Python descriptor protocol analog in other languages?

Asked 16 years, 3 months ago Modified 10 years, 8 months ago

Viewed 1k times

10

Is there something like the Python descriptor protocol implemented in other languages? It seems like a nice way to increase modularity/encapsulation without bloating your containing class' implementation, but I've never heard of a similar thing in any other languages. Is it likely absent from other languages because of the lookup overhead?

python language-features encapsulation

Share

Improve this question

Follow

asked Aug 29, 2008 at 9:24

cdleary

71.3k • 55 • 164 • 194

2 Answers

Sorted by: Highest score (default)

4

I've not heard of a direct equivalent either. You could probably achieve the same effect with macros, especially in a language like Lisp which has extremely powerful macros.

I wouldn't be at all surprised if other languages start to incorporate something similar because it is so powerful.

Share Improve this answer

Follow

answered Aug 29, 2008 at 9:39

Andrew Wilkinson

10.8k ● 3 ● 36 ● 38

0

Ruby and C# both easily let you create accessors by specifying getter/setter methods for an attribute, much like in Python. However, this isn't designed to naturally let you write the code for these methods in another class the way that Python allows. In practice, I'm not sure how much this matters, since every time I've seen an attribute defined through the descriptor protocol its been implemented in the same class.

EDIT: Darn my dyslexia (by which I mean careless reading). For some reason I've always read "descriptor" as "decorator" and vice versa, even when I'm the one typing both of them. I'll leave my post intact since it has valid information, albeit information which has absolutely nothing to do with the question.

The term "decorator" itself is actually the name of a design pattern described in the famous "Design Patterns" book. The Wikipedia article contains many examples in different programming languages of decorator usage: http://en.wikipedia.org/wiki/Decorator_pattern

However, the decorators in that article object-oriented; they have classes implementing a predefined interface which lets another existing class behave differently somehow, etc. Python decorators act in a functional way by replacing a function at runtime with another function, allowing you to effectively modify/replace that function, insert code, etc.

This is known in the Java world as Aspect-Oriented programming, and the AspectJ Java compiler lets you do these kinds of things and compile your AspectJ code (which is a superset of Java) into Java bytecode.

I'm not familiar enough with C# or Ruby to know what their version of decorators would be.

edited Mar 29, 2014 at 21:09

Mick MacCallum

130k ● 40 ● 281 ● 281

answered Sep 8, 2008 at 1:31

Eli Courtwright

193k ● 68 ● 218 ● 257

The descriptor protocol is not the same as the decorator pattern in Python -- I'm referring to the ability to make an object instance into another object's attribute that is "proxied" to dynamically. – cdleary Aug 29, 2008 at 19:11