Subject: Java coffee...
Posted by bytefield on Wed, 17 Sep 2008 17:14:41 GMT
View Forum Message <> Reply to Message

I want to know your opinion about Java language, programming in Java and web applications made with Java.

I don't know programming in Java, well, maybe just parts from Java which are like C++(for me Java is a C++ without some features). I started programming with C++ (I'm sure I didn't made a mistake maybe the mistake is that i don't know english very well to be clear in my expression), i was also learning C as a subset of C++, I've read one book about Visual Basic.NET(I dislike this language) and then tried php(with [X]HTML and CSS) for web programming, but web programming didn't attracted me at all... From when I'm using Linux I've tried bash programming but just to solve my problems and to automate some process.

Nowadays one professor from my university is seeking to employ me and one classmate for working in Java, in special with Struts and another web technology from IBM (which i don't remember now) but i know their IDE is based on Eclipse. I have to specify that we both don't know Java and after talking with my classmate i found that he didn't like programming in Java too. (well maybe he like a bit if it will bring some money to him). I've showed also how GUI programming looks in C++ with Upp and he was astonished how simple and few lines of code you write in it compared with GUI programming in Java (We have some books about Java).

I'm asking if is worth learning Java and get employed(part time) by our professor or to stay with C++ and learning it better and better even if now I don't get a job for it. I would get that job just to get some new knowledge which maybe some day will help me. I think that 150 euro don't worth to learn Java yak, they are exploiting students to do programming for they .

I know that is a long post and maybe still incomplete... but on your answers stay my future in programming well, maybe not at all because i'm thinking to not that job because i don't like to be employed.

Subject: Re: Java coffee...
Posted by cbpporter on Wed, 17 Sep 2008 18:19:24 GMT
View Forum Message <> Reply to Message

Well I for one absolutely hate Java .

First of all I hate the language. It is extremely unexpressive. My biggest complaint is that you don't have delegates. Listeners are horrible when misused to simulate delegates. I hate that it does not have an unsigned int and when I need to do binary processing I get nice code bloat. I hate that you have to explicitly catch all exceptions. This is a very good idea on paper, but practice has shown that most people just wrap the code in a dummy try-catch with an empty catch clause because they don't know what to do with that exception. The generics from Java are laughable. I actually done a benchmark and using a class similar to Vector in Java with fundamental times is 10 or 20 times slower (don't remember which one).

I also hate the Java library. Just reading out of a damn file may require up to 4 classes, depending on buffering and how you obtain the filename. And Java toolkits have an obscene obsession with Xml configuration files. A lot of them with very redundant stuff.

On the other hand, I have great respect for what Java is trying to do. The platform independence is great. Performance was bad a lot of time ago, but now it is pretty good. I no longer have such a great opinion about GC because of U++ and it's deterministic way to handle resources, but still GC solves the problem cleaner than RAII. Compilation speed is great. You get an extensive set of libraries for anything that you could possible want.

I am still often tempted to solve my issues with C++ and the tool chain by switching over to Java. Then I need to write a delegate or something, and my hate for Java revives.

Actually, my hate for Java is more related to the fact that it could have been the needed innovation in this domain, but somehow it managed to completely screw it up when compared to my standards. It is not as much hate, as disappointment towards something I wanted to love, but just couldn't because of it's design.

Yet I had to work sometimes in it, and I do get used to it. It is definitely worth learning, because it has it's place in this world. Also, Java 7 looks quite promising.

But IMO it will never catch up. There is a Java like product in this world that pretty much got everything right (not on the first try though, which was pretty bad). It is called .NET. I love .NET. It is like a copy of Java which is so much better. I even features true matrices for when you do graphic processing (the lack of that double indirection from jagged matrices is a real performance boost in this case), and even pointers when you need to get down and dirty. Also library is more to my taste.

The problems with .NET are it's size and lack of portability. Do I really need to install that huge runtime? I don't need WPF. Silverlight? Sorry, I'm a Flash fan. And even with that huge runtime, it only runs on Windows.

On the other hand, Mono is growing up slowly. It is also possible to get a 7MB minimal Mono runtime.

This is a long post, so I'll stop here.

The conclusion: I hate Java, but in the same time I like the idea and find Java something that every programmer should know. It you know C++, you'll be able to code Java in under a week. Just need to learn the library next. So I think you should go ahead with learning Java. And check out .NET if you have time, preferably after some experience acquired in Java, so you can best appreciate the features that it adds to the Java idea.

Subject: Re: Java coffee...
Posted by bytefield on Wed, 17 Sep 2008 20:33:24 GMT
View Forum Message <> Reply to Message

Well, you make me thinking to take the book from bookcase and give Java a run. I was scared a bit by the huge library that Java have . The next question is to start learning it with an IDE (Eclipse, NetBeans, etc) or to start with old good way of doing console programs first? I also looked at SWT, is it better than AWT or Swing? Seems easy to create an application using SWT (with Eclipse) than using Swing or AWT.
I also wait others opinion if there are more Java programmers.

Subject: Re: Java coffee...
Posted by captainc on Wed, 17 Sep 2008 21:00:00 GMT
View Forum Message <> Reply to Message

I am similar to cbpporter in my experience with Java.
I dislike using Java guis; from a usability perspective, they are clunky.
I love their goal though: "write once, run anywhere".
I look to achieve this through the use of C++ (U++) and Python. I use the Python C API and call that from C++ applications. This allows you to have all of the python modules available, but you only have to include with your program the ones that you use. In one project, I incorporated a FTPS server (FTP over SSH) using Python and added it to my C++ application which was doing network sockets and other stuff.

On the other hand, learning another language will help you become a better programmer. Especially understanding the differences between languages. Java is easy to learn, especially for a C++ developer. It is much simpler.

I would give it a shot just for the learning experience, although I predict you will have a similar experience as cbpporter and I.

Subject: Re: Java coffee...
Posted by mr_ped on Thu, 18 Sep 2008 07:43:06 GMT
View Forum Message <> Reply to Message

I did some small projects (most of them for school or for fun) in java in my past. The language itself is quite similar to C++, although there are major differences. GC, no pointers, and everything-is-object and maybe some more. So you may sometimes end with a tad different design of program structure, if you compare java with C++ version.

Then there's the API... the basic java libs are what they are, once you learn to use them, you can use them, it may be quite different to clib, but would you start with Java and not C++, you would probably have bitter taste from trying clib then.

Then comes enterprise level. XML, Struts, Swing and all those other fancy buzzwords. The total amount of API calls available is *huge* for my mind, and I can't work with this effectively. I think it's aimed at memory-strong programmers who can't do 6 lines of code doing some actual *stuff* (at least not without including some major bug there), but they don't have any problem to call 150

API calls out of head and form some application of them, once their analyst put some basic structure ahead of them.
You need then just couple of strong architects of system to lay down the basic infrastructure and code the couple of modules which do something new, and hire tens of those API programmers, and let them finish the stuff (usually twice or more faster than would you keep those 1-2 analysts doing it all). So all these huge API and modules IMHO allow enterprises to split the work meaningfully to get at least that twice+ better development times (for mere price of 10x more people developing it). And that's very good scalability from business point of view. Because you can't add those analytic programmers at project and expect the development time going down, after certain threshold adding more programmers will either not improve development time or bring it down. In this aspect I do suspect those buzzword technologies really work. Although I often can't understand how and I'm always very suspicious about them and their usage.

Anyway, if you want good job in big corporation, going Java+struts+other enterprise level technologies is a right way, and you will have very good chance to find a good job with such knowledge. (With U++ you will hardly find any job, it's more suitable to be the vendor of SW solution itself and sell final applications done with U++, as there are basically zero corporations using U++ for development right now. Java... is different story , and .NET did spread even faster.. go figure.)
