

Chapter 2

Pd 1,Zhiyu Zhu

Findings of experiment with checked and unchecked exception

Throwable is a class, not interface

Javadoc annotation: `@throws ExceptionType explain`

Method signature: `throws ExceptionType1[, ExceptionType2...]`

Can include exceptions types that doesn't even throw

Errors extends Throwable, is unchecked, for unrecoverable situation.

Are mostly thrown by JVM, but we can throw it too.

Exception extends Throwable

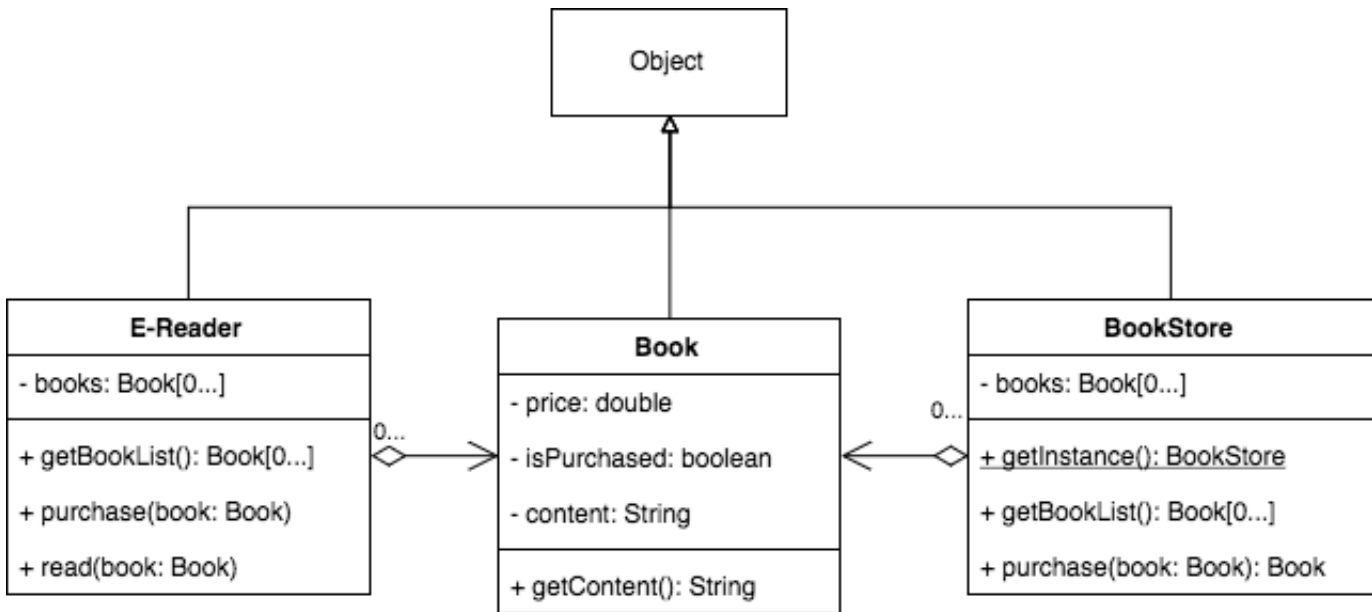
RuntimeException extends Exception, is unchecked, for identifying mistakes.

<? extends Exception>, is checked, for special case to handle

May try-catch multiple using `catch (AException | BException e)` or use multiple catch clause

```
1 // R-2.14
2 public class GuardBufferOverflow {
3     public static void main(String[] args) {
4         try {
5             args[args.length] = args[-1];
6         } catch (ArrayIndexOutOfBoundsException ignored) {
7             System.out.println("Don't try buffer overflow attacks in Java!");
8         }
9     }
10 }
```

```
1 // R-2.15
2 public class CreditCard {
3     protected double balance;
4     public void makePayment(double amount) {
5         if (amount < 0) throw new IllegalArgumentException();
6         balance -= amount;
7     }
8     /*...*/
9 }
```



```

1 // C-2.21
2 public class A { int x; }
3 public class B extends A { int x = 13; }
4 public class C extends B {
5     int x = -1;
6
7     void checkedSetAX(int i)
8         throws NoSuchFieldException, IllegalAccessException {
9         A.class.getDeclaredField("x").setInt(this, i);
10    }
11
12    void uncheckedSetAX(int i)
13        throws ClassCastException {
14        ((A) this).x = i;
15    }
16
17    public static void main(String[] args) {
18        C c = new C(); // A.x = 0, B.x = 13, C.x = -1
19
20        try {
21            c.checkedSetAX(1);
22            System.out.println(
23                A.class.getDeclaredField("x").getInt(c) + " " +
24                c.getClass().getSuperclass().getDeclaredField("x").getInt(c)
25                + " " + c.x
26            );
27        } catch (NoSuchFieldException | IllegalAccessException e) { }
28
29        c.uncheckedSetAX(2);
30        System.out.println(((A) c).x + " " + ((B) c).x + " " + c.x);
31    }
32 }
  
```