

Re: input stream 101

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.help/2005-08/msg00131.html>

- *From:* "hawat.thufir@xxxxxxxxxx" <hawat.thufir@xxxxxxxxxx>
 - *Date:* 5 Aug 2005 03:55:01 -0700
-

blmblm@xxxxxxxxxxxxxxxx wrote:

....

> >Partly it's just to apply that idea and see what happens, I think
> >you've shot down all my other reasons ;)
>
> But in a kind and gentle way, I hope. :-)

Very much so, thank you.

....

> Wow. To me the benefit (automatically generating documentation that
> looks like the "standard" documentation) justifies a certain amount
> of ugliness in the source — and anyway aside from the use of the
> javadoc tags (e.g., "@param"), I don't find that my comments look
> much different from the way they did before I discovered javadoc.
> May be a "YMMV" thing, though.

I've also never used it, yet. It looks like it's just like "javac" so
I'll probably add it to my build file. It sounded perhaps more
intimidating that it really is, to me.

....

> I think there's also a school of thought that claims that if you
> choose your method and variable names really carefully, you won't need
> comments — e.g., if you're writing a Circle class, does getArea()
> really need a comment to explain what it does? I don't think this
> always works, but sometimes it does.

I think that breaks down once you move away from circles and into
widgets. If you have a bit of high school geometry then it's
accessible. I *imagine* that there are going to be times that heavy
documentation is required. For instance, a medical application. Unless
you're a doctor it might be "greek" to follow that naming convention.
It'd probably be Latin, actually.

But, yeah, I at least subscribe to the school of thought that the name
of a method/primitive/class/etc is very important. Not that I always
follow it ;)

Re: input stream 101

I have a *little* Fortran experience, and when I've looked over my (usually undocumented) Fortran the name for a subroutine is what I key to.

> [snip]
>
>>Yes, Test16 is a crappy name, I kept it largely because I just wanted
>>to use it as someone elses code and not muck with it. I believe that
>>it now makes sense to go back, get rid of the BasicTidy interface and
>>rename Test16 to BasicTidy.
>
> That's what I'd probably suggest.
....

I did that, and am happier with the results. I'm a bit more comfortable about going into the guts of the borrowed code, too.

One thing I found kind of interesting is that I can send a url like "file:///home/thufir/html/foo.html" through what I have. I know that JTidy can be run from the command line on its own, but I like having my own java code to wrap around it. I'm not sure that would work on a Windows computer, so it's a bit fragile, but it does work. Also, I have a hunch that passing around Strings isn't so great, but it works for now.

I'm thinking about how to approach mass-tidy-izing. Performance isn't an issue, so simply iterating file[n].html, for n=1 to 99, might be my next step. To make it a bit more practical, I'm going to have to research how to do a "ls" (linux) or "dir" (windows) command to read in the contents of a directory. Once I have the contents of a directory, perhaps in an array or List, then it should be fairly easy to iterate through. I could even add a GUI, which is something I'd like to do. I've just been caught up in some other stuff and haven't had time :(

> And as I'm sure you know -- you probably progress faster if you're
> willing to listen to other people's ideas, and keep the ones that make
> sense to you.

Absolutely.

> If you're going to run from the command line, you could pass in
> the URL and output file name as command-line parameters. main()
> has a parameter of type String[], right? which at runtime contains
> an array of Strings, one for each command-line argument. I think
> this would be a very easy change from your current code.

That's an idea, I'll play with that soon, hopefully.

....

Re: input stream 101

Re: input stream 101

> And if the intended purpose of this code is to mass-convert 100+
> files, my guess is that you'd be better served by something you can
> run from the command line, because then Well, I'm thinking
> in Unix/Linux terms, where the obvious way to do this would be with
> shell-script constructs to loop over all the files, etc. I seem to
> remember messages from you to a Linux mailing list? so maybe that
> approach would work for you.

Heh, which list?

Yeah, but this primarily about learning Java and only secondarily about this particular project. Really, I could've just typed some stuff into a spreadsheet :)

> I'm guessing this is another artifact of your having adapted the
> code Nothing wrong with that, by the way; I think it's a fine
> way to get started using stuff you don't understand 100%. But then
> as you learn, you figure out what parts can be modified to suit your
> purposes better.

Yes. I'm having trouble figuring out the capabilities of JTidy, so that's holding me back right now. On the flip side of that, I'm looking

at xerces and xalan to tie into this, now that I'm a bit more comfortable with the build process for third party jar's.

I saw a neat book on hibernate and xerces, so I'm thinking that if I can get the HTML into XML then I can pretty much follow the recipe to get it into a database. I need to do a lot of reading to get from point "a" to "b" on that, though.

> Anyway, an easy change to your code would be to remove "extends
> Runnable" from your interface definition, and just call your run()
> method directly from the main program, rather than creating a Thread
> object and calling its start(). If you wanted to avoid possible
> confusion for readers who think run() must have something to do with
> threads, you could rename it convert() or doIt() or whatever.

Thanks, I didn't know how to do that. Much more comfortable with it now. As I said, just somewhat hampered by not knowing the JTidy API.

thanks,

Thufir

.

- *Follow-Ups:*
 - ◆ **Re: input stream 101**
 - ◇ *From:* blmblm

- *References:*
 - ◆ **input stream 101**
 - ◇ *From:* hawat.thufir@xxxxxxxxxx
 - ◆ **Re: input stream 101**
 - ◇ *From:* hawat.thufir@xxxxxxxxxx
 - ◆ **Re: input stream 101**
 - ◇ *From:* blmblm
 - ◆ **Re: input stream 101**
 - ◇ *From:* hawat.thufir@xxxxxxxxxx
 - ◆ **Re: input stream 101**
 - ◇ *From:* blmblm

- Prev by Date: **Re: c.l.j.help charter**
- Next by Date: **One class destroying another**
- Previous by thread: **Re: input stream 101**
- Next by thread: **Re: input stream 101**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**