Subject: Constructor with parameter
Posted by michael on Mon, 14 May 2007 05:56:16 GMT
View Forum Message <> Reply to Message

I have a class that creates a Dialog:

```
class EditDialog : public TopWindow
{
 Button b;

 void DoClose()
 {
  Close();
   }

 public :

 typedef EditDialog CLASSNAME;

 EditDialog()
 {
    SetRect(0, 0, 200, 50);
    Title("Eintrag editieren");
    Add(b.SetLabel("Close EditDialog").SizePos());
    b <<= THISBACK(DoClose);
 }
};
```

And i have to constructors:

```
EditDialog createDlg;
EditDialog editDlg;
```

How can i use variables when creating a new EditDialog? For example:

I want to create a new EditDialog newDlg with a different Title? Can i deliver a String with the constructor like this?

```
EditDialog newDlg(String Title);
```

Michael

Subject: Re: Constructor with parameter
Posted by mirek on Mon, 14 May 2007 08:07:32 GMT
View Forum Message <> Reply to Message

class EditDialog : public TopWindow

```
{
 Button b;

 void DoClose()
 {
  Close();
   }

 public :

 typedef EditDialog CLASSNAME;

 EditDialog(const char *title)
 {
    SetRect(0, 0, 200, 50);
    Title(title);
    Add(b.SetLabel("Close EditDialog").SizePos());
    b <<= THISBACK(DoClose);
 }
};
```

The question is whether it is worth the trouble.

In most cases, calling the Title method is as complicated as this.

Mirek

Subject: Re: Constructor with parameter
Posted by michael on Mon, 14 May 2007 08:57:21 GMT

If if use your class example, how does my constructor has to look like?

What i want to do?

I want to use the class EditDialog for editing existings database entries and for creating new database entries.

Therefor i want to construct two objects of my class EditDialog, one for example with the title "Create" and one with the title "Edit".

What does my Class and my constructors has to look like do realize this?

Michael

Subject: Re: Constructor with parameter
Posted by Zardos on Mon, 14 May 2007 09:32:16 GMT

I guess in the example from luzr you can not construct the two dialogs with different titles as members directly:

```
class MainDialog : public TopWindow
{
     ....

     // Does not work!
 EditDialog createDlg("Neuen Eintrag erstellen");
     EditDialog editDlg("Eintrag editieren");
     ....
}
};
```

Instead I would just call the Title function in the constructor in MainDialog:

```
class MainDialog : public TopWindow
{
     MainDialog()
     {
         createDlg.Title("Neuen Eintrag erstellen");
         editDlg.Ttile("Eintrag editieren");
     }
     ....

 EditDialog createDlg;
     EditDialog editDlg;
     ....
}
};
```

Subject: Re: Constructor with parameter
Posted by michael on Mon, 14 May 2007 09:50:44 GMT

Ok, that works.

Is there a way to get the Title of the Dialog after is was created?

I want to use this to do some if-selections with the title.

For example:

if(title == "create") do this
if(title == "edit") do that
Michael

Subject: Re: Constructor with parameter
Posted by mirek on Mon, 14 May 2007 11:01:33 GMT

michael wrote on Mon, 14 May 2007 05:50Ok, that works.

Is there a way to get the Title of the Dialog after is was created?

I want to use this to do some if-selections with the title.

For example:

if(title == "create") do this
if(title == "edit") do that
Michael

Well, this does not sound as a good idea to me at all, but if you insist, you can use GetTitle method (just beware, it returns WString - you can convert back to String by calling ToString - dlg.GetTitle().ToString()).

The correct solution would be either to use virtual methods, or some int member variable with enum (depends on situation).

Mirek

Subject: Re: Constructor with parameter
Posted by michael on Mon, 14 May 2007 12:28:03 GMT

Thanks Mirek,

you're right. Looks very ugly to me, too. I will think about a better solution for my problem. Maybe i should just create a separate class for my two dialogs...

Michael

Subject: Re: Constructor with parameter
Posted by mr_ped on Mon, 14 May 2007 19:46:21 GMT

View Forum Message <> Reply to Message

I case you go for quick hack I would rather test the pointer of object with &createDlg memory address, than title strings.

But where do you need these tests? Maybe you can avoid this completely (even with same class for both dialogs, or with very simple two wrapper classes derived from original class with just few functions overloaded).

Usually every time you finish writing some code, you can immediately write "second" version which will be much better.