

Re: Mutual object interdependency trouble

Source: <http://coding.derkeiler.com/Archive/Delphi/comp.lang.pascal.delphi.misc/2005-01/0491.html>

From: Tom de Neef (*tdeneef_at_qolor.nl*)

Date: 01/26/05

Date: Wed, 26 Jan 2005 09:38:49 +0100

"Rune" <rune[insert current year]@runevision.com> schreef in bericht
news:xzCJd.85270\$Vf.3768803@news000.worldonline.dk...

> *I've encountered a problem in the design of a game I'm working on.*

>

> *I need certain information to be available practically everywhere.*

>

> *As a simplified example, I have these four resources:*

>

> *Creature*

> *Level*

> *Maze*

> *Images*

>

> *Each of them needs all the ones below. The Maze needs the Images (so it
> can*

> *draw itself); the Level needs the Maze (which it is build after) and the*

> *Images (so it can draw itself); and the Creature needs the Level and Maze*

> *(so it can navigate) and the Images (so it can draw itself). Creature*

> *objects also need data about other Creature objects.*

>

> *There is no interdependency here. The problem is that it's very cumbersome*

> *to manually have to hand over references to all the needed resources after*

> *new objects are created. It'd be nice if objects after being created could*

> *just fetch the needed references themselves.*

>

> *This led me to the idea of some kind of Keeper class that keep references*

> *to*

> *the various resources. Objects could then just be given a reference to*

> *this*

> *Keeper object, and then they could request all needed resources from the*

> *Keeper object.*

>

> *However, this creates the mutual interdependency. The Keeper class must*

> *know*

> *about the four other classes, and the other classes (except the Images*

> *class). must also know about the Keeper class.*

>

> As we know, in Delphi classes are bundled in units. As I understand it, a
> unit is one big text file that contain one or more classes. Since my game
> is big, I have several units to keep a better overview and the four
> mentioned classes are all in different units.
>
> The problem is that it seems that in Delphi, if Unit A knows about Unit B,
> then Unit B can't know about Unit A. This means that the Keeper concept
> doesn't seem to work.
>
> I'm not sure what to do. I've had classes about programming patterns, but
> I'm used to Java, not Delphi, and I only have limited practical experience
> with complex projects...
>
> Anyone have any ideas?
>
> Rune
> --
>

In Delphi, the dependency between units is in the interface section. Unit A can use unit B and B can use data from A, as long as there is no circular reference via the interface sections. So, you could have:

```
unit A;  
interface uses B;  
implementation  
...  
end.
```

```
unit B;  
interface  
...  
implementation uses A;  
...  
end;
```

This allows you a lot of freedom in separation code across units. The limits are reached when you want to declare two types that depend on each other in their declaration, such as:

```
type  
  Ta = class(Tb);  
  Tb = class  
    c : Ta;  
  end;
```

In such cases Ta and Tb need to be in the same unit. What you could still do though, is change the declaration of Tb into:

```
type  
  Tb = class  
    c : Tobject;  
  end;
```

comp.lang.pascal.delphi.misc: Re: Mutual object interdependency trouble

Now, Ta can be in a unit with 'interface uses B' and Tb in unit B with 'implementation uses A'; But you have to introduce some extra code to typecast the variable c.

Tom