Understanding Exception Types

Jim Wilson
MOBILE SOLUTIONS DEVELOPER & ARCHITECT

@hedgehogjim   jwhh.com

Overview

Exceptions as classes

Handling exceptions by type

Checked vs. unchecked exceptions

Exceptions and methods

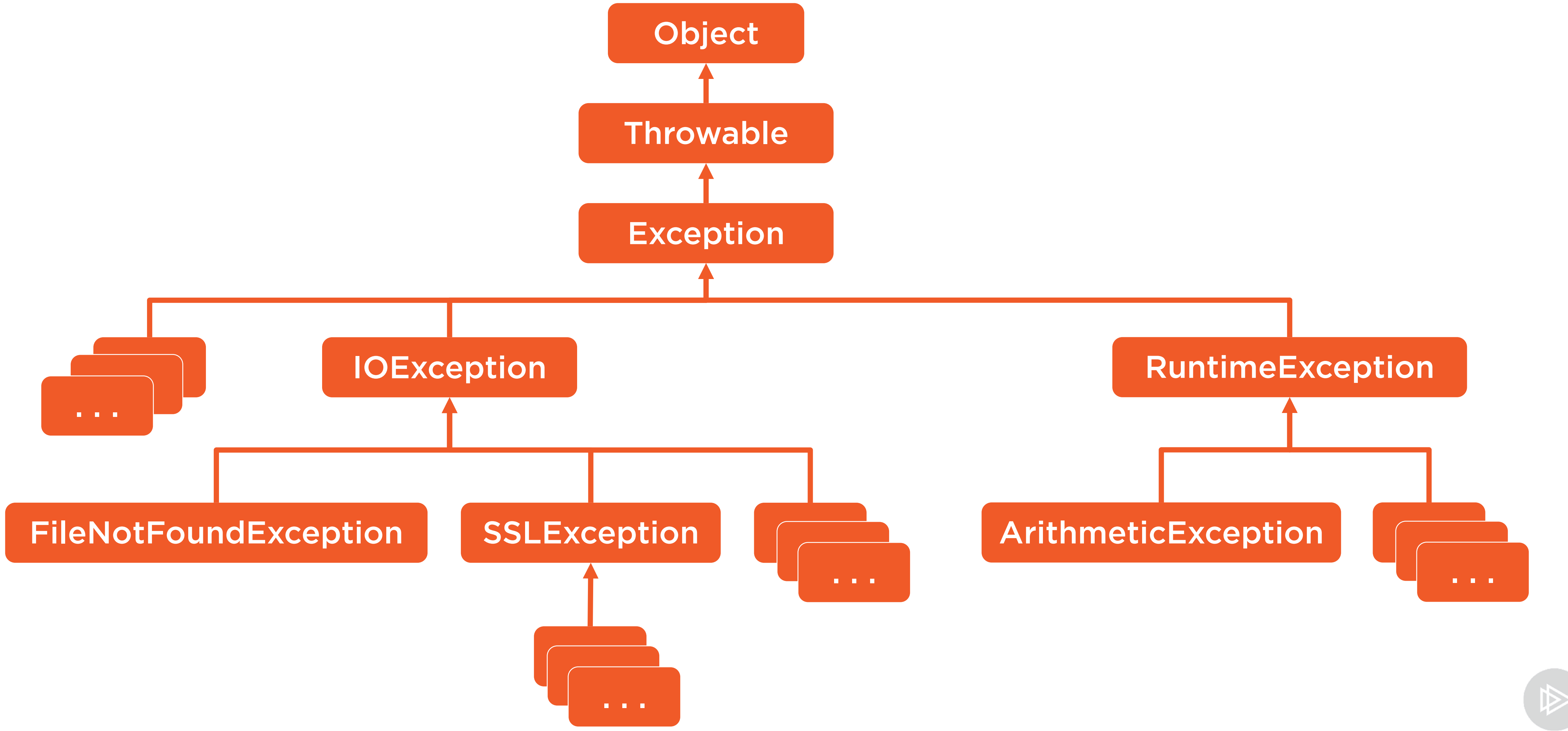
Exceptions Are Represented by Classes

All Inherit from Exception class

Some classes represent broad category of exceptions

Some classes represent more specific exceptions

Exception Class Hierarchy



Exceptions Can Be Handled by Type

A try can have multiple catches associated with it

Tested in order from top-to-bottom

First assignable catch is selected

Place more specific exceptions before less specific exceptions

Main.java

```java
int i = 12

int j = 2;

try {

    int result = i / (j - 2);

    System.out.println(result);

} catch (Exception ex) {

    System.out.println("Error: " + ex.getMessage());

} catch (ArithmeticException ex) {

    System.out.println(“Invalid math operation - " + ex.getMessage());

}
```

Main.java

```java
int i = 12

int j = 2;

try {

    int result = i / (j - 2);

    System.out.println(result);

} catch (ArithmeticException ex) {

    System.out.println(“Invalid math operation - " + ex.getMessage());

} catch (Exception ex) {

    System.out.println("Error: " + ex.getMessage());

}
```

Exceptions Fall into Two Broad Categories

In both cases, your program will crash if an exception gets thrown but is not caught

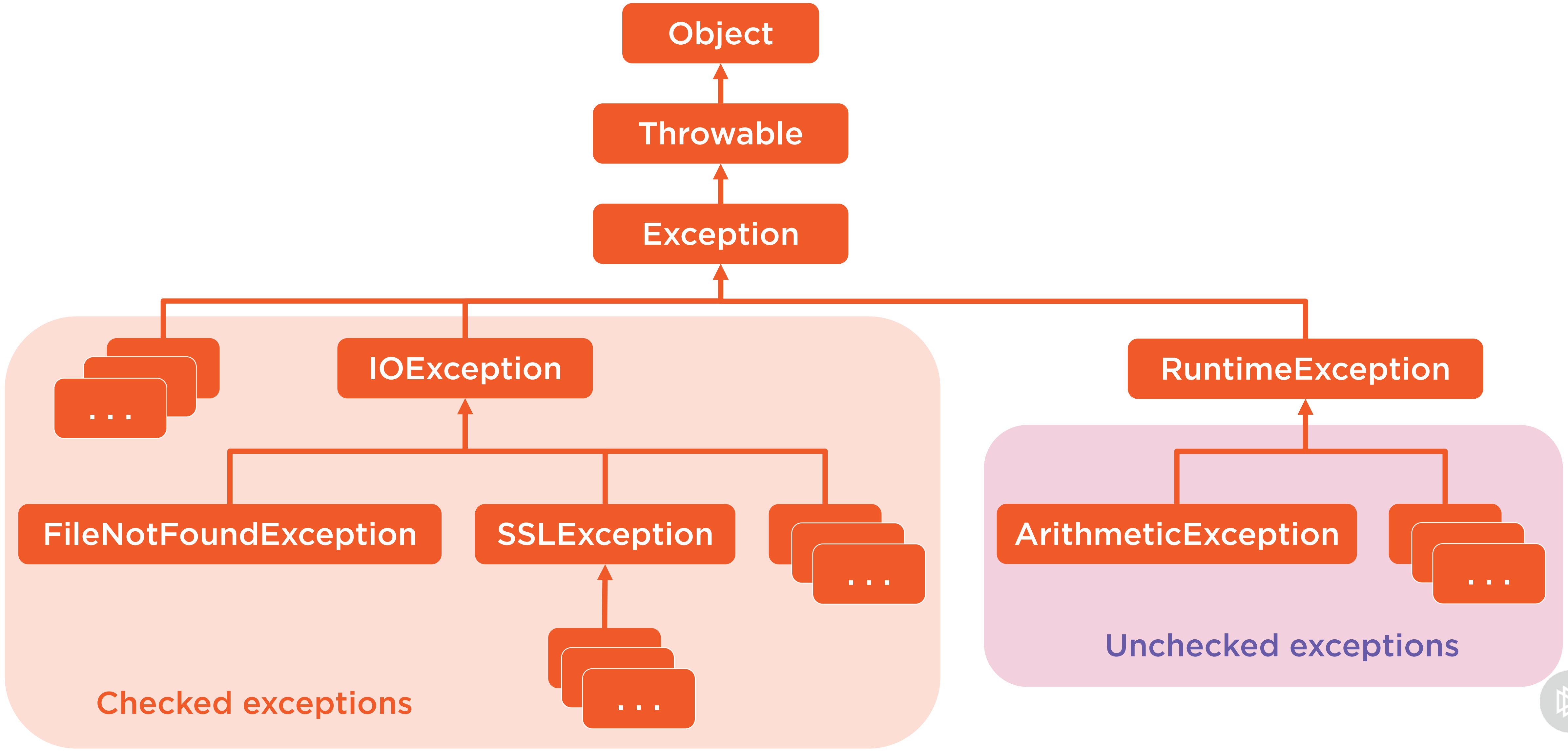
Checked exceptions

Compiler raises an error if not handled

Unchecked exceptions

Compiler does not enforce handling

Exception Class Hierarchy



Exceptions and Methods

Exceptions can cross method boundaries

If not handled, will propagate up the call stack

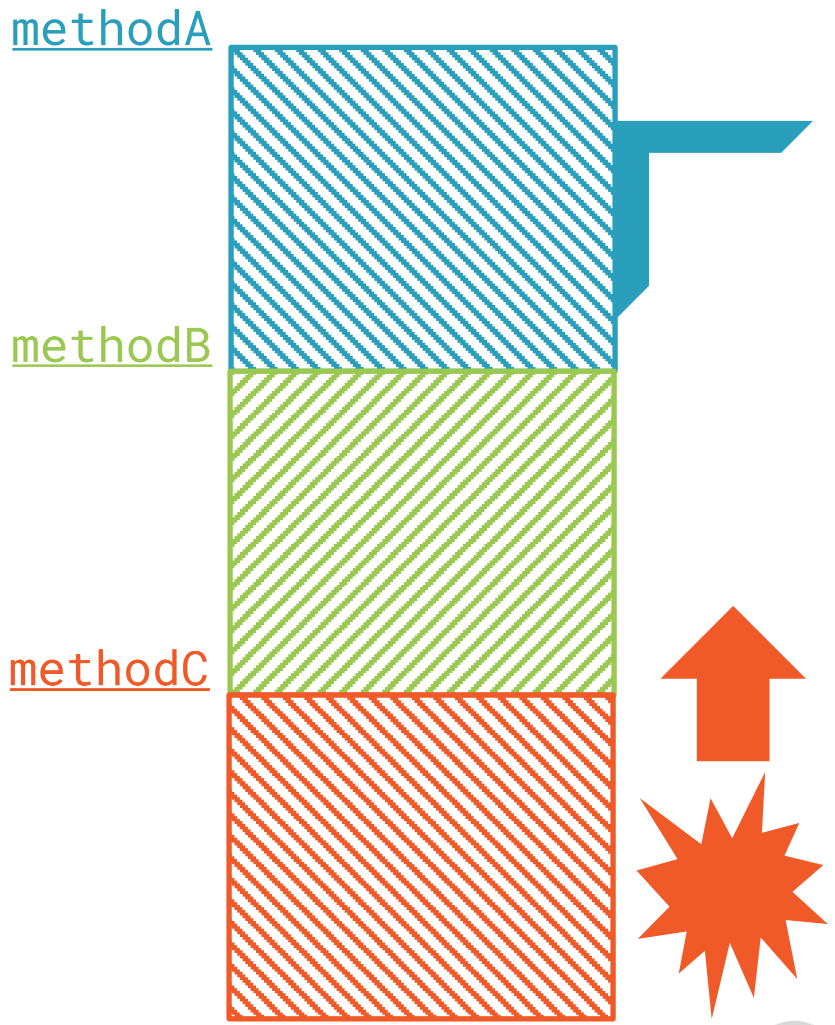
An exception thrown within a method can be caught by the code that called the method

Exceptions and Methods

```
void methodA() {
  try {
    methodB();
  } catch (. . .) {
    . . .
  }
}
```

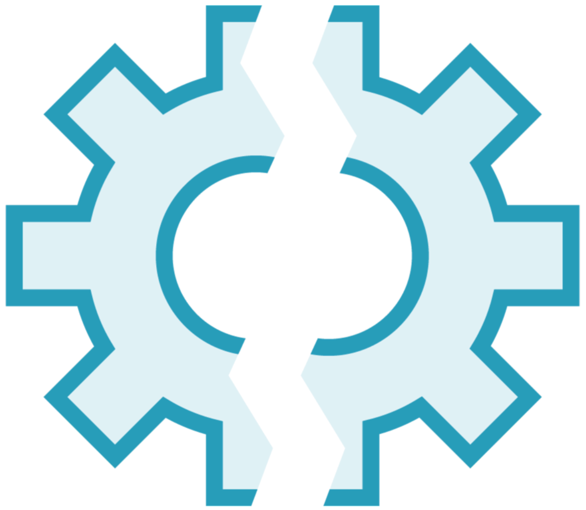
```
void methodB() {
  . . .
  methodC();
}
```

```
void methodC() {
  // Does something
  // that throws an
  // exception
}
```

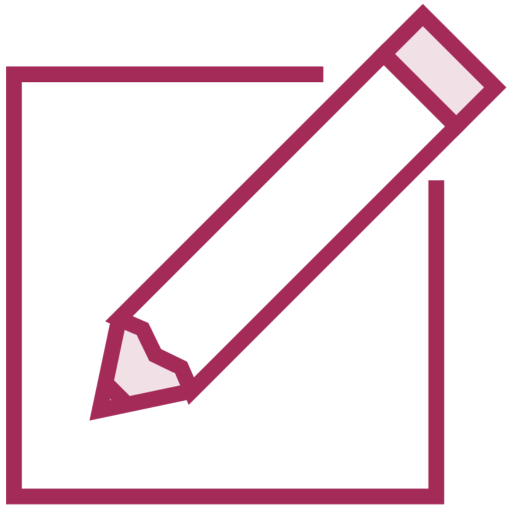


Exceptions Are Part of a Method's Contract

A method must deal with any checked exceptions



Catch the exception



Document that exception might occur

Use the throws clause

Summary

Exceptions are represented by classes

- All inherit from the Exception class
- Some represent broad set of errors
- Some represent very specific errors

Summary

Exceptions can be handled by type

- A try can have multiple catches
- Tested in order from top-to bottom
- A catch will handle the exception type or a type that inherits from that type

Summary

Checked exceptions

- Compiler raises an error if not handled

Unchecked exceptions

- Compiler does not enforce handling
- Will crash your program if thrown and not handled

Summary

Exceptions can cross method boundaries

- Can handle an exception thrown by a method your code calls

Exceptions part of a method's contract

- Can catch exception
- Can document exception with throws