Throwing Exceptions

Andrejs Doronins

Overview

How to throw

What rethrowing means

Throwing from the OO perspective:
- Overriding & overloading

Printing the exceptions

What the course didn’t cover

Wrap up

```java
void setAge(int age) {
    this.age = age;
}
```

```java
Person p = new Person();
p.setAge(30);
```

```
✓  void setAge1(int age) throws IllegalArgumentException {
       this.age = age;
   }
   // OR
✓  void setAge2(int age) throws IOException {
       this.age = age;
   }
```

Declaring but not actually throwing!

Will this compile?

```java
void setAge1(int age) throws IllegalArgumentException {
    if(age <= 0) { throw new IllegalArgumentException("…");}
    this.age = age;
}
// OR
void setAge2(int age) throws IOException {
    //check age
    if(checkSomething()) { throw new IOException ("…");}
    this.age = age;
}
```

Should I declare runtime or checked?

```java
Person p = new Person();
// compiles
p.setAge1(30);
```

```java
Person p = new Person();
// fails, unhandled exception
p.setAge2(30);
```

```
Person p = new Person();

// compiles

p.setAge1(30);
```

```
Person p = new Person();

try {
  p.setAge2(30);
} catch (...) { }
```

`if(whatever) { throws new Exception(); }`	✗
`void setAge() throw Exception {...}`	✗
`if(whatever) { throw new Exception(); }`	✓
`void setAge() throws Exception {...}`	✓

Runtime exceptions can occur anywhere in a program, and in a typical one they can be very numerous.

Having to add runtime exceptions in every method declaration would reduce a program's clarity.

Thus, the compiler does not require that you catch or specify runtime exceptions (although you can).

TLDR: You *can* add Runtime exceptions to the method signature, but avoid it.

```
void calculate() {
    Data d = fetchData();
    // handle data
}
Data fetchData() {
    try {
    Connection conn = openAConnection();
    } catch (IOException e) { ... }
    return  conn.queryDb("...");
 }
```

```
void calculate() {
    Data d = fetchData();
    // handle data
}
Data fetchData() throws IOException {
    Connection conn = openAConnection();
    return  conn.queryDb("...");
 }
```

You handle it!

Let the next one do it...

```
void calculate() throws IOException {
    Data d = fetchData();
    // handle data
}
```

You handle it!

```
Data fetchData() throws IOException {
    Connection conn = openAConnection();
    return  conn.queryDb("...");
 }
```

Demo

Declaring exceptions in the method signature

Exceptions in Method Signatures

Overriding

Overloading

```
class Parent {
    void doThing() throws IOException {
    }
}

class Child extends Parent {
    @Override
    void doThing() throws Exception { ✗
    }
}
```

When:

A class overrides a method from a super class or implements a method from an interface

Then:

It's not allowed to add new checked higher-level exceptions to the method signature

```
class Parent {
    void doThing() { }
}

class Child extends Parent {
    @Override
    void doThing() /* no throwing of checked exceptions */ { }
}
```

```java
class Parent {
    void doThing() throws IOException { }
}
class Child extends Parent {
    @Override
    void doThing() throws
            FileNotFoundException,  ✓
            IOException,            ✓
             Exception {    }       ✗
}
```

```
class SomeClass {

    void doThing() throws IOException { }

    void doThing() throws RuntimeException { }
}
```

signature

not part of the signature

```java
public static void main(String[] args) {
    try {
        throw new RuntimeException("oops!");
    } catch (Exception e) {
        System.out.println(e);
        System.out.println(e.getMessage());
        e.printStackTrace();
    }
}
```

```
java.lang.RuntimeException: oops!
```

```
oops!
```

```
java.lang.RuntimeException: oops!
        at com.package.main(ClassName.java:7)
```

Exception handling rules!

Clean code principles!

Further Study

Course: Java: Writing Readable and Maintainable Code

- Module “Handling Exceptions”

Book: Effective Java

- Chapter on Exceptions

Summary

Exception handling is indispensable in programming

Syntax and rules of try/catch/finally

Catch chaining

Exception class hierarchy

How to throw and print exceptions

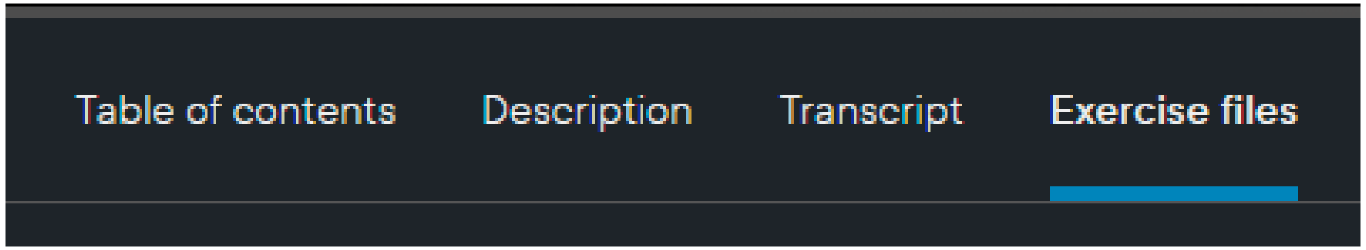
Table of contents
Description
Transcript
Exercise files

Rating

Thank you!

(Happy coding)