JAVA 7 features

Feature 1 **: try-resource**

<https://stackoverflow.com/questions/213958/new-features-in-java-7>

1. Earlier .

BufferedReader br = new BufferedReader(new FileReader(path));

try {

return br.readLine();

} finally {

br.close();

}

becomes:

try (BufferedReader br = new BufferedReader(new FileReader(path)) {

return br.readLine();

}

You can declare more than one resource to close:

try (

InputStream in = new FileInputStream(src);

OutputStream out = new FileOutputStream(dest))

{

// code

}

WHY ? whats the use ?

Ans :

A resource is an object that must be closed once your program is done using it. For example a File resource or JDBC resource for database connection or a Socket connection resource. Before Java 7, there was no auto resource management and we should explicitly close the resource once our work is done with it. Usually, it was done in the finally block of a try-catch statement. This approach used to cause memory leaks and performance hit when we forgot to close the resource.

So if try – resource is used then resource will be closed automatically without any need of manually closing in finally block .

* in a scenario , try- resource is used with finally …and exception occurs in try and also also in finally then **the try exception will be displayed** .
* but without try-resource it would be finally exception displayed which is not useful as we need to find the actual exception.

\*\* to try-resource to work the Resource i.e it may be like FileReader , FileInputstream , Bufferreader etc should implement **java.lang. Closable** interface

If any resource you are using which is not implementing this clonable resource then it will nt allow to use the resource in try block i.e comple time error

Ex: 1.

try( FileReader file = new FileReader("inputy.txt"); ){

// do something

}

Here the Resource is FileReader which internally implemts Closable Interface and provide the .close method implementation.

public class FileReader extends InputStreamReader{}

public class InputStreamReader extends Reader{}—close() implementation is given in this class

public abstract class Reader implements Readable, **Closeable** {}

ex2 : try(Fiile ff = new File(“input.txt”);){

}

* Compile time error : resource doesn’t implement closeable interface