Autoclose Resources with a Try-With-Resources Statement

The **try-with-resources** statement in Java ensures that each resource is closed at the end of the statement. A resource is an object that must be closed after the program is finished using it, such as a file or a socket. Resources are closed automatically when the try block is exited. This statement can manage multiple resources and is a significant improvement over traditional try-catch-finally blocks.

Here's an example:

```
package com.mphasis.exceptions;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class TryWithResourcesExample {
    public static void main(String[] args) {
        try (FileReader fr = new
FileReader("d://example.txt");
             BufferedReader br = new
BufferedReader(fr)) {


            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
           System.out.println("Exception");
        }
    }

}
```

In this example:

- BufferedReader and FileReader are resources that need to be closed.
- The try-with-resources statement ensures that both BufferedReader and FileReader are closed automatically.