* Avoid unnecessary comments in code?

# [**What's the purpose of try-with-resources statements?**](https://stackoverflow.com/questions/17739362/whats-the-purpose-of-try-with-resources-statements)

It was introduced because of some resources used in Java (like SQL connections or streams) being difficult to be handled properly; as an example, in java 6 to handle a **InputStream** properly you had to do something like:

InputStream stream = new MyInputStream(...);

try {

// ... use stream

} catch(IOException e) {

// handle exception

} finally {

try {

if(stream != null) {

stream.close();

}

} catch(IOException e) {

// handle yet another possible exception

}

}

Do you notice that ugly double try? now with try-with-resources you can do this:

try (InputStream stream = new MyInputStream(...)){

// ... use stream

} catch(IOException e) {

// handle exception

}

and **close()** is automatically called, if it throws an IOException, it will be supressed (as specified in the [Java Language Specification 14.20.3](https://docs.oracle.com/javase/specs/jls/se8/html/jls-14.html#jls-14.20.3)) . Same happens for **java.sql.Connection**

<https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example>

collection:

