Subject: What do you do first when you start developing an application?
Posted by lectus on Thu, 13 Dec 2012 20:13:10 GMT
View Forum Message <> Reply to Message

Some people recommend that we create the database first and then we create the application interface and logic to handle that.

What's your opinion?

Subject: Re: What do you do first when you start developing an application?
Posted by zsolt on Thu, 13 Dec 2012 21:00:22 GMT
View Forum Message <> Reply to Message

I write a textfile with requirements first.
Then I start implementation with one of the requirements.
I create a very small part of the database and code fulfilling only the selected requirement. I like creating unit tests first, but not always.
After the first requirement I continue with an other one and so on.

This is a natural way of software development, I think.
Nowadays this is called agile development

Subject: Re: What do you do first when you start developing an application?
Posted by dolik.rce on Fri, 14 Dec 2012 11:02:51 GMT
View Forum Message <> Reply to Message

I usually follow the same flow as zsolt, minus the writing to textfile (I'm very lazy)

Also, I usually start without GUI (if possible), writing the main functionality as console app, testing it with some fixed values representing user inputs and only later binding the functions to GUI.

Honza

Subject: Re: What do you do first when you start developing an application?
Posted by mirek on Fri, 14 Dec 2012 13:13:41 GMT
View Forum Message <> Reply to Message

lectus wrote on Thu, 13 December 2012 15:13Some people recommend that we create the database first and then we create the application interface and logic to handle that.

What's your opinion?

Well, if it is some client/server SQL stuff, I usually develop GUI and DB together.

Sometimes even going so far that I design dialog layout first, then copy dialog widget ids into .sch file and make a SQL table out of it.

Generally, I always liked to create simple skeleton app first, then continually add features, improving things in all parts.

I do not even care too much about requirements. Usually it is faster for me to create more iterations of code - usual problem being is that to interrogating users about requirements is time consuming process, but it is much easier to them to tell you what is wrong with the application (something they can see and try to use).
