Java exceptions

The following picture illustrates the simplified hierarchy of exceprions:

RuntimeException

Error

Exception

Throwable

Base class for all Exceptions is **java.lang.Throwable.** This class provides a set of common methods for all exceptions: String getMessage(), Throwable getCause(), printStackTrace.

* Subclasses of the Error class represents low-level exceptions in JVM: OutOfMemory, StackOverflow
* Subclasses Exception class deal with exceptional events inside application: RunTimeException,

IOException

* The RuntimeException is special subclass for represent unchecked exception, including:

ArithmeticException, NumberFormatException, NullPointerException.

**Checked exception** are represented by the Exception class, excluding RuntimeException

If a method throws a checked exception, this must be **marked in the declaration using special throws keyword.**

public static String readLineFromFile() throws FileNotFoundException {

Scanner scanner = new Scanner(new File(“file.txt”)); // java.io.FileNotFoundException

return scanner.nextLine();

}

**Unchecked Exceptions** are represented by the RuntimeException class and all its subclasses.

The compiler does not check whether the programmer expects their occurrence.

public static Long converString2Long(String str) {

return Long.parseLong(str);

}

Note, runtime exceptions may occur anywhere in a program. Adding them to each method's declaration would reduce the clarity of a program. Thus, the compiler doesn't require that you specify runtime exceptions in declarations