Subject: Re: subclassing LineEdit is ugly
Posted by hojtsy on Mon, 17 Apr 2006 21:38:28 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 10 April 2006 04:08

```
bool MyLineEdit::Key(dword key, int count)
{
   if(key == K_SHIFT|K_TAB) {
      MyAlignChar();
      return true;
   }
   return LineEdit::Key(key, count);
}
```

Fine! Now let's discuss this solution. When you are duplicating part of a method just to replace a call in it, you can make two mistakes. Mistake 1 is failure to reproduce the context of the original call you are replacing. This was an easy case and you was familiar with it, yet you made this mistake. Now imagine this being attempted by somebody less familiar with library code AND less skilled than you, attempting the same. The error is that this overloaded method invokes MyAlignChar even if the LineEdit is read only. Another problem is that in the future the code you duplicated could change. In fact I hope that sometime it will change to configurable hotkeys. When such change happens you need to maintain, and change the duplicate version too. With lots of such duplication it becomes impossible to monitor all original locations for any possible change in the future.

luzr wrote on Mon, 10 April 2006 04:08pray that no other code does expect original AlignChar behaviour... See, by not makeing some methods virtual, you enforce stronger contract - you guarantee single behaviour.Expecting the default AlignChar behaviour is like expecting the default LeftDouble behaviour. Or the default behaviour of Key method for Shift-Tab as input paremeter. The point is that it is not a wise expectation. When you are calling AlignChar you mean that you want the selected text unindented. When I override this method in a subclass I am redefining what "unindent" means, not what Shift-Tab does. It is a conceptual difference - I would like my method called whenever "unindent" is needed, even if that is not when Shift-Tab is pressed. With this method as virtual I can choose from redefining "unindent" by overriding AlignChar, or changing the action triggered by Shift-Tab in writable mode by overriding Key.

luzr wrote on Mon, 10 April 2006 04:08adding characters at the end of console window only is too trivial to justice adding 10 virtual methods
So should every client of your library ask for each method to be virtual each time the need arises? That will be long story. And then the client need is weighted against what cost? The only cost is minimal CPU overhead. Almost any user benefit seem to justify that cost. BTW it is 6 virtual methods not 10, and they are not added just made virtual to enable more elegant subclassing.