



Data flows

by Vlad Costel Ungureanu
for "Learn Stuff"

```
open communication channel; //new FluxClass ([params]);  
while (flow still has data in it) {  
    // Read/write data;  
    // operations  
    //read(), write()  
}  
close channel; //close()  
manage exceptions;
```



```
// Write
FileOutputStream fos = new FileOutputStream("test.ser");
ObjectOutputStream out = new ObjectOutputStream(fos);
out.writeObject(new Date());
out.writeObject("Current Date");
out.writeInt(12345);
out.flush();
fos.close();
```

```
//Read
FileInputStream fis = new FileInputStream("test.ser");
ObjectInputStream in = new ObjectInputStream(fis);
Date data = (Date)in.readObject();
String mesaj = (String)in.readObject();
int i = in.readInt();
fis.close();
```



Exception Management

by Vlad Costel Ungureanu
for "Learn Stuff"

```
try {  
    // instructions  
    method1();  
    method2(); // generates an error  
    method3();  
}  
catch (ExceptionType1 ex1) {  
    // handle exception type 1  
}  
catch (ExceptionType2 | ExceptionType3 ex23) {  
    // handle exception type 2 or 3  
}  
finally {  
    // code that is executed regardless if there is an exception or not  
}
```

- ✓ Exception handling is not a feature!
- ✓ Exception handling is a constraint!
- ✓ Correct exception management is a mandatory part of any application

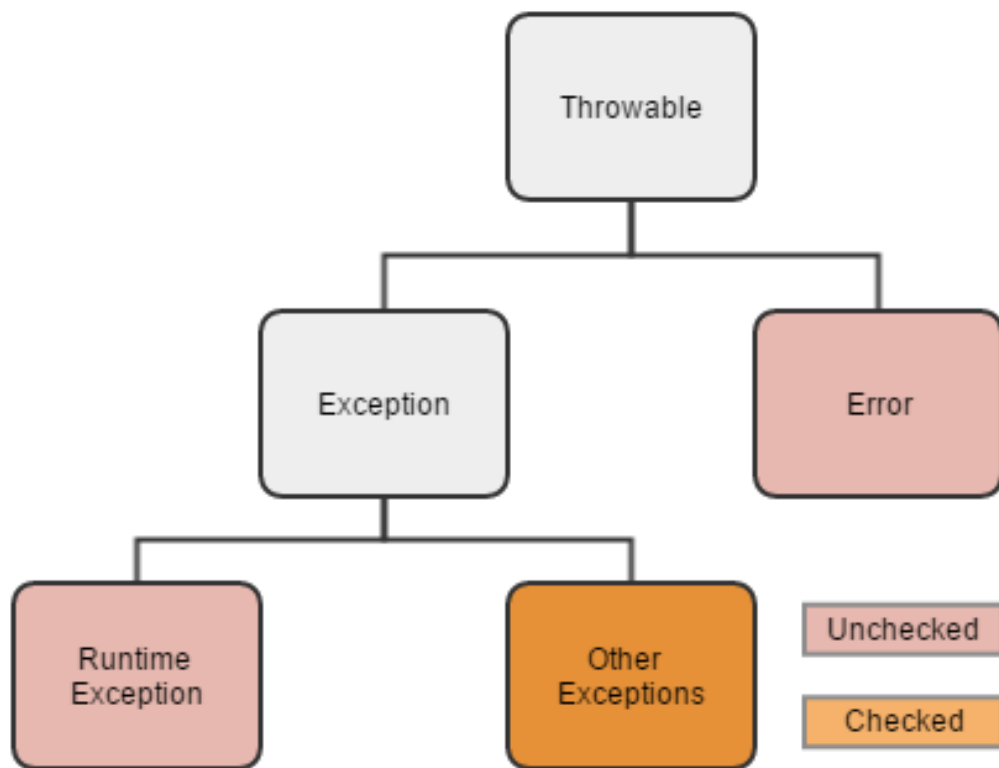
```
public static void readFile(String fis) {  
try {  
    FileReader fr = new FileReader(File file = new File(fis));  
    BufferedReader br = new BufferedReader(new FileReader(fin));  
    String line;  
    while ((line = br.readLine()) != null) {  
        System.out.println(line);  
    }  
    fr.close();  
} catch (FileNotFoundException e) {  
    System.err.println("File " + fis + " not found!");  
} catch (IOException e) {  
    System.out.println("Reading error!");  
}
```



```
public static void readFile(String fis) throws FileNotFoundException, IOException {
    FileReader fr = new FileReader(File file = new File(fis));
    BufferedReader br = new BufferedReader(fr);
    String line;
    while ((line = br.readLine()) != null) {
        System.out.println(line);
    }
    fr.close();
}

public static void main(String args[]) {
    try {
        readFile(new File());
    } catch (Exception e){
        // do stuff
    }
}
```

```
try (Connection con = createConnection(); Statement stmt = con.createStatement();  
    ResultSet rs = stmt.executeQuery(query)) {  
    return (rs.next() ? rs.getObject(1) : null);  
} catch (SQLException e) {  
    System.err.println(e);  
}
```



- ✓ Read two matrix from two distinct files
- ✓ Execute matrices operations (addition, multiplication) and write the results in a file, making sure that that each time you write you don't write over the previous results
- ✓ Make a call hierarchy for 3 or more method in different classes. Throw errors in the last method call and make sure they reach the first method in the call hierarchy
- ✓ Create a custom exception using inheritance and use the custom exception in the previous example
- ✓ Write examples that generate(without using "throw") and properly manage the following exceptions: NullPointerException (accessing a class method), ArithmeticException (for division by 0), NumberFormatException(for parsing an integer), NotSerializableException (for serialization of an object)

THANK YOU!

Vlad Costel Ungureanu
ungureanu_vlad_costel@yahoo.com

This is a free course from [LearnStuff.io](https://learnstuff.io)
– not for commercial use –