Subject: Re: Questions about static casting Polymorphic Array Elements, iterator, Ptr and Pte

Posted by mirek on Mon, 24 Dec 2012 13:40:05 GMT

View Forum Message <> Reply to Message

> navi wrote on Mon, 24 December 2012 04:57
> I am casting element at a later time depending on the "type" variable which can be access via the base type pointer/referance returned by the Array's index operator, to access the derived part of the object. I am under impression that since the Array is of the base type, I cant access the derived part of the object using the operator[] without casting.

Correct, but virtual methods of that type CAN.

Of course, it is not an universal truth and sometimes casting you suggest is really needed, but quite often you can better handle issues by good design of base class interface, with the right set of virtual methods doing things.

All in all, the whole purpose of such design is to handle all subclasses the same way...

To illustrate, say that your classes are intended for some graphical designer and that you want a menu with a set of operations for each type. Then you can do something like

```
if(a[1].type==2){
 triangle *m = static_cast<triangle *> (&a[1]);
       CreateMenuForTriangle(m);
}
```

or you can just use

```
struct shape {
   virtual void Menu(Bar& bar);
}
```

Mirek