Subject: Re: Controls & classes design questions
Posted by Mindtraveller on Thu, 16 Aug 2007 12:00:54 GMT
View Forum Message <> Reply to Message

mrjt wrote on Thu, 16 August 2007 13:46I think your design is mixing up access to controls and data at different levels and will probably lead to complications later. For instance: why does MyFile need access to the TabCtrl? Shouldn't that be managed by the MainWindow?
I rearranged structure and now MyFile has local tab object only. Main TabCtrl object access is needed only for creating and destroying specific file in editor (adding & removing tabs from control). And this adding & removal should be done inside MyFile class, I suppose.
For now, I have MyFile::ctor and MyFile::Close using TabCtrl (but not storing it`s reference anymore).

mrjt wrotePersonally I would do it something like this:

```
class MyFile : public GLCtrl {
    MyFile();

    virtual void GLPaint();
    //...
};
```

You need to inherit from GLCtrl and overload GLPaint to be able to draw anything. I agree, in this case inheritance is better than inclusion. Thanks.

mrjt wroteclass MainWindow : public TopWindow

```
{
    TabCtrl       tabs;
    Array<MyFile> files;
};
//...
void MainWindow::OnNew()
{

    MyFile &file = files.Create<MyFile>();
    file.SizePos();
    tabs.Add(file, "filename");
}
...
```

2) I'm confused about this, because MSC8 won't even compile the code you posted, let alone run it.
It compiled, actually. There even was no rintime error(!).
I`m shocked too.
mrjt wrote
3) No Ctrl or derived class is Moveable because they contain internal pointers to other Ctrls. However, what is not clear from the docs is that virtual methods are Moveable, but abstract methods are not (ie. virtual void Abstract() = 0; is not allowed). Array can store non-moveable types, but Vector et al. cannot.
Wow... it`s just unclear moment. Suppose author should focus on it in manual. Proposing new conatiner classes and more important, new approach of handling with containers and elements -

this must be explained much more than 2 articles. Examples must also be included. U++ is great, but it is unusual and complicated in moments, so good documentation is greatly needed.
mrjt wrote4) As far as I can see you shouldn't need a copy-constructor.
Hope that helps,
James If my class isn`t moveable, then of course not.
Thanks, James!
