why virtual keyword is used

See more: C#

★★★★★ 5.00/5 (1 vote)

Why do we use virtual override. We use this implementation without using the virtual keyword than why do we use this?

Posted 17-Oct-11 23:01pm
muzamil1

Updated 17-Oct-11 23:05pm v2
jim lahey

Add a Solution

3 solutions

Top Rated Most Recent

Solution 1

★★★★★

Marking a method as virtual means you can override it in derived classes, but you don't have to. If you mark a method as abstract, you are subsequently forced by the compiler to override it in any derived classes.

Further info:

http://www.compiledthoughts.com/2008/03/abstract-vs-virtual-explained.html[^]

Posted 17-Oct-11 23:07pm
jim lahey

Solution 2

★★★★★

Brought to you by google and MSDN:

"The override modifier is required to extend or modify the abstract or virtual implementation of an inherited method, property, indexer, or event."

—MSDN, override[^]

"The virtual keyword is used to modify a method, property, indexer, or event declaration and allow for it to be overridden in a derived class.For example, this method can be overridden by any class that inherits it."

—MSDN, virtual[^]

Cheers!

—MRB

Posted 17-Oct-11 23:10pm
Manfred Rudolf Bihy

Solution 3

☆☆☆☆☆

Understanding on virtual methods and polymorphic behaviour (provided by overriding them) is fundamental to understanding how to build OO systems. The technical description is as provided in the answers above: virtual methods may be overridden in subclasses and those overrides will be called when the base method is invoked on an instance of the subclass. (Abstract methods are the same except that you must provide an override – something which is clearer in the C++ 'pure virtual' terminology.)

But virtual methods are merely C#'s (and .Net's) way of providing polymorphism within a class hierarchy. To understand why and when you should use them, you need to understand how to write polymorphic designs. That's something which your course/online tutorial/book should have a section on.

Posted 17-Oct-11 23:52pm
BobJanova