *can* have multiple interface inheritance.

Implementation inheritance is the *extends* mechanism and you've only got a single version of that. Do you *really* need multiple implementation inheritance? I bet you don't, it's chock full of unpleasant consequences, unless you're an Eiffel programmer anyway.

Share Improve this answer Follow

answered Sep 16, 2008 at 9:13

Tim Hoverd

1

You could probably "simulate" it by managing the set of superclasses explicitly and using reflection to search all the superclasses for the target method. I wouldn't want to do this in production, but it might an interesting toy program. You could probably do a lot of weird stuff by leveraging reflection, creating classes on the fly, and invoking the compiler programatically.

Share Improve this answer

Follow

answered Sep 16, 2008 at 9:14

Bert F

87.4k • 12 • 113 • 126

0

JAVA doesn't support multiple Inheritence.

You can get it to implement multiple interfaces and some see this as a way round the problem. Personally I have yet to use multiple inheritence, so I can't really understand its appeal.

Normally when someone suggests multiple inheritence within c# or JAVA its due to the fact that 'they could' in c++. Im a fan of 'just because you can doens't mean you should'. As c# & JAVA doesn't support it, why try and force it to do something it wasn't designed to do. This is not to say that there are unique cases where it is a valid technique to empoly, just the code can usually be refactored to not need it.

Share Improve this answer

Follow

edited Sep 16, 2008 at 9:14

answered Sep 16, 2008 at 9:02

TK.

47.8k ● 47 ● 121 ● 148

You probably mean it doesn't support *multiple* inheritance. – Jon Limjap Sep 16, 2008 at 9:03

Trait inheritance (essentially mixin inheritance) in Scala shows a few ways in which multiple inheritance can be useful, although Java's choice not to implement it is understandable, as many implementations end up being needlessly complex to reason about. – Calum Sep 16, 2008 at 9:06

0

I was thinking about this a little more and realised that while dynamic proxies will work (it's how RMI (used?) to work), if you really want this sort of functionality you would be better off looking at aspect oriented programming (AOP) using something like AspectJ (eclipse.org/aspectj).

This way you can get several different aspects into a class, giving you pseudo mixin inheritance, without the hideously fragile inheritance heirarchies.

As everyone else has pointed out, wanting/needing multiple inheritance generally indicates you aren't approaching the problem from the right perspective. Remember the GoF principle of "prefer composition over inheritance" for a start!

Share Improve this answer

Follow

answered Sep 16, 2008 at 9:38

Aidos

0

There was an effort to bring mixins into Java. Check this link out: http://www.disi.unige.it/person/LagorioG/jam/

Share Improve this answer

Follow

answered Sep 16, 2008 at 10:21

Elnur

0

By using Inner Classes, this is what C++ sometimes prefers as well: Inner Class Idiom.

Share Improve this answer

Follow

answered May 12, 2009 at 20:29

community wiki

Özgür

-1

Yes you can say that it's a trick and it is very Interesting you cannot inherit multiple classes to a single class but it is possible to implement multiple Interfaces to a class like

```
public class parents implements first, second{

}
```

but remember, you have to override methods declared in interfaces.

Share Improve this answer Follow

edited Oct 15, 2017 at 10:16

community wiki
2 revs, 2 users 67%
Abdur Rehman Khalid