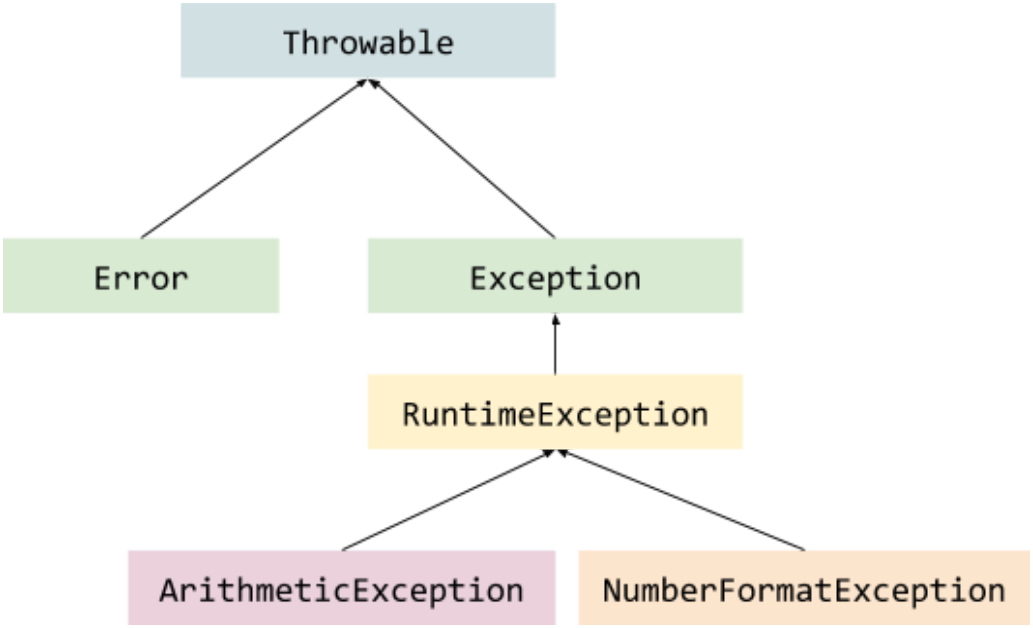
Exceptions

Reminders on Exceptions, parent class, declaration, throwing and RuntimeException specifics.



Dealing with exceptions using Try / Catch / Finally blocks. The keywords `throw` and `throws`.

```
try {

} catch (ExceptionType name) {

} catch (ExceptionType1|ExceptionType2 name) {

} finally {

}
```

A word on block variables.

Challenge #1: Guess the system output when running the class ExceptionHandling:

```
public class ExceptionHandling {

    int x;

    public void method(int i) {
        try {
            if (i > 2) {
                i = (1 / 0) + Integer.parseInt("x");
            } else {
                i = Integer.parseInt("x") + (1 / 0);
            }
        } catch (ArithmeticException e) {
            x += 4;
        } catch (RuntimeException e) {
            x += 3;
        } catch (Exception e) {
            x += 5;
        } finally {
            x += 2;
        }
        x++;
    }

    public static void main(String[] args) {
        ExceptionHandling eh = new ExceptionHandling();
        eh.method(3);
        eh.method(2);
        System.out.print(eh.x);
    }
}
```

Resolution: using Debugger.

Throwing Exceptions:

```
public Object pop() {
    Object obj;

    if (size == 0) {
        throw new EmptyStackException();
    }

    obj = objectAt(size - 1);
    setObjectAt(size - 1, null);
    size--;
    return obj;
}
```

Exceptions with JUnit

```java
import org.junit.Test;

public class ExceptionTest {

    @Test(expected = ArithmeticException.class)
    public void testNaNException() {
        @SuppressWarnings("unused")
        int i = 1 / 0;
    }

}
```

Exercise:
1/ Test the `title` value upon `Book` creation (in its Constructor), if it's `null`, throw an `IllegalArgumentException` with the appropriate message.

Make the appropriate tests in your JUnit `BookTest` class to test this functionality.

Tip:

- The JUnit `@Test` annotation takes a parameter named "`expected`". Give it the name of an `Exception` class that the tested code is supposed to throw. If the exception you've mentioned is thrown during the test, it will be considered a success.

2/ Create the `BookAttributeFormatException` class inheriting from `IllegalArgumentException` and the `TitleIsEmptyException` and `TitleIsNullException` both inheriting from `BookAttributeFormatException`. Make the appropriate changes in the `Library.newBook` method and write the appropriate unitary tests in your JUnit `BookTest` class.

3/ Create a new class `KindleIsbnFormatException` inheriting from `BookAttributeFormatException`. Override the method `setIsbn(String)` in the `KindleBook` class, in order to check if the parameter isbn starts with the letter 'e'. If not, throw a new `KindleIsbnFormatException`, with the appropriate error message.

Make the appropriate tests in your JUnit `BookTest` class to test this functionality.

Tip:

- The method `startsWith(String)` of the class `String` tests if a String starts with another passed in parameter.