Types of Errors

- Compile-Time Errors
  - Detected by compiler
  - Detected prior to running/executing your program
  - Syntax Errors

- Run-Time Errors
  - Not detected by the compiler
  - Occurs at run time (when executing your program)
  - Potentially could cause your program stop working or crash
  - Exceptions are run-time errors

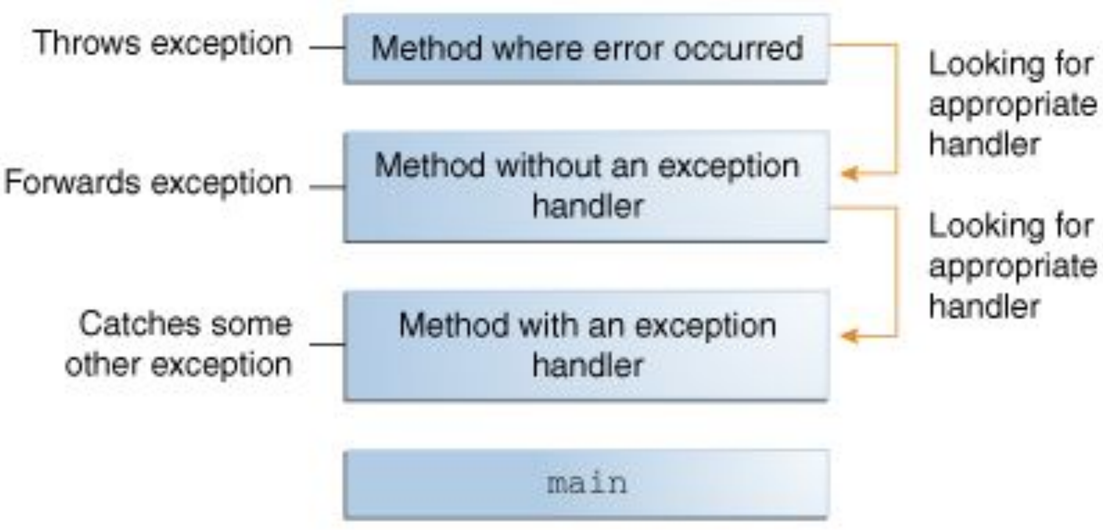
Exception

- Errors will happen in your program no matter how good of a programmer you are
  - User input errors, Device errors, Physical limitations, Coding errors
- But what really matters is
  - What happens after the error?
  - How is the error handled?
  - Who handles the error?
  - Can the program recover or should it just die gracefully?
- The Java language uses exceptions to provide error-handling capabilities for its programs

Throwing Exception

When an error occurs within a method, the method creates an object and hands it off to the runtime system. The object, called an *exception object*, contains information about the error, including its type and the state of the program when the error occurred. Creating an exception object and handing it to the runtime system is called *throwing an exception*.

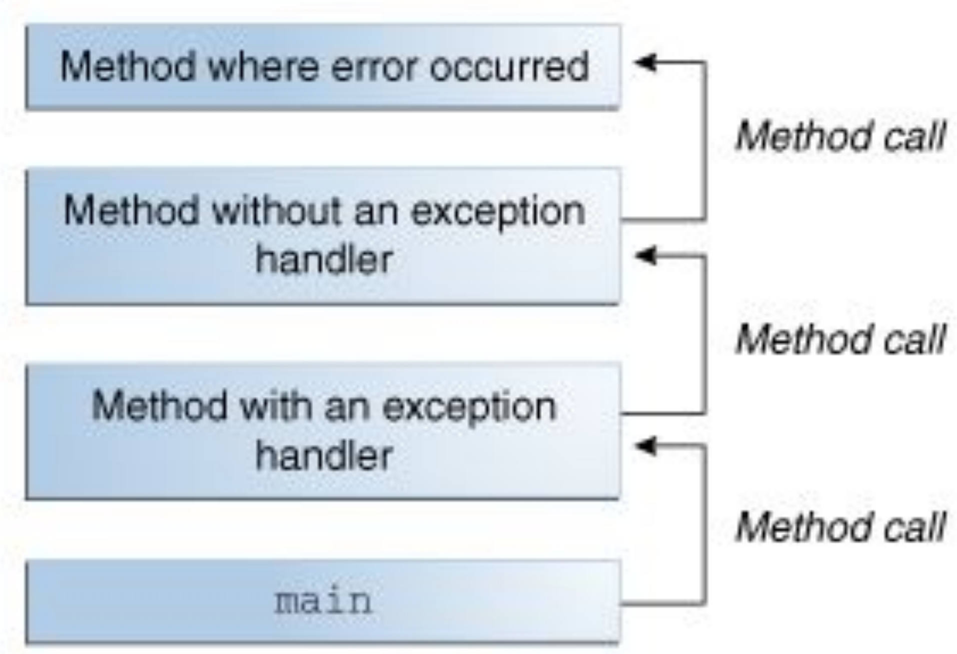
After a method throws an exception, the runtime system attempts to find something to handle it. The set of possible "somethings" to handle the exception is the ordered list of methods that had been called to get to the method where the error occurred


Throws exception
Method where error occurred
Looking for appropriate handler
Forwards exception
Method without an exception handler
Looking for appropriate handler
Catches some other exception
Method with an exception handler
main

Searching the call stack for the exception handler.

Call Stack

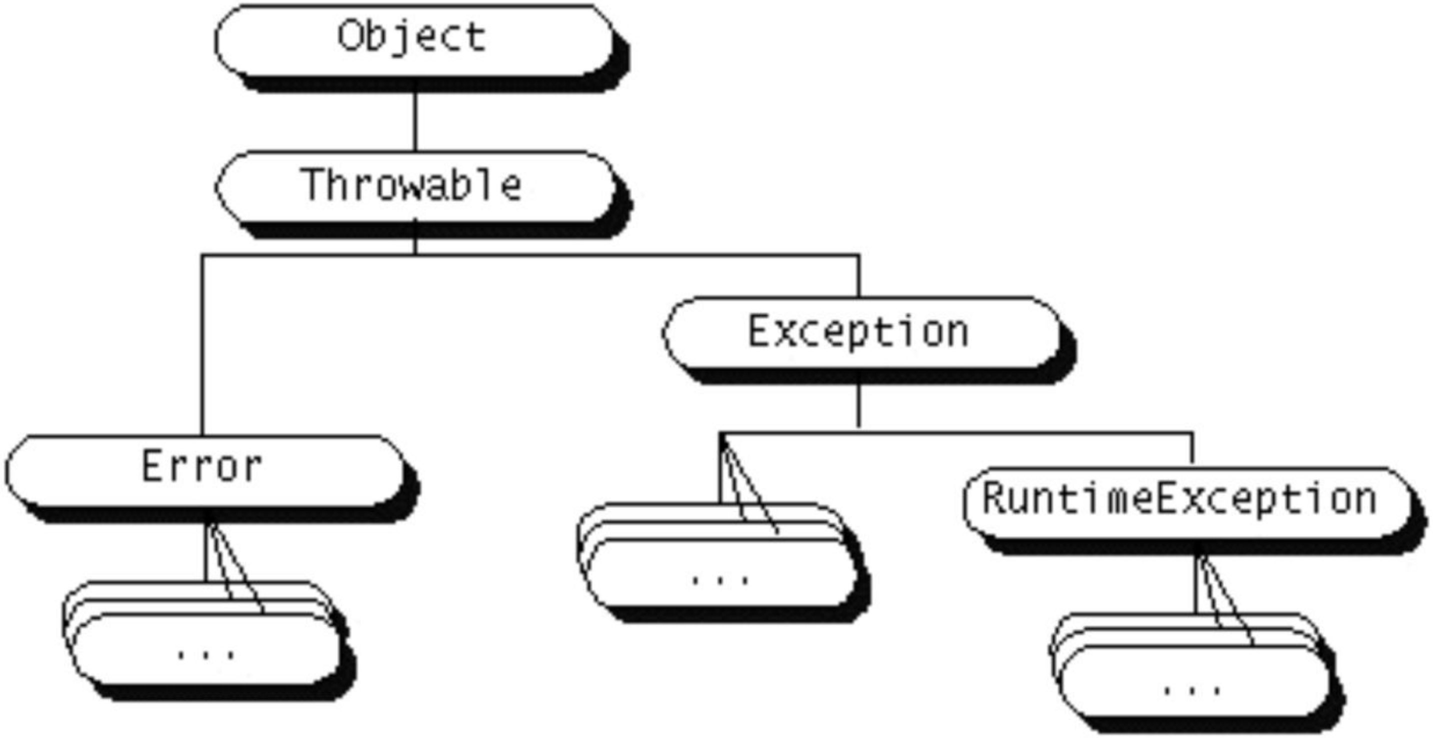
- The list of methods is known as the *call stack*


Method where error occurred
Method without an exception handler
Method with an exception handler
main
Method call
Method call
Method call

The call stack.

Throwable Objects

- Anything that is a descendant of the class **Throwable** can be **thrown**



Throwable Objects

- Errors
  - Typically a hard failure in the Java Virtual Machine
  - Java programs should not try and catch these errors
- Exceptions
  - Most programs throw and catch objects that derive from the Exception class
  - Exceptions indicate that a problem occurred but that the problem is not a serious systemic problem
  - If specified, must be caught somewhere in the calling hierarchy (except for RuntimeExceptions)
- Runtime Exceptions
  - Special subclass of Exception class
  - Exceptions that occur within the Java virtual machine (during runtime)
  - The compiler allows runtime exceptions to go uncaught and unspecified

Runtime Exception

- A subclass of **`Exception`** that does not have to be caught
- Subclasses of **`RuntimeException`** defined in the standard package java.lang:
  - Arithmetic Exception
  - IndexOutofBoundsException
  - NegativeArraySizeException
  - NullPointerException
  - etc