Understanding the Exception Hierarchy

Esteban Herrera
JAVA ARCHITECT

@eh3rrera http://eherrera.net

Overview

The Exception hierarchy

Exception

RuntimeException

Error

Common methods

The Exception Hierarchy

Exceptions are objects

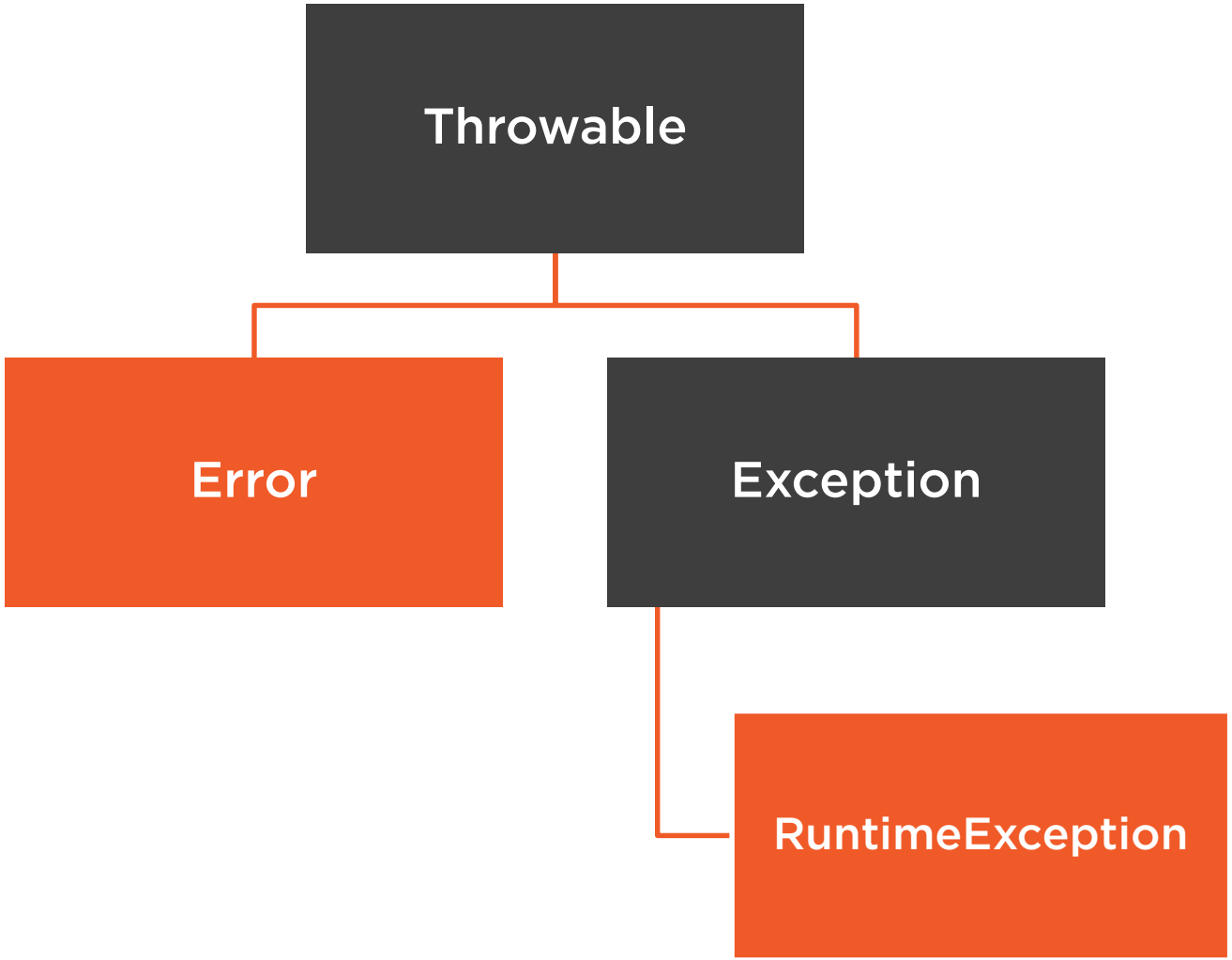
Like everything in Java

Inheritance is used to categorize different types of exceptions

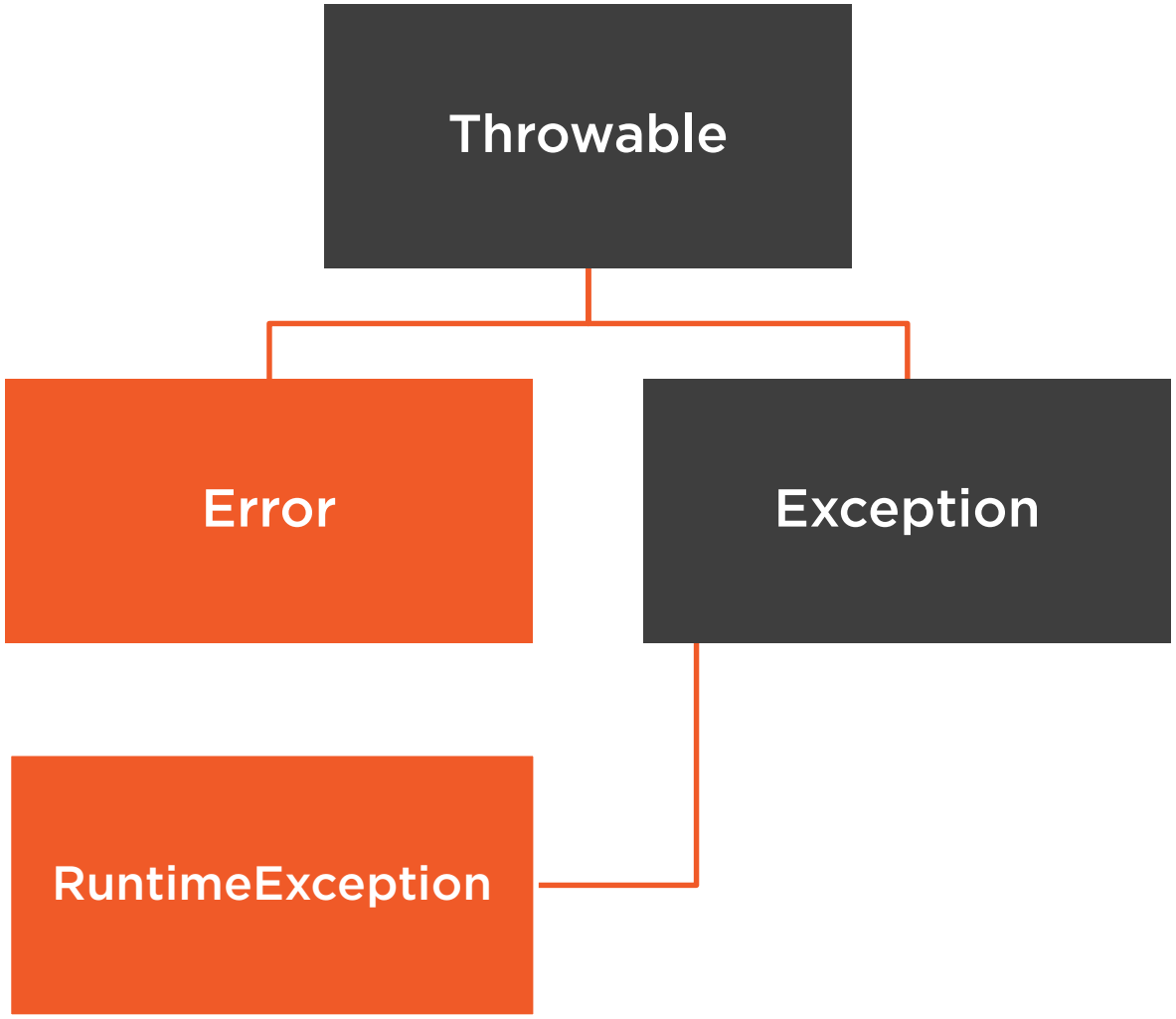
Three Most Important Exception Classes

The Parent of All Exceptions

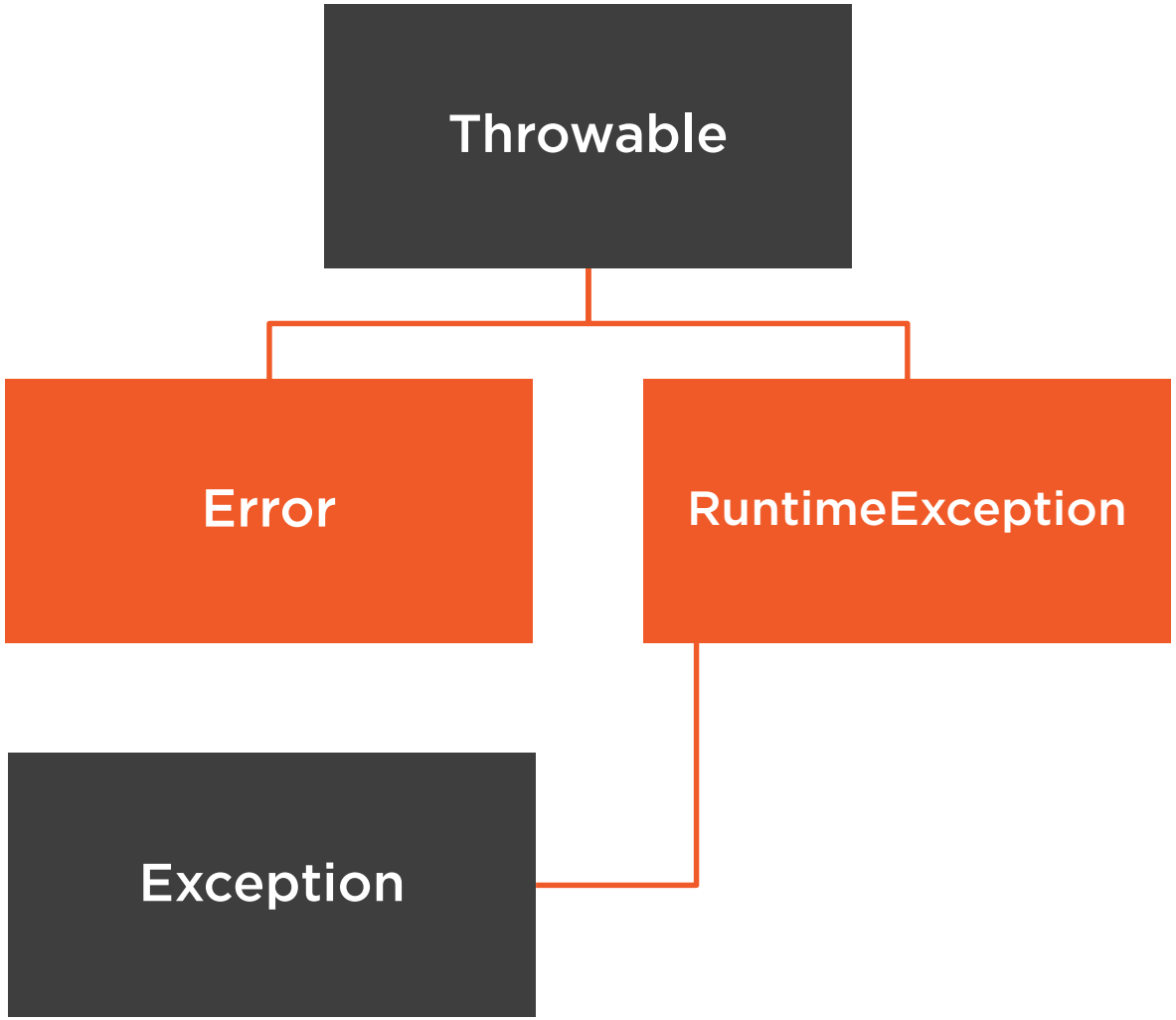
The Exception Hierarchy



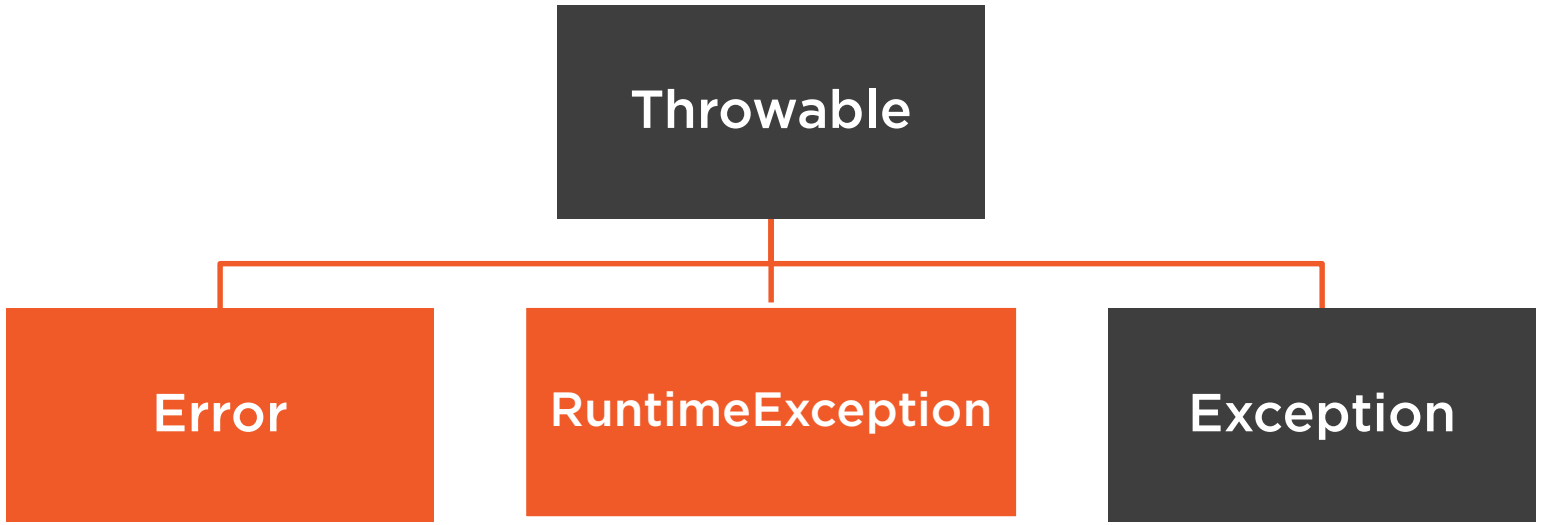
The Exception Hierarchy



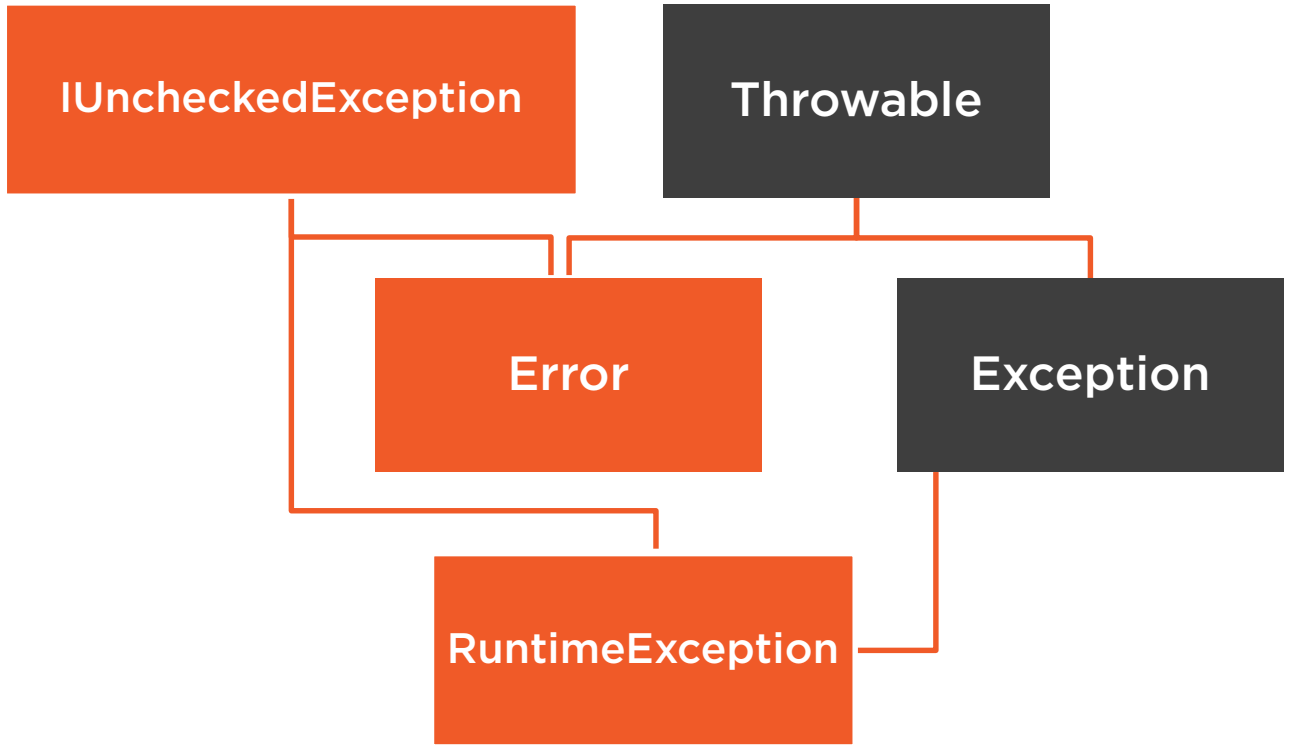
Another Exception Hierarchy?



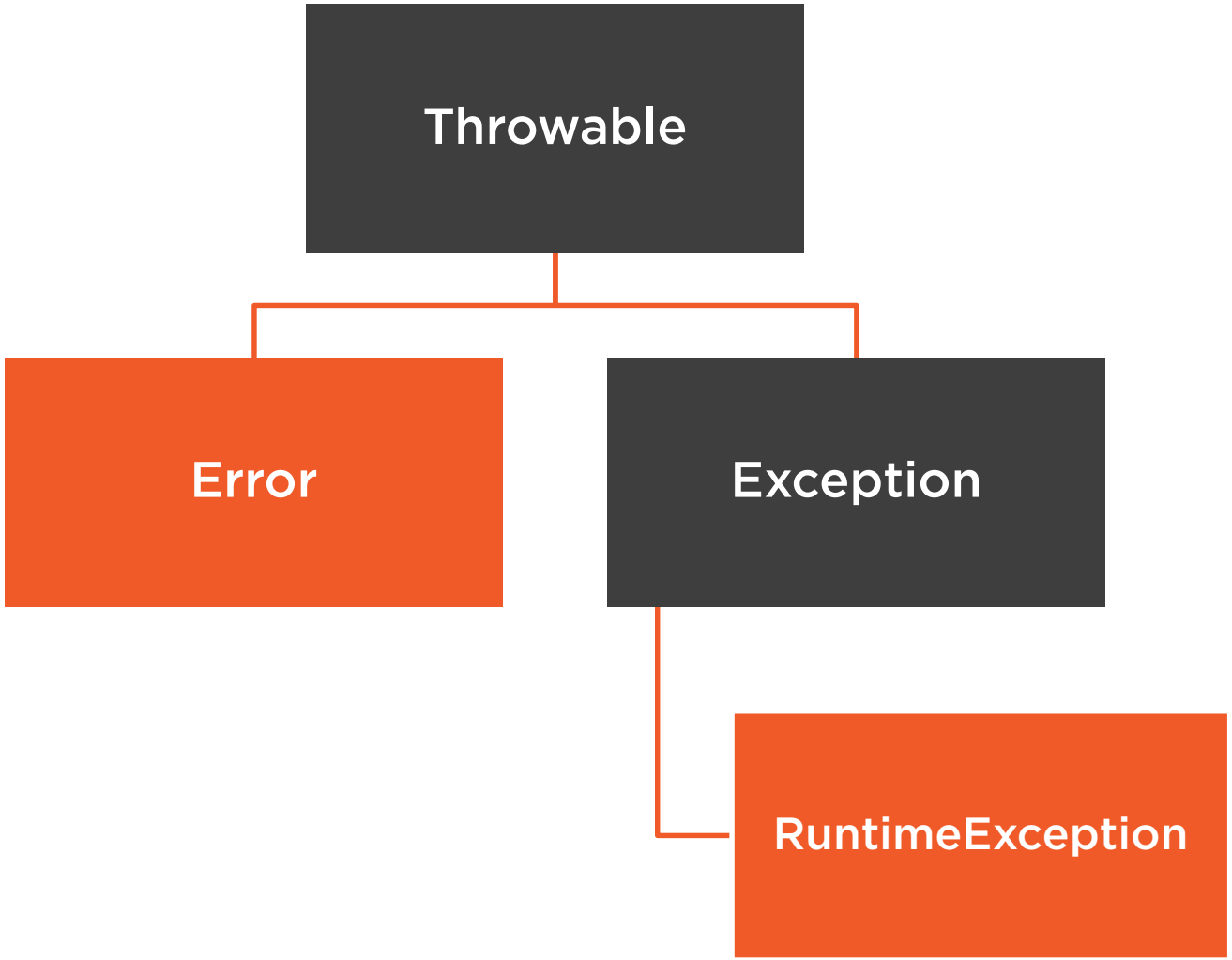
Another Exception Hierarchy?



Another Exception Hierarchy?



The Exception Hierarchy



Exception

```
try {

    if (error)

        throw Exception

} catch (Exception) {

}
```

- Code that can raise an exception

- Throw
  (Create exception and transfer control)

- Catch
  (where execution is transferred and exception handled)

compact1, compact2, compact3

java.lang

Class Exception

java.lang.Object
    java.lang.Throwable
        java.lang.Exception

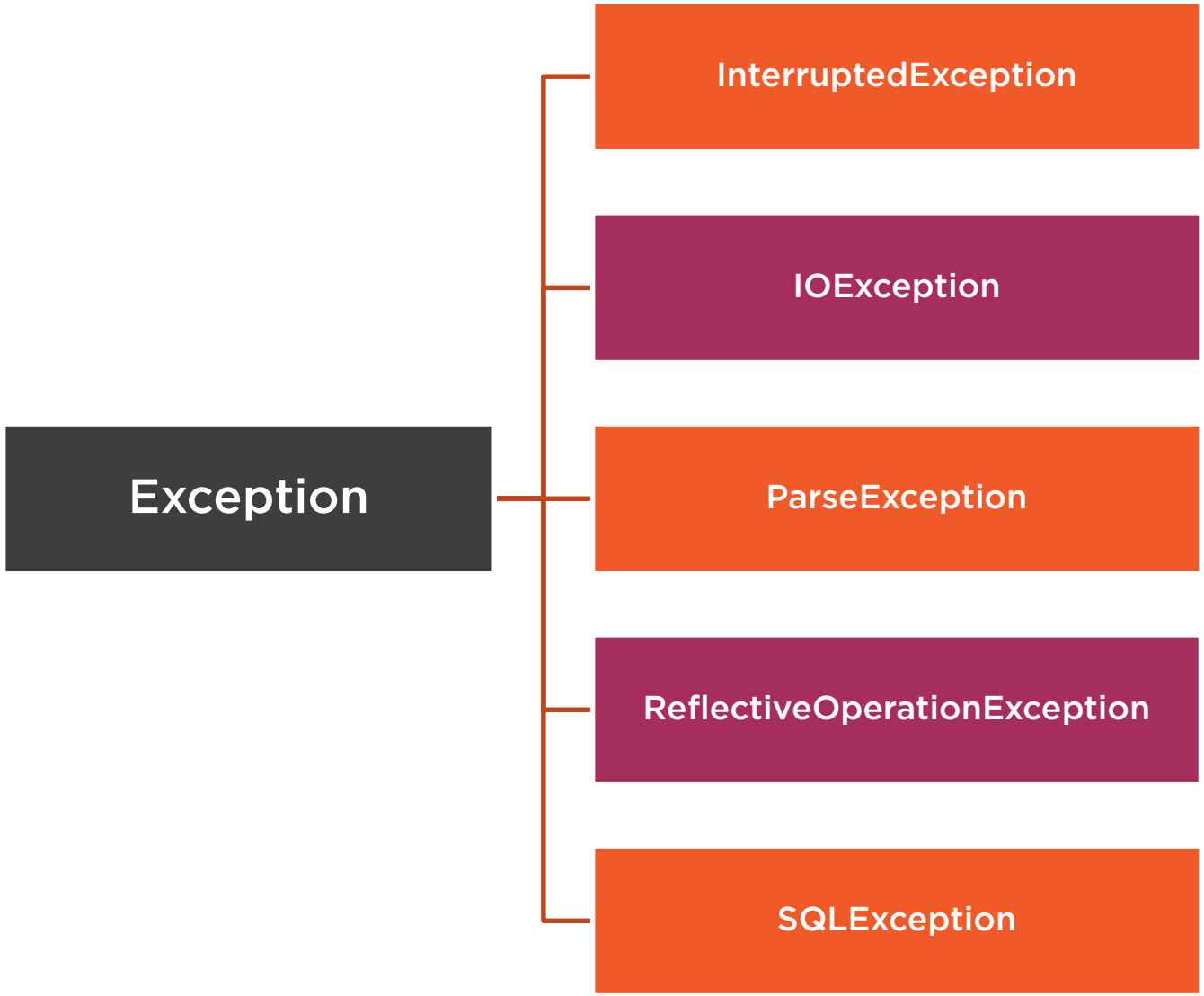
All Implemented Interfaces:

Serializable

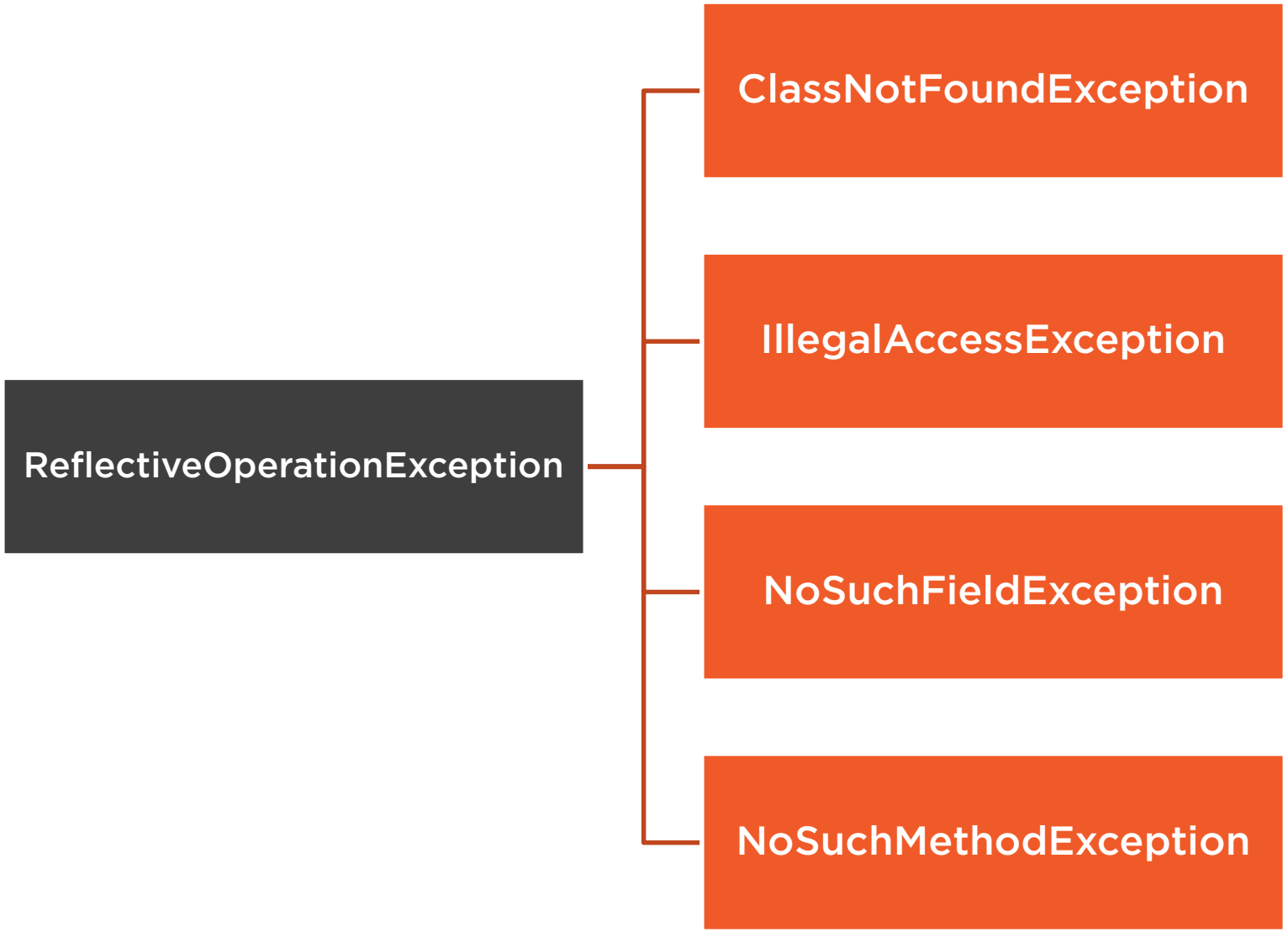
Direct Known Subclasses:

AclNotFoundException, ActivationException, AlreadyBoundException, ApplicationException, AWTException, BackingStoreException, BadAttributeValueExpException, BadBinaryOpValueExpException, BadLocationException, BadStringOperationException, BrokenBarrierException, CertificateException, CloneNotSupportedException, DataFormatException, DatatypeConfigurationException, DestroyFailedException, ExecutionException, ExpandVetoException, FontFormatException, GeneralSecurityException, GSSException, IllegalClassFormatException, InterruptedException, IntrospectionException, InvalidApplicationException, InvalidMidiDataException, InvalidPreferencesFormatException, InvalidTargetObjectTypeException, IOException, JAXBException, JMException, KeySelectorException, LambdaConversionException, LastOwnerException, LineUnavailableException, MarshalException, MidiUnavailableException, MimeTypeParseException, MimeTypeParseException, NamingException, NoninvertibleTransformException, NotBoundException, NotOwnerException, ParseException, ParserConfigurationException, PrinterException, PrintException, PrivilegedActionException, PropertyVetoException, ReflectiveOperationException, RefreshFailedException, RemarshalException, RuntimeException, SAXException, ScriptException, ServerNotActiveException, SOAPException, SQLException, TimeoutException, TooManyListenersException, TransformerException, TransformException, UnmodifiableClassException, UnsupportedAudioFileException, UnsupportedCallbackException, UnsupportedFlavorException, UnsupportedLookAndFeelException, URIReferenceException, URISyntaxException, UserException, XAException, XMLParseException, XMLSignatureException, XMLStreamException, XPathException

Common Exception Classes



Checked Exceptions?



```
Class.forName(String) throws ClassNotFoundException

ClassLoader.findSystemClass(String) throws ClassNotFoundException

ClassLoader.loadClass(String) throws ClassNotFoundException
```

ClassNotFoundException

Thrown when an application tries to load in a class through its name, but no definition for the class with the specified name could be found.

compact1, compact2, compact3

java.io

Class IOException

java.lang.Object
    java.lang.Throwable
        java.lang.Exception
            java.io.IOException

All Implemented Interfaces:

Serializable

Direct Known Subclasses:

ChangedCharSetException, CharacterCodingException, CharConversionException, ClosedChannelException, EOFException, FileLockInterruptionException, FileNotFoundException, FilerException, FileSystemException, HttpRetryException, IIOException, InterruptedByTimeoutException, InterruptedIOException, InvalidPropertiesFormatException, JMXProviderException, JMXServerErrorException, MalformedURLException, ObjectStreamException, ProtocolException, RemoteException, SaslException, SocketException, SSLException, SyncFailedException, UnknownHostException, UnknownServiceException, UnsupportedDataTypeException, UnsupportedEncodingException, UserPrincipalNotFoundException, UTFDataFormatException, ZipException

```
File file = new File("file.txt");
BufferedReader reader = new BufferedReader(new FileReader(file));
String text = null;

while ((text = reader.readLine()) != null) {
    System.out.println(text);
}
```

```java
try {
    File file = new File("file.txt");
    BufferedReader reader = new BufferedReader(new FileReader(file));
    String text = null;

    while ((text = reader.readLine()) != null) {
        System.out.println(text);
    }
} catch (FileNotFoundException e) {
    // Do something with the exception
}
```

```java
try {
    File file = new File("file.txt");
    BufferedReader reader = new BufferedReader(new FileReader(file));
    String text = null;

    while ((text = reader.readLine()) != null) {
        System.out.println(text);
    }
} catch (FileNotFoundException e) {
    // Do something with the exception
} catch (IOException e) {
    // Do something with the exception
}
```

```java
try {
    File file = new File("file.txt");
    BufferedReader reader = new BufferedReader(new FileReader(file));
    String text = null;


    while ((text = reader.readLine()) != null) {
        System.out.println(text);
    }
} catch (IOException e) {
    // Do something with the exception
}
```

```java
try {
    File file = new File("file.txt");
    BufferedReader reader = new BufferedReader(new FileReader(file));
    String text = null;

    while ((text = reader.readLine()) != null) {
        System.out.println(text);
    }
} catch (Exception e) {
    // Do something with the exception
}
```

RuntimeException

Run-time exception classes are **exempted** [from compile-time checking] because, in the judgment of the designers of the Java programming language, having to declare such exceptions would not aid significantly in establishing the correctness of programs.

The Java® Language Specification 11.2

http://bit.ly/jls11-2

A RuntimeException Represents

An error you cannot anticipate

An error that you should have checked for in your code

compact1, compact2, compact3

java.lang

Class RuntimeException

java.lang.Object
    java.lang.Throwable
        java.lang.Exception
            java.lang.RuntimeException

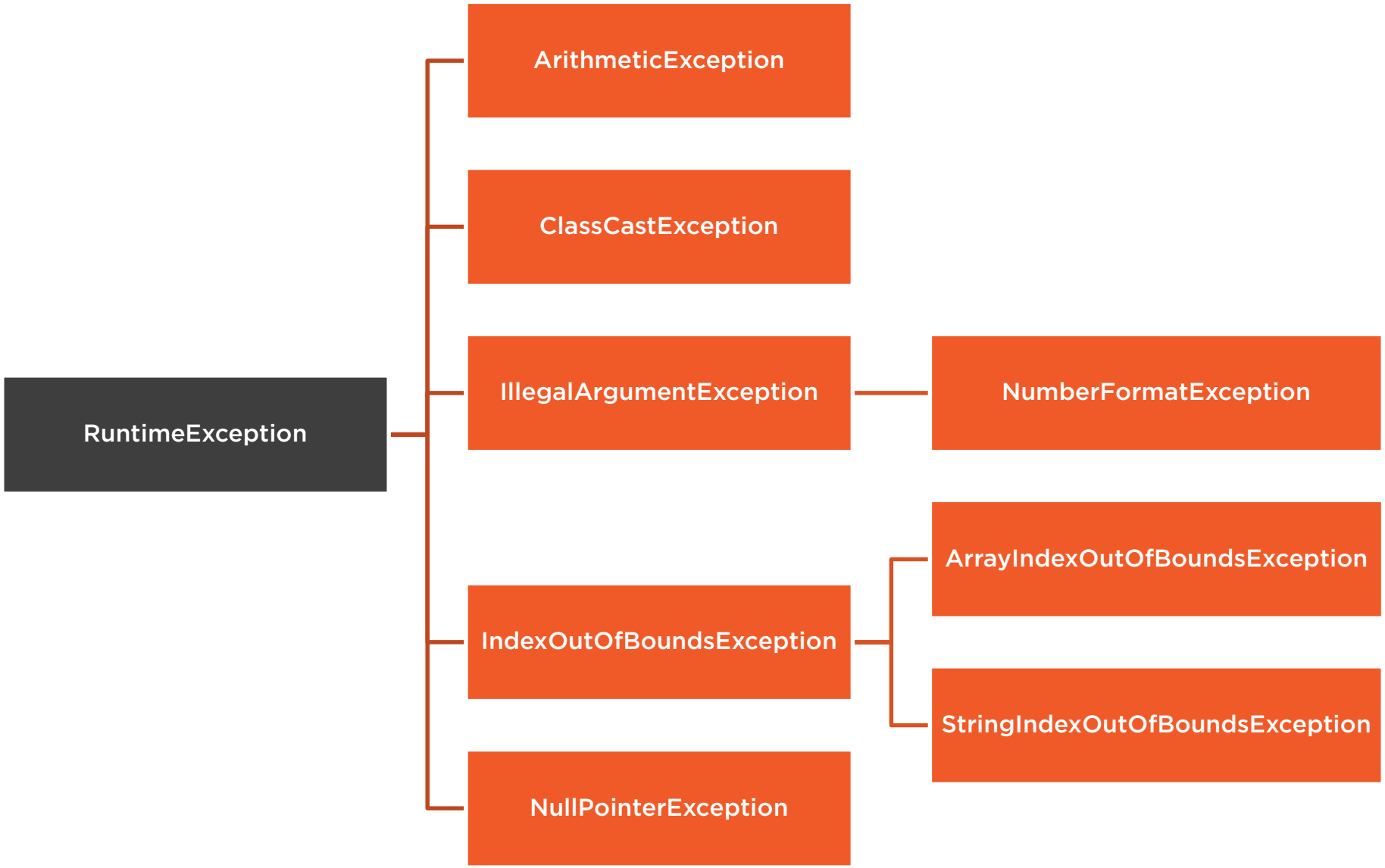
All Implemented Interfaces:

Serializable

Direct Known Subclasses:

AnnotationTypeMismatchException, ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException, CannotUndoException, ClassCastException, CMMException, CompletionException, ConcurrentModificationException, DataBindingException, DateTimeException, DOMException, EmptyStackException, EnumConstantNotPresentException, EventException, FileSystemAlreadyExistsException, FileSystemNotFoundException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException, IllegalStateException, IllformedLocaleException, ImagingOpException, IncompleteAnnotationException, IndexOutOfBoundsException, JMRuntimeException, LSException, MalformedParameterizedTypeException, MalformedParametersException, MirroredTypesException, MissingResourceException, NegativeArraySizeException, NoSuchElementException, NoSuchMechanismException, NullPointerException, ProfileDataException, ProviderException, ProviderNotFoundException, RasterFormatException, RejectedExecutionException, SecurityException, SystemException, TypeConstraintException, TypeNotPresentException, UncheckedIOException, UndeclaredThrowableException, UnknownEntityException, UnmodifiableSetException, UnsupportedOperationException, WebServiceException, WrongMethodTypeException

Common RuntimeException Classes



```
char[] latinAlphabet = new char[26];

...

char letter = latinAlphabet[30];

...
```

```
try {
    char[] latinAlphabet = new char[26];
    ...
    char letter = latinAlphabet[30];
    ...
} catch(ArrayIndexOutOfBoundsException e) {
    // Do something with the exception
}
```

```
try {
    char[] latinAlphabet = new char[26];
    ...
    char letter = latinAlphabet[30];
    ...
} catch(IndexOutOfBoundsException e) {
    // Do something with the exception
}
```

```
try {
    char[] latinAlphabet = new char[26];
    ...
    char letter = latinAlphabet[30];
    ...
} catch(RuntimeException e) {
    // Do something with the exception
}
```

```
try {
    char[] latinAlphabet = new char[26];
    ...
    char letter = latinAlphabet[30];
    ...
} catch(Exception e) {
    // Do something with the exception
}
```

Error

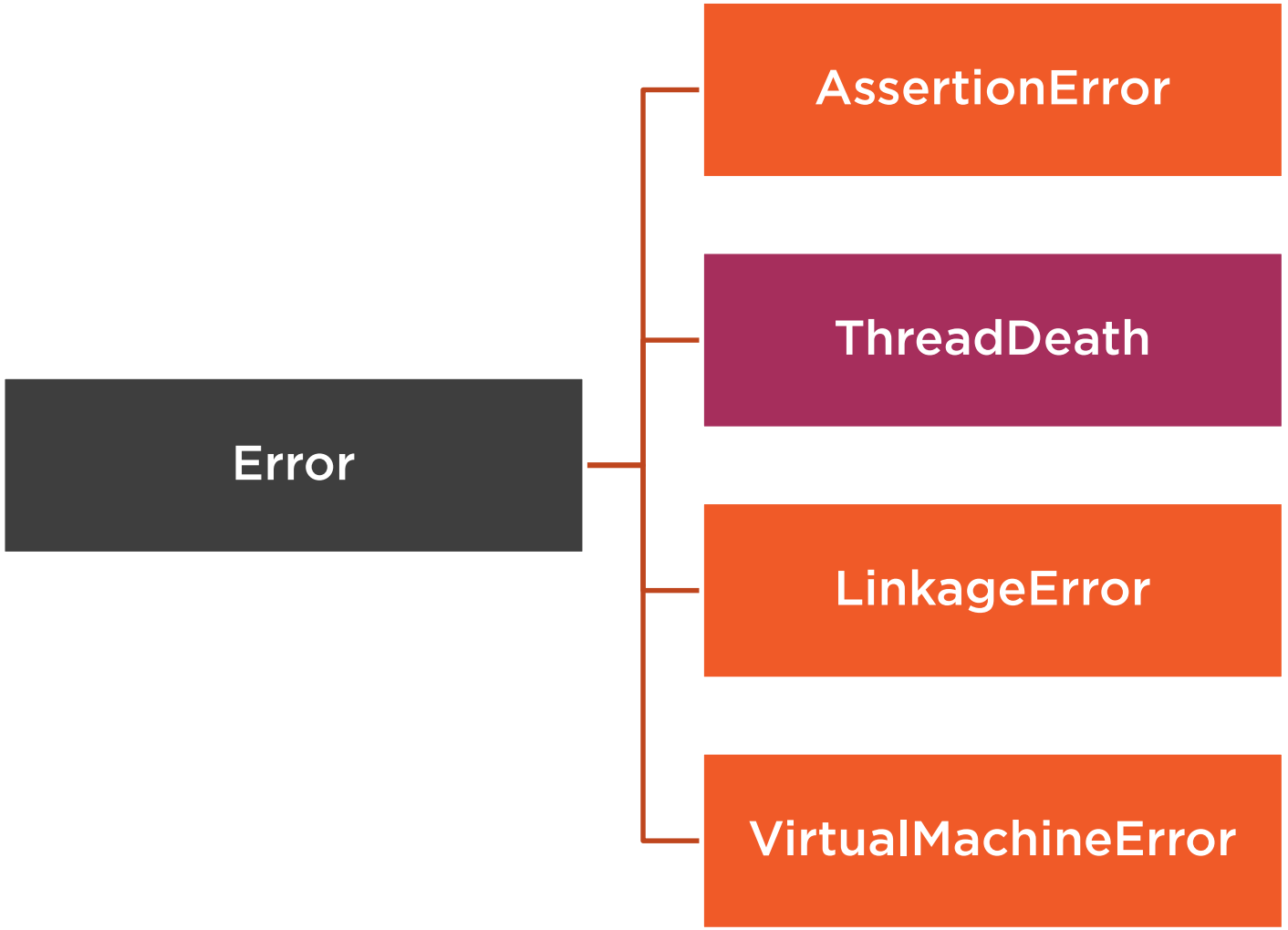
Error classes are **exempted** [from compile-time checking] because they can occur at many points in the program and recovery from them is difficult or impossible. A program declaring such exceptions would be cluttered, pointlessly. Sophisticated programs may yet wish to catch and attempt to recover from some of these conditions.

The Java® Language Specification 11.2

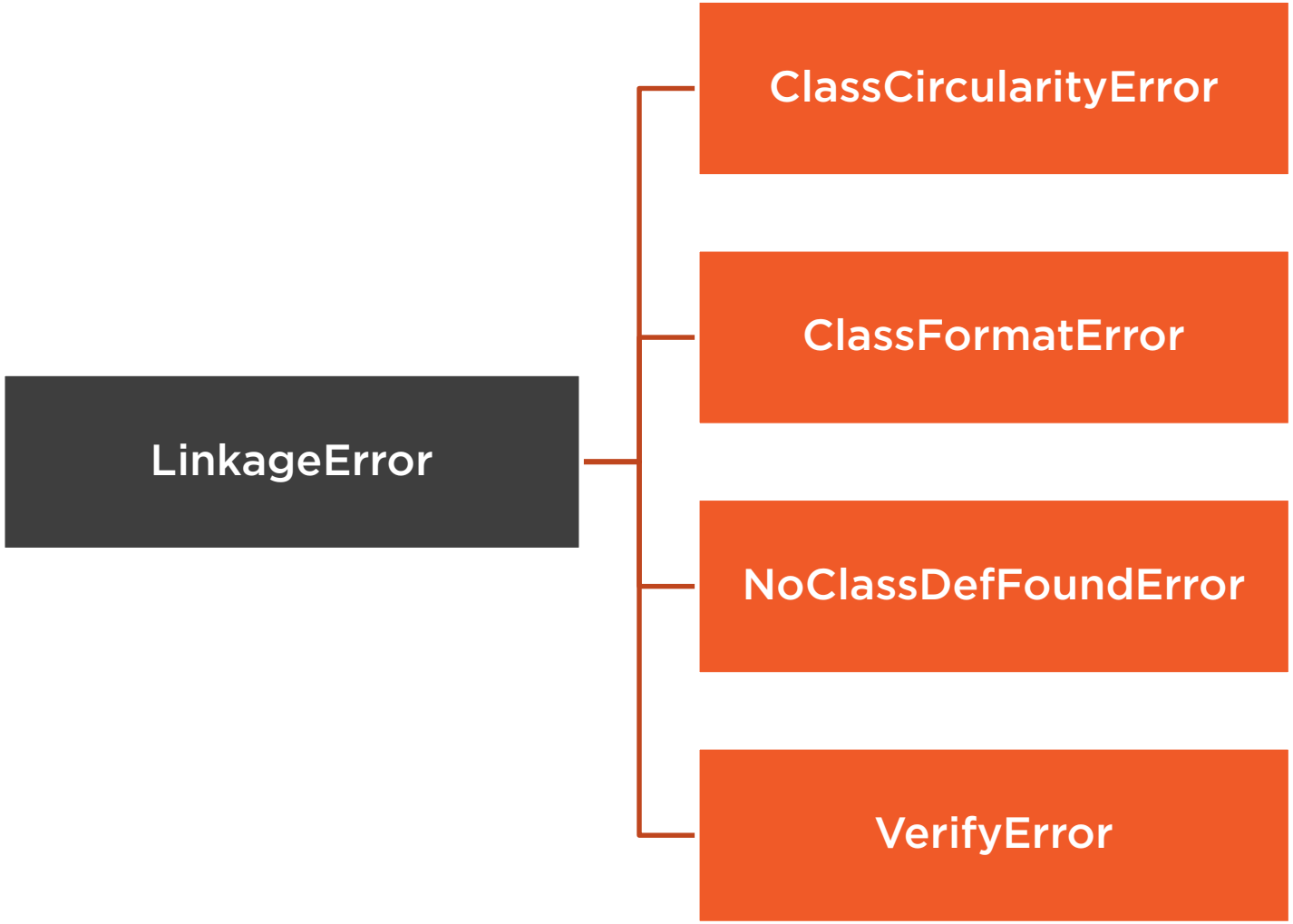
http://bit.ly/jls11-2

Never catch an Error class

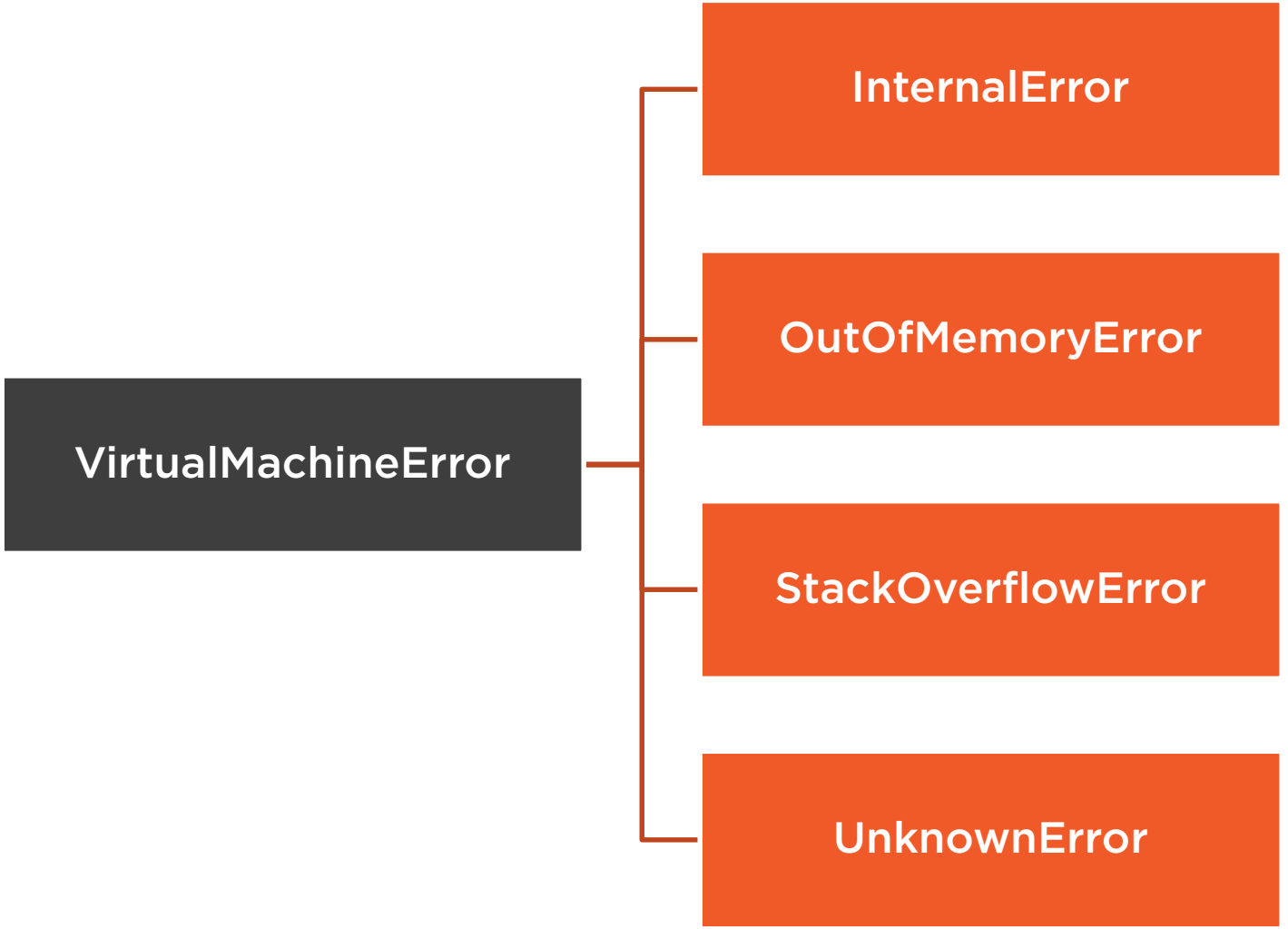
Common Error Classes



Common LinkageError Classes



Common VirtualMachineError Classes



Never catch an Error class

Common Methods

```
try {
    // ...
} catch (Exception e) {
    System.out.printf("Message: %s", e.getMessage());
}
```

String getMessage( )
String getLocalizedMessage( )

Returns the message of the exception, or the message adjusted for your particular locale.

```
try {
    // ...
} catch (Exception e) {
    if (e.getCause() != null) {
        e.getCause().printStackTrace();
    }
}
```

Throwable getCause()

Returns the cause of this throwable or null if the cause is nonexistent or unknown.

```
try {
    // ...
} catch (Exception e) {
    e.printStackTrace();
}
```

void printStackTrace( )
void printStackTrace(java.io.PrintStream)
void printStackTrace(java.io.PrintWriter)

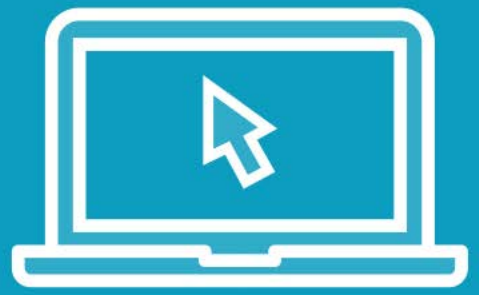
Prints the exception's stack trace to standard error or to the specified stream or writer.

```
try {
    // ...
} catch (Exception e) {
    e.fillInStackTrace();
    e.printStackTrace();
}
```

public Throwable fillInStackTrace()

Fills the exception with information about the current state of the stack trace for the current thread.

Demo



Common methods

Summary

The Exception hierarchy

Exception

RuntimeException

Error

Common Methods