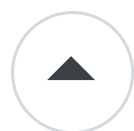


What reason is there for C# or Java having lambdas?

Asked 16 years, 2 months ago Modified 8 years, 6 months ago

Viewed 6k times



18



What reason is there for C# or java having lambdas?

Neither language is based around them, it appears to be another coding method to do the same thing that C# already did.

I'm not being confrontational, if there is a reason I would like to know the reason why. For the purpose of full disclosure I am a Java programmer with a C++ background with no lisp experience, I may just be missing the point.

c#

java

lambda

Share

Improve this question

Follow

edited Oct 16, 2008 at 18:58



tvanfosson

532k ● 102 ● 699 ● 798

asked Oct 16, 2008 at 18:56



WolfmanDragon

7,952 ● 14 ● 51 ● 61

8 Answers

Sorted by:

Highest score (default)



39



There are common use-cases which require passing (or storing) a block of code to be executed later. The most common would be event listeners. Believe it or not, the following bit of code uses a lambda-ish construct in Java:

```
JButton button = new JButton("Push me!");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("Pressed!");
    }
});
```

The anonymous inner-class is acting as a lambda, albeit an extremely verbose one. With a little bit of implicit conversion magic, we can write the following equivalent in Scala:

```
val button = new JButton("Push me!")
button.addActionListener { e =>
    println("Pressed!")
}
```

C# makes this sort of thing fairly easy with delegates and (even better) lambdas.

Share Improve this answer

edited Oct 16, 2008 at 19:23

Follow

answered Oct 16, 2008 at 19:02



Daniel Spiewak

55.1k ● 14 ● 111 ● 120

1 I can see the reason why by your example, I can also see how it could cause a lot of confusion trying to debug until one learned lambdas extremely well. – [WolfmanDragon](#) Oct 16, 2008 at 19:22

5 Actually, in my experience, lambda-rich code is *easier* to debug than the equivalent code written without lambdas. They're really pretty intuitive once you get your mind around them, and used judiciously, they can cut down on bulk (which obviously helps any code-related task, including debugging). – [Daniel Spiewak](#) Oct 16, 2008 at 19:25

The subtle difference between delegate lambdas (more comparable to anonymous methods) and Expression lambdas is also a major consideration, but good post. – [Marc Gravell](#) Oct 17, 2008 at 8:26



9



I see lambdas in C# as a very convenient short-cut for doing delegates. Much more readable to have the code right there where it is being used rather than having to search elsewhere for the delegate definition.

Share Improve this answer

answered Oct 16, 2008 at 18:57

Follow



tvanfosson

532k ● 102 ● 699 ● 798

Anonymous functions and lambdas are different in C# - blogs.msdn.com/ericlippert/archive/2007/01/10/... – [johnstok](#) Oct 16, 2008 at 19:07

2 @johnstok - A little late to be pointing this out, but that article says that they're only different from the perspective of the implementer of a C# compiler (which the author of that blog is). – [Daniel Earwicker](#) May 26, 2010 at 11:09



Syntactic Sugar.

4

It provides a convenient and more-readable way to represent an idea, in this case a tiny throw-away method.



Under the hood, the compiler expands that out to a delegate and method call, but it's the thing doing the work, not you.



Share Improve this answer

answered Oct 16, 2008 at 19:03

Follow



Bob King

25.8k ● 7 ● 57 ● 66



Lambdas allow you write less verbose, more expressive code. For example, [list comprehensions](#)...

2



BTW, work is under way to explore the possibility of adding closures to Java - in the meantime it is necessary to use anonymous classes instead (ugly).



Share Improve this answer

edited Oct 16, 2008 at 19:08

Follow

answered Oct 16, 2008 at 19:02



johnstok

98.1k ● 12 ● 56 ● 76



1



C# isn't going for purity to a particular school of language design (unlike Java, which was designed by the Smalltalkers as to be something of a pure OO language). C# is going for all-things-to-all-people, and it's pretty good at it. C# is based around gathering the best of the various styles of programming into one high-quality, well-supported language. That includes procedural, object-oriented, functional, dynamic, logic, etc. styles of programming. Obviously, so far it doesn't have much in the way of dynamic or logic styles of programming, but that is soon to come (dynamic programming coming with C# 4.0).

Share Improve this answer

answered Oct 16, 2008 at 19:11

Follow



yfeldblum

65.4k ● 12 ● 131 ● 169



1

In case of C#, lambdas are used internally to implement LINQ. See the article [The Evolution Of LINQ And Its Impact On The Design Of C#](#)



Share Improve this answer

answered Oct 16, 2008 at 19:29

Follow



Nemanja Trifunovic

24.5k ● 4 ● 52 ● 88



updated link

web.archive.org/web/20080828034246/http://msdn.microsoft.com/... – anon Jul 4, 2018 at 16:54



0

Lambda's allow for more readable code in that they allow operations to be defined closer to the point of use rather than like the current C++ method of using function objects whose definition is sometimes far from the point of use.



(This is not including some of the boost libraries). I think the key point from lambdas is that they allow more concise and easy to understand code.



Share Improve this answer

answered Oct 17, 2008 at 23:34

Follow



Raindog

1,488 ● 3 ● 14 ● 28



-2

They offer better security using the multi threading in Java by implying in many cases the "final" option. So you are not error prone for multitasking.



Share Improve this answer

Follow

answered May 31, 2016 at 20:46



[pkiulian](#)

1

