

# THE COMPOSITE PATTERN

## HERE IS HOW INPUT OUTPUT IN JAVA WORKS -

### INPUTS AND OUTPUTS ARE REPRESENTED AS STREAMS

INPUTS CAN BE - FILES, MEMORY BUFFERS (ARRAYS),  
PIPES, NETWORK CONNECTIONS, SYSTEM.IN (KEYBOARD INPUT)

LIKEWISE OUTPUTS CAN BE - FILES, MEMORY BUFFERS (ARRAYS),  
PIPES, NETWORK CONNECTIONS, SYSTEM.OUT AND SYSTEM.ERROR  
(SCREEN OUTPUT)

STREAMS ARE DATA FLOWS - INPUT STREAMS  
ORIGINATE FROM INPUTS, AND OUTPUT STREAMS  
FROM OUTPUTS

## JAVA PROVIDES OBJECTS TO ENCAPSULATE INPUTS, OUTPUTS AND STREAMS

```
    }  
  } else {  
    if (file.IsZipped()) {  
      if (file.IsBinary()) {  
        processEncryptedUnZippedBinaryFile(file);  
      } else {  
        processEncryptedUnZippedTextFile(file);  
      }  
    } else {  
      if (file.IsBinary()) {  
        processUnEncryptedUnZippedBinaryFile(file);  
      } else {  
        processUnEncryptedUnZippedTextFile(file);  
      }  
    }  
  }  
}
```

AND ADDING ONE  
MORE VARIABLE,  
SAY FOR WHETHER  
THE FILE IS ENCODED  
FOR WINDOWS OR UNIX  
WOULD INVOLVE  
CHANGING ALL THIS  
CODE!

**THIS CODE VIOLATES THE OPEN-CLOSED  
PRINCIPLE: CLASSES SHOULD BE OPEN  
FOR EXTENSION BUT CLOSED FOR  
MODIFICATION**

**A TERRIBLE WAY OF HANDLING THIS:**

**THE NUMBER OF  
SPECIAL CASES  
IS ALREADY  
OUT-OF-CONTROL..**

```
if (file.IsEncrypted()) {  
    if (file.IsZipped()) {  
        if (file.IsBinary()) {  
            processEncryptedZippedBinaryFile(file);  
        } else {  
            processEncryptedZippedTextFile(file);  
        }  
    } else {  
        if (file.IsBinary()) {  
            processUnEncryptedZippedBinaryFile(file);  
        } else {  
            processUnEncryptedZippedTextFile(file);  
        }  
    }  
} else {  
    if (file.IsZipped()) {  
        if (file.IsBinary()) {  
            processEncryptedUnZippedBinaryFile(file);  
        } else {  
            processEncryptedUnZippedTextFile(file);  
        }  
    } else {  
        if (file.IsBinary()) {  
            processUnEncryptedUnZippedBinaryFile(file);  
        } else {  
            processUnEncryptedUnZippedTextFile(file);  
        }  
    }  
}
```

**AND ADDING ONE  
MORE VARIABLE,  
SAY FOR WHETHER  
THE FILE IS ENCODED  
FOR WINDOWS OR UNIX  
WOULD INVOLVE  
CHANGING ALL THIS  
CODE!**

**THIS CODE VIOLATES THE OPEN-CLOSED  
PRINCIPLE: CLASSES SHOULD BE OPEN  
FOR EXTENSION BUT CLOSED FOR  
MODIFICATION**

**JAVA OBVIOUSLY DOES  
NOT DO IT THIS WAY,  
RATHER IT USES -**

# **THE DECORATOR PATTERN**