Java Fundamentals

Section 3 - Exceptions

Topics

- What are exceptions?
- Types of exceptions
- Exception handling

What are exceptions?

What are exceptions?

- Events that occur during an execution of a program that signal an abnormal state, a problem.
  - Eg: a application that tries to write in a file, but the file is not found
  - Eg: a division by 0 in a calculator application
- In Java, the exceptions are also objects; they contain relevant information about the abnormal sytuation that has occurred:
  - Exception type (the class type)
  - **Stack trace** (list of calls, showing where exactly did the error occur)
- We can throw new exceptions - keyword **throw**

Type of exceptions

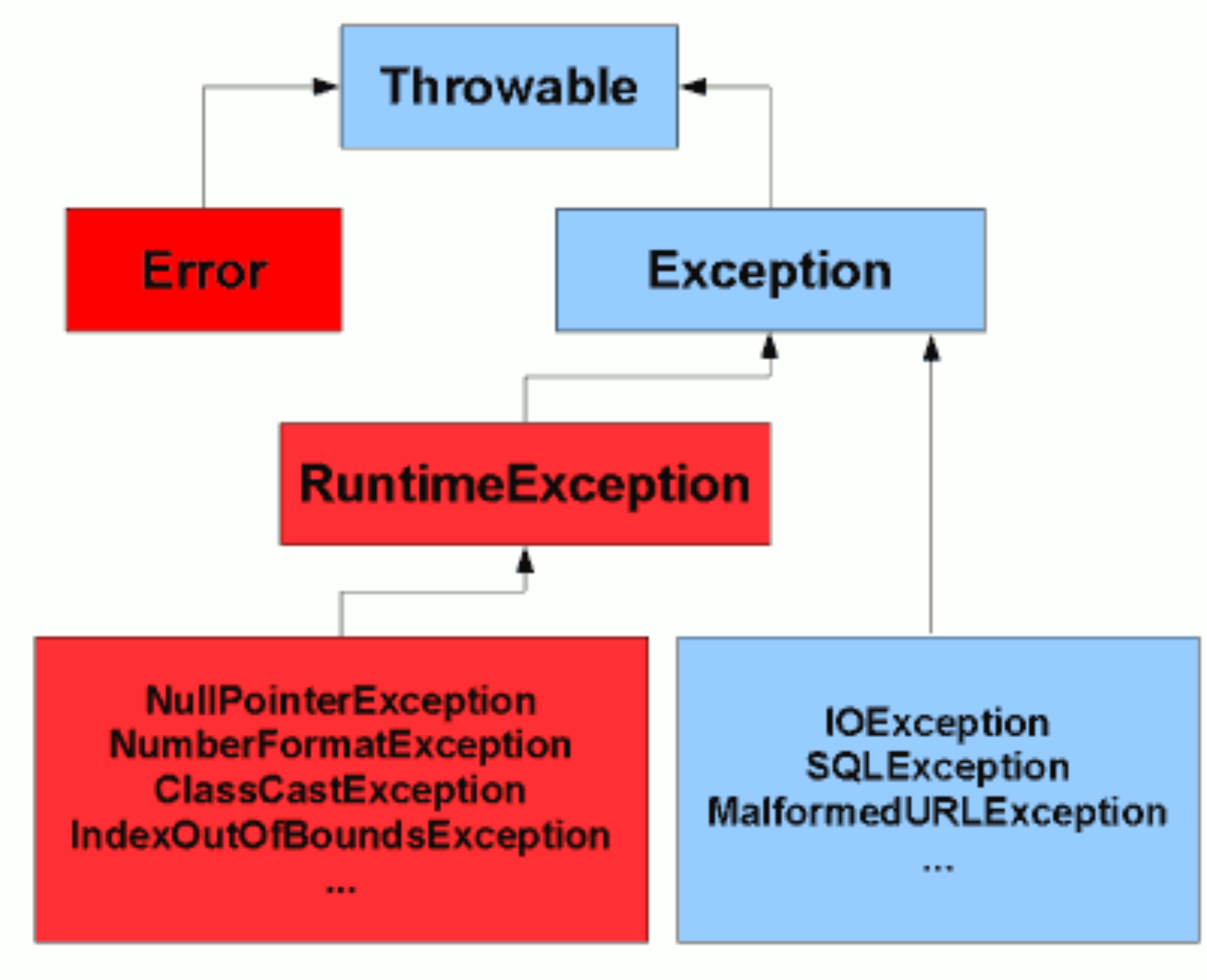
Types of exceptions

- **Checked exceptions** - exception that can be anticipated, should be cached/handled (e.g.: I/O Exceptions)

- **Unchecked exceptions** – don’t need to be handled, usually they mean there is a bug in the code that needs to be fixed (e.g.: NullPointerException)

- **Errors** – serious system issues that usually cannot be handled (e.g.: OutOfMemoryError)

- We can also define custom exceptions by extending an exception class (eg: Exception, RuntimeException)



Exception handling

Exception handling

- We can handle an exception by:
  - Throwing it so it can be handled by the caller method (keyword **throws**)
  - Using a **try – catch** block
    - we can have multiple catch blocks; in this case the exceptions should be from the most specific to the most generic
    - additionally, we can use a **finally** block (code that will always execute)