

Subject: Polymorphic Array doubt
Posted by [koldo](#) on Sun, 31 Jul 2011 21:52:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello all

I have a doubt with polymorphic Arrays. In my cases methods are called properly but the destructor called is the one of the base class.

For example:

```
struct Number {
    virtual double Get() const = 0;
};

struct IntegerNumber : public Number {
    int n;
    virtual double Get() const { return n; }

    IntegerNumber(int n) : n(n) {}
};
```

```
struct StringNumber : public Number {
    String n;
    virtual double Get() const { return atof(n); }

    StringNumber(String n) : n(n) {}
};
```

If you include this in your code:

```
Array<Number> num;  
num.Add(new IntegerNumber(3));  
num.Add(new StringNumber("15.5555555555555555555555555555555555"));
```

you will get a "HEAP LEAKS" error when deletion, as for all Array members the Number destructor is called, instead of StringNumber or IntegerNumber.

Subject: Re: Polymorphic Array doubt
Posted by [dolik.rce](#) on Mon, 01 Aug 2011 05:46:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Koldo,

You need to provide a virtual destructor in your base class, so that the compiler knows it should look in the derived classes as well:

```
struct Number {  
    virtual double Get() const = 0;  
};
```

```
virtual ~Number(){}  
};
```

This will cause both ~Number() and ~{Integer,String}Number() to be called upon deletion. Which in turn removes the leak

Best regards,
Honza

Subject: Re: Poluporphyc Array doubt
Posted by [koldo](#) on Mon, 01 Aug 2011 07:53:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Oups!

Thank you Honza!

Subject: Re: Poluporphyc Array doubt
Posted by [Didier](#) on Mon, 01 Aug 2011 09:52:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Koldo and Honza,

I noticed that the virual destructor was sometimes missing in some controls, ex:
RichEditWithToolBar

Fortunatly this does not lead to any leaks (most of the time) thank's to the "everything belongs somewhere" in U++.

But this is a real bug.

Subject: Re: Poluporphyc Array doubt
Posted by [mirek](#) on Mon, 01 Aug 2011 10:50:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

Didier wrote on Mon, 01 August 2011 05:52Hi Koldo and Honza,

I noticed that the virual destructor was sometimes missing in some controls, ex:
RichEditWithToolBar

Fortunatly this does not lead to any leaks (most of the time) thank's to the "everything belongs somewhere" in U++.

But this is a real bug.

'virtuality' of destructor is inherited in C++. Therefore, as long as Ctrl has virtual destructor (and it does...), all derived classes have one too.

Mirek

Subject: Re: Poluporphy Array doubt
Posted by [Didier](#) on Tue, 02 Aug 2011 08:35:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Mon, 01 August 2011 12:50Didier wrote on Mon, 01 August 2011 05:52Hi Koldo and Honza,

I noticed that the virtual destructor was sometimes missing in some controls, ex:
RichEditWithToolBar

Fortunately this does not lead to any leaks (most of the time) thank's to the "everything belongs somewhere" in U++.

But this is a real bug.

'virtuality' of destructor is inherited in C++. Therefore, as long as Ctrl has virtual destructor (and it does...), all derived classes have one too.

Mirek

Humm,

I remember some work I did 10 years ago and this was not true then, but if gcc deals with it now : great !

Subject: Re: Poluporphy Array doubt
Posted by [mirek](#) on Wed, 03 Aug 2011 07:39:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Didier wrote on Tue, 02 August 2011 04:35mirek wrote on Mon, 01 August 2011 12:50Didier wrote on Mon, 01 August 2011 05:52Hi Koldo and Honza,

I noticed that the virtual destructor was sometimes missing in some controls, ex:
RichEditWithToolBar

Fortunately this does not lead to any leaks (most of the time) thank's to the "everything belongs somewhere" in U++.

But this is a real bug.

'virtuality' of destructor is inherited in C++. Therefore, as long as Ctrl has virtual destructor (and it does...), all derived classes have one too.

Mirek

Humm,

I remember some work I did 10 years ago and this was not true then, but if gcc deals with it now : great !

Well, this is so basic feature that if it would not work, compiler would be seriously broken...

Subject: Re: Poluporphyc Array doubt
Posted by [tojocky](#) on Thu, 04 Aug 2011 08:21:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

If in the base class is not set virtual destructor, and detect memory leak in example of Koldo, is this a bug?

In my case on win32 and linux 32 (ubuntu 11.04) have the same behavior (memory leak).

Regards,
Ion.

Subject: Re: Poluporphyc Array doubt
Posted by [cbpporter](#) on Thu, 04 Aug 2011 08:30:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

tojocky wrote on Thu, 04 August 2011 11:21Hello,

If in the base class is not set virtual destructor, and detect memory leak in example of Koldo, is

this a bug?

In my case on win32 and linux 32 (ubuntu 11.04) have the same behavior (memory leak).

Regards,
Ion.

I say it is a programming error. Without the virtual destructor, the compiler can't free the call the destructor of the string from StringNumber. It will always call the destructor for type T, in this case Number.

Subject: Re: Poluporphyc Array doubt
Posted by [koldo](#) on Thu, 04 Aug 2011 10:36:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes, that's it. I thought destructors were virtual by default.

Subject: Re: Poluporphyc Array doubt
Posted by [cbpporter](#) on Thu, 04 Aug 2011 10:38:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

koldo wrote on Thu, 04 August 2011 13:36 Yes, that's it. I thought destructors were virtual by default.

Nothing is virtual by default because of the added cost of the vtable. C++ inherits from C, so they wouldn't force you to use something with less performance and higher memory use by default .
