

Writing your own Java I/O Decorator

Okay, you know the Decorator Pattern, you've seen the I/O class diagram. You should be ready to write your own input decorator.

How about this: write a decorator that converts all uppercase characters to lowercase in the input stream. In other words, if we read in "I know the Decorator Pattern therefore I RULE!" then your decorator converts this to "i know the decorator pattern therefore i rule!"

Don't forget to import
java.io... (not shown)

First, extend the FilterInputStream, the abstract decorator for all InputStreams.

```
public class LowerCaseInputStream extends FilterInputStream {  
    public LowerCaseInputStream(InputStream in) {  
        super(in);  
    }
```

```
    public int read() throws IOException {  
        int c = super.read();  
        return (c == -1 ? c : Character.toLowerCase((char)c));  
    }
```

```
    public int read(byte[] b, int offset, int len) throws IOException {  
        int result = super.read(b, offset, len);  
        for (int i = offset; i < offset+result; i++) {  
            b[i] = (byte)Character.toLowerCase((char)b[i]);  
        }  
        return result;  
    }
```

Now we need to implement two read methods. They take a byte (or an array of bytes) and convert each byte (that represents a character) to lowercase if it's an uppercase character.



No problem. I just have to extend the FilterInputStream class and override the read() methods.

Test out your new Java I/O Decorator

Write some quick code to test the I/O decorator:

```
public class InputTest {  
    public static void main(String[] args) throws IOException {  
        int c;  
        try {  
            InputStream in =  
                new LowerCaseInputStream(  
                    new BufferedInputStream(  
                        new FileInputStream("test.txt")));  
  
            while((c = in.read()) >= 0) {  
                System.out.print((char)c);  
            }  
  
            in.close();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Just use the stream to read characters until the end of file and print as we go.

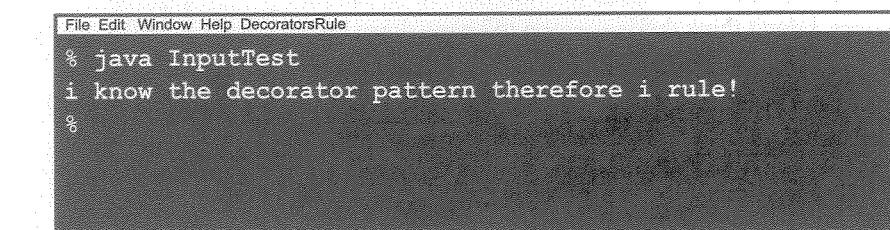
Set up the FileInputStream and decorate it, first with a BufferedInputStream and then our brand new LowerCaseInputStream filter.

I know the Decorator Pattern therefore I RULE!

test.txt file

You need to make this file.

Give it a spin:



```
File Edit Window Help DecoratorsRule  
% java InputTest  
i know the decorator pattern therefore i rule!  
%
```

REMEMBER: we don't provide import and package statements in the code listings. Get the complete source code from the headfirstlabs web site. You'll find the URL on page xxxiii in the Intro.