
Subject: Access to parent class

Posted by [koldo](#) on Wed, 11 Dec 2019 09:31:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

Sometimes it is necessary for a child to get access to some data from a parent class. Specially in a multiple TopWindow project, parent pointer cannot be saved in a static location, as there are some instances of it.

A solution could be to give as argument a pointer to the parent, and spread it through all children until the one that needs it.

However I wonder if this could be a finer solution:

```
template <class T>
T *GetDefinedParent(Ctrl *ths) {
    T *main;
    while (ths->GetParent() != nullptr) {
        ths = ths->GetParent();
        if ((main = dynamic_cast<T*>(ths)) != nullptr)
            return main;
    }
    return nullptr;
}
```

And used like this from a children:

```
MyParentClass *parent = GetDefinedParent<MyParentClass>(this);
ASSERT(class);
```

```
int neededParameter = ~parent->importantParameter;
What do you think?
```

Subject: Re: Access to parent class

Posted by [mirek](#) on Thu, 12 Dec 2019 09:33:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

koldo wrote on Wed, 11 December 2019 10:31 Sometimes it is necessary for a child to get access to some data from a parent class. Specially in a multiple TopWindow project, parent pointer cannot be saved in a static location, as there are some instances of it.

A solution could be to give as argument a pointer to the parent, and spread it through all children until the one that needs it.

However I wonder if this could be a finer solution:

```
template <class T>
T *GetDefinedParent(Ctrl *ths) {
    T *main;
    while (ths->GetParent() != nullptr) {
        ths = ths->GetParent();
        if ((main = dynamic_cast<T*>(ths)) != nullptr)
            return main;
    }
    return nullptr;
}
```

```
    return main;
}
return nullptr;
}
```

And used like this from a children:

```
MyParentClass *parent = GetDefinedParent<MyParentClass>(this);
ASSERT(class);
```

```
int neededParameter = ~parent->importantParameter;
What do you think?
```

I am using this approach here and there, but never seemed like worthwhile addition to Core.h...

Subject: Re: Access to parent class
Posted by [koldo](#) on Thu, 12 Dec 2019 12:17:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, thank you. It is clear this is not perfect.

Subject: Re: Access to parent class
Posted by [mirek](#) on Thu, 12 Dec 2019 13:09:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

On the second thought, if we have things like GetChildCount, we can definitely have this as Ctrl method...

```
#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;
```

```
struct TestCtrl : Ctrl {
    virtual void LeftDown(Point p, dword keyflags) {
        TopWindow *w = GetAscendant<TopWindow>();
        if(w)
            w->Title(AsString(p));
    }
};
```

```
GUI_APP_MAIN
{
    TopWindow win;
    TestCtrl h;
    win << h.SizePos();
    win.Run();
}
```

}