Why const keyword is not used in Java? [duplicate]

Asked 9 years, 10 months ago Active 4 months ago Viewed 43k times

33

8

This question already has answers here:

Closed 9 years ago.

Possible Duplicate:
Why is there no Constant keyword in Java?

Why const keyword is not used in Java?
Can you see any disadvantages of using some transitive const or immutable keyword in Java syntax or why common cumbersome approach was chosen?

Can you see **reasons** for closing the request, does Sun provides any explanations?

java

Share Improve this question Follow

edited May 23 '17 at 11:54
Community ♦
1 1

asked Sep 15 '11 at 9:03
Mike
17.2k 23 85 114

4 What do you want `const` to do? Obviously something else than `final` (otherwise it would be redundant). But surely not making the referenced object automagically immutable? – Thilo Sep 15 '11 at 9:05

2 same thread at stackoverflow.com/questions/2735736/... – Umesh Awasthi Sep 15 '11 at 9:06

Check this out. stackoverflow.com/questions/2735736/... – Santosh Sep 15 '11 at 9:08

That it! Making the referenced object automagically immutable! As D programming language is already implemented. It is clear for me why goto is 'deprecated' (reserved and not used) but why const is not part of Java specification till now?.. Are there any reasons for this? – Mike Sep 15 '11 at 9:10

1 Wikipedia says the JCP found it "impossible to implement" (in a backwards-compatible way) – Thilo Sep 15 '11 at 9:15

4 Answers

Active Oldest Votes

27

> Can you see reasons for closing the request, does Sun provides any explanations?

"There are no current plans to add this feature to Java. In addition to creeping featurism, we see the following problems with this feature:

1. Adding const is too late now. Had this been added from 1.0, the situation could have been different.
2. Const pollution: the C++ approach requires all const methods to be marked with a const keyword. This means that most methods will have to be marked const explicitly. This tend to clutter all methods in C++.
3. Compatibility is a very important feature of the JDK. Arguably, the collection classes should be modified to indicate that the elements are const. That would require all existing implementations to be updated in the same way, effectively breaking all existing non-JDK implementations of the collection interfaces. Similarly, hashCode would have to be const, breaking the current implementation of String."

UPDATE

Out of curiousity, I spent a few minutes trawling through the subject lines of the Project COIN mailing list. Somewhat to my surprise, nobody bothered to suggest `const`. (Or if they did, I missed it.)

So either nobody cares (enough) for this idea any more, or people with sufficient expertise to formulate a project COIN proposal recognize that there's no chance that it would pass muster.

Share Improve this answer Follow edited Feb 16 at 4:07 answered Sep 15 '11 at 9:46

Stephen C
637k 86 735 1104

+1. I guess the third reason is the biggest one. I don't completely understand the second one and if it is relevant, but the first one is (without further explanation) a bit of hand-waving (why is it too late for features if they in and of themselves have no problems, and why does that not apply to generics and closures, who also bring rather big changes). – Thilo Sep 15 '11 at 9:55

@Thilo - I agree that the reasons 1 and 2 are not great. But they ARE reasons provided by Sun. Which is what the OP asked for. – Stephen C Sep 15 '11 at 10:02

can't grasp this idea (really tried to). Can't see any broken backward compatibility or pollution of code in providing ADDITIONAL OPTIONAL way of forcing compiler (and help developer) to check that some variable marked immutable not to be used as 'left hand' value. – Mike Sep 15 '11 at 11:19

2 @Mykhaylo Adamovych - I just quoted what Sun said. I don't pretend to understand all of the issues involved, but I'm SURE it is not as simple as you think ... – Stephen C Sep 16 '11 at 5:19

It is not used because it has no function in Java. I quote from Wikipedia: