

Exceptions

1 Exceptions

There are at least two classes that extends to exception.

Class diagram

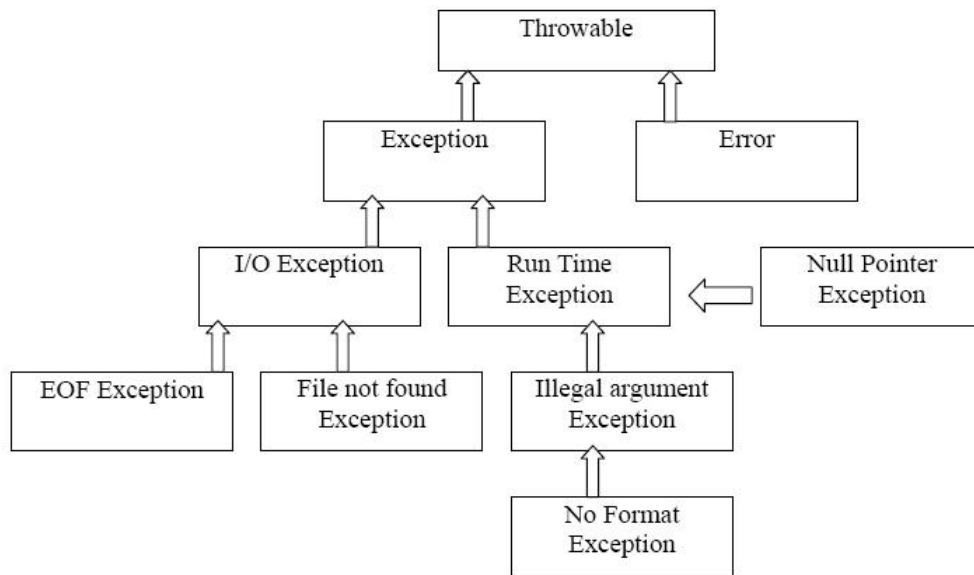


Figure 1: Class structure

Exceptions can be of two types.

- Checked Exception: Program or library can throw such types of exceptions. It requires to be thrown explicitly.
- Unchecked Exception: Runtime time system throws this type of exceptions.

A method throwing exception can be of the following format.

```
public / private method(..) throws EOFException, ...{  
.  
.
```

```
.  
. }  
}
```

This is an example of checked exception. Here `throws` is a unary operator. It requires object of type `throwable`
e.g.

```
f1(...) throws e1{  
:  
}  
F2(...){  
    F1(...);    //Give error  
    Try{        //valid  
        F1(...);  
    }  
    Catch(e1 e){  
    }  
}
```

If there is an exception, being caught by a function then any other function calling previous function must throw the exception or catch the exception using `try-catch`.

```
F2(...) throws e1{  
    F1();  
}
```

Files

Up to now we used to do input/output from standard input/standard output. Java provides us notion of streams for I/O. Primitive classes are

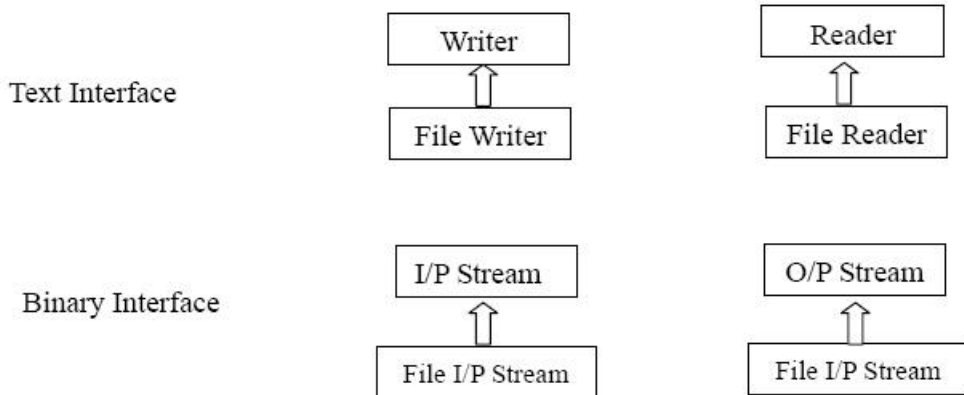


Figure 2: class structure