

# Understanding Exception Types

---



**Andrejs Doronins**

# Overview



**Explore the exceptions hierarchy**

**Identify two broad categories of exceptions**

- **Checked and unchecked**
- **Enumerate the most frequent ones**
- **Review their relationships**

**Revisit chained catch blocks**

**Errors: Do's and Don'ts**

In Java, (almost) everything is  
an object.  
This includes exceptions.

```
class SomeException {  
    public SomeException() {  
    }  
}
```



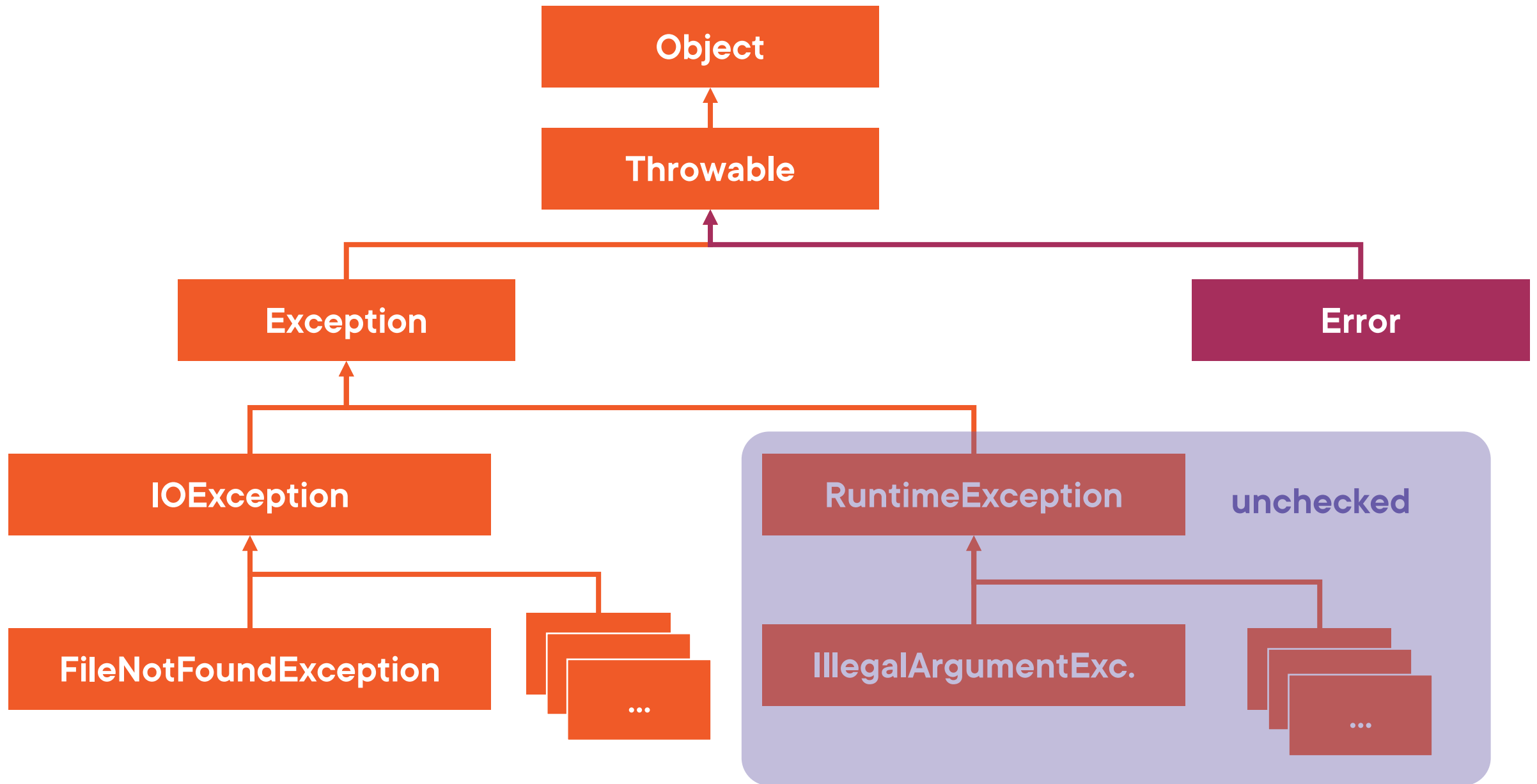
```
new SomeException();
```

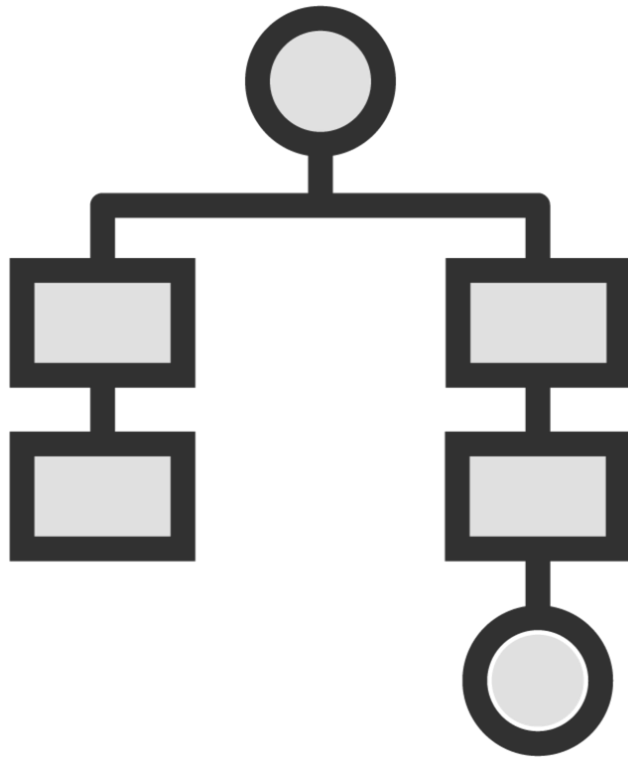


```
throw RuntimeException();
```



```
throw new RuntimeException();
```





### Purpose of Throwable:

- Bring Exceptions and Errors into a single hierarchy
- Provide useful methods e.g. `printStackTrace()`;

# Exception Types



**Checked**

(by the compiler)



**Unchecked**

a.k.a. Runtime Exceptions



Both checked and unchecked exceptions will crash the program if left unhandled.

Common RuntimeException Classes
ArithmeticException
ArrayIndexOutOfBoundsException
ClassCastException
IllegalArgumentException
IllegalStateException
NullPointerException
NumberFormatException

# ArithmeticException

```
int result = 5 / 0;
```

Exception in thread "main" java.lang.ArithmeticException: / by zero

# ArrayIndexOutOfBoundsException

```
int[] arr = new int[1];  
System.out.println(arr[1]); // off by one
```

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException:  
Index 1 out of bounds for length 1
```

# ClassCastException

```
String s = "1";  
Integer n = (Integer) s;
```

N/A – won't compile

# ClassCastException 2

```
String s = "1";  
Object obj = s;  
Integer n = (Integer) obj;
```

Exception in thread "main" java.lang.ClassCastException: class java.lang.String cannot be cast to class java.lang.Integer

# IllegalArgumentException

```
int age = 15;  
if (age < 21) {  
    throw new IllegalArgumentException("message");  
}
```

Exception in thread "main" java.lang.IllegalArgumentException: message

# IllegalStateException

```
if(!isSwitchedOn){  
    throw new IllegalStateException("XYZ must be switched on");  
}
```

Exception in thread "main" java.lang.IllegalStateException: XYZ must be switched on



# NullPointerException

```
String s1 = null;  
s1.toUpperCase();
```

Exception in thread "main" java.lang.NullPointerException

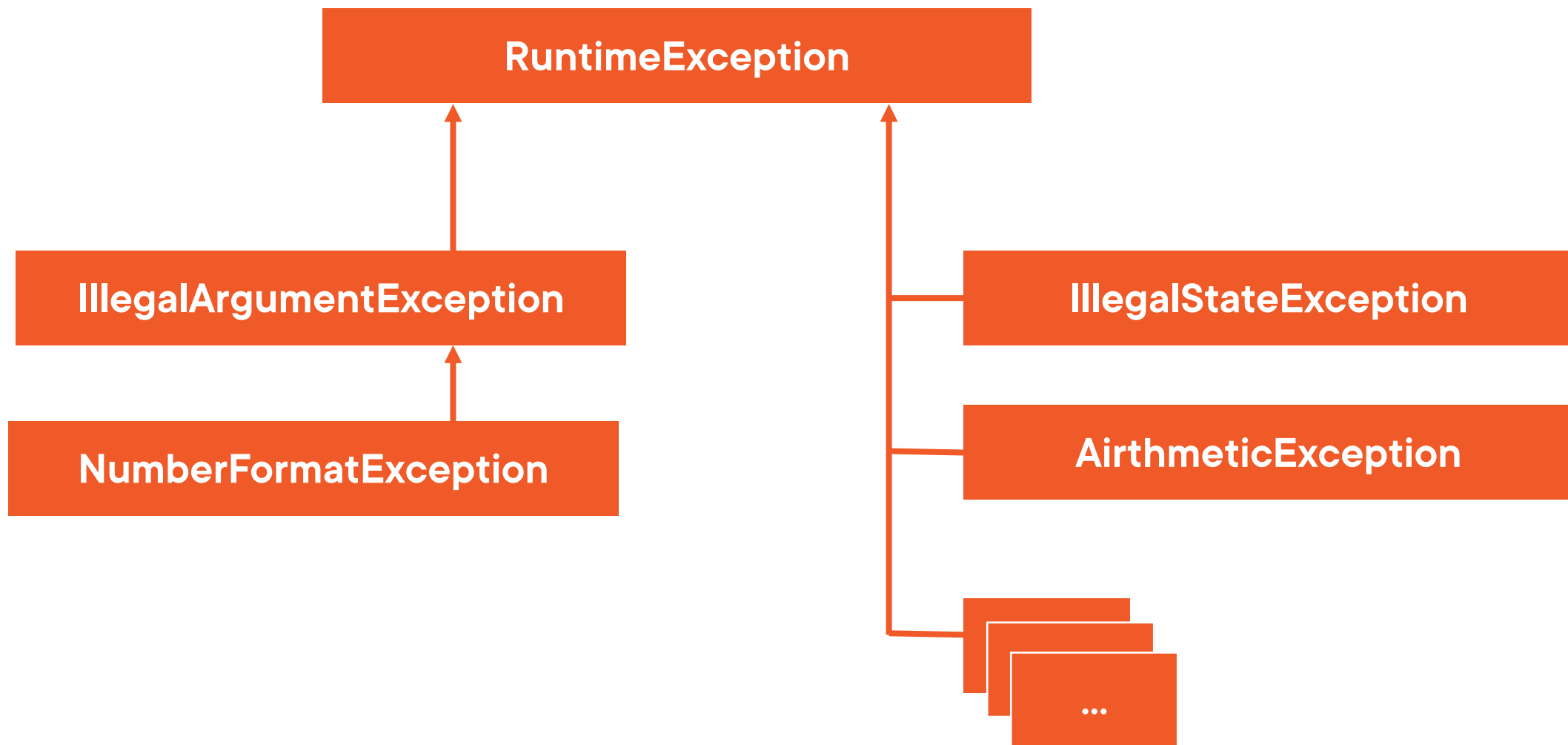
# NumberFormatException

```
Integer.parseInt("a");
```

```
Double.parseDouble("b");
```

```
Exception in thread "main" java.lang.NumberFormatException: For input  
string: "a"
```

```
Exception in thread "main" java.lang.NumberFormatException: For input  
string: "b"
```



## Common Checked Exception Classes

**IOException**

**FileNotFoundException**

# FileNotFoundException

```
try {  
    var br = new BufferedReader(new FileReader("file.txt"));  
} catch (FileNotFoundException e) {  
    e.printStackTrace();  
}
```

java.io.FileNotFoundException: file.txt (The system cannot find the file specified)

```
try {  
    throw new IOException();  
}  
catch (IOException e) { }  
catch (FileNotFoundException e) { }
```

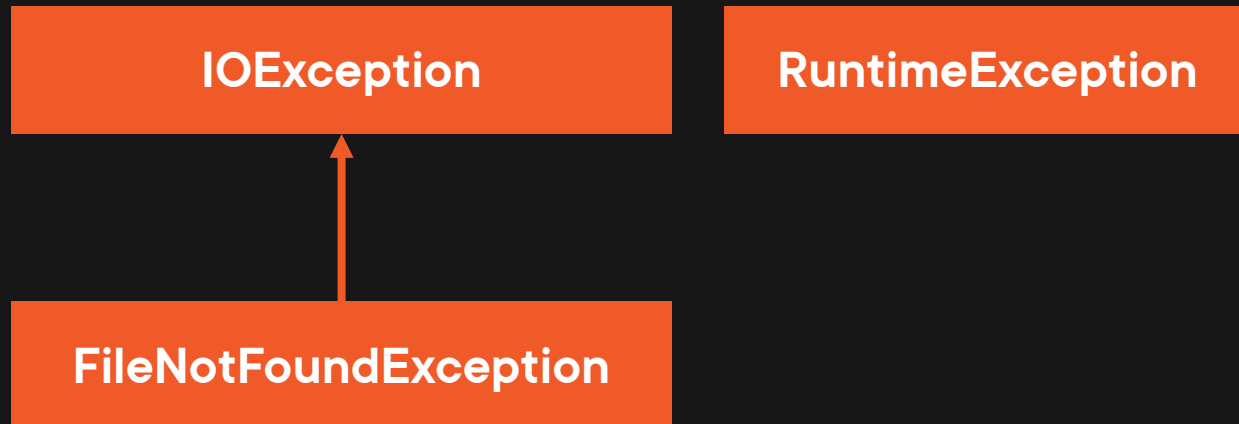


IOException



FileNotFoundException

```
try {  
    throw new IOException();  
}  
catch (FileNotFoundException e) { }  
catch (RuntimeException e)      { }  
catch (IOException e)           { }
```



```
try {
```

```
}
```

```
catch (NumberFormatException e) { }
```

```
catch (IllegalArgumentException e) { }
```

```
catch (IllegalStateException e) { }
```



**IllegalArgumentException.**

**IllegalStateException.**

**NumberFormatException**





Common Errors	
ExceptionInInitializerError	
StackOverflowError	
NoClassDefFoundError	
OutOfMemoryError	



**Don't catch, handle or try  
to recover from errors**



## Compile Time

ClassA

ClassB

ClassC

## Runtime

ClassA.jar

ClassB.jar



NoClassDefFoundError

# Summary



**All exceptions extend Throwable**

- **Exceptions vs. Errors**

**Exception types:**

- **Checked vs. Unchecked**

**Review of specific exceptions and errors**

**Practice!**

# Throwing Exceptions

---