Subject: does U++ use Smalltalk(Java) approach? [SOLVED] -No.
Posted by fudadmin on Mon, 12 Dec 2005 13:21:08 GMT

View Forum Message <> Reply to Message

Hi, I'm writing my article...
So, to be sure I want your confirmation and/or comments:

1. "U++ does not use Smalltalk(Java) approach - where everything derived from a single inheritance tree and at the root of the tree is a class called Object – the same class that each Smalltalk(Java) container holds, because...[I want your comments here]"

2. what are the differences then between U++'s Value and Java's Object?

Subject: Re: does U++ use Smalltalk(Java) approach?
Posted by mirek on Mon, 12 Dec 2005 13:50:30 GMT

View Forum Message <> Reply to Message

1.: Yes. Smalltalk model is unsuitable for C++. For C++ it is better to have unrelated ("decoupled") classes and class hierarchies and glue them together using templates (usually containers).

2.: Value is intended just for storing 'values'. Now it is a little bit hard to explain what exactly 'value' is, and at the same time it is very important for U++ programming style.... (there is quite a lot of effort put into dealing with various types of objects - values, pick values, non-copyable objects - non-values etc...)

Perhaps better is to provide some examples: number is a Value, date is a Value, String is a Value, etc...

Maybe a right descrition is that "for value, object identity does not matter, just its content (well, ehm, its value).

I wish I could explain that more....

Not also that Value supports some value-like operations, like comparing values (by their, well, uhm, values...) converting similar types of values (you can assign 'int' to the Value and then assign this value to 'double', same for Date / Time etc.).