

Custom Exceptions and Processing Text Files

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

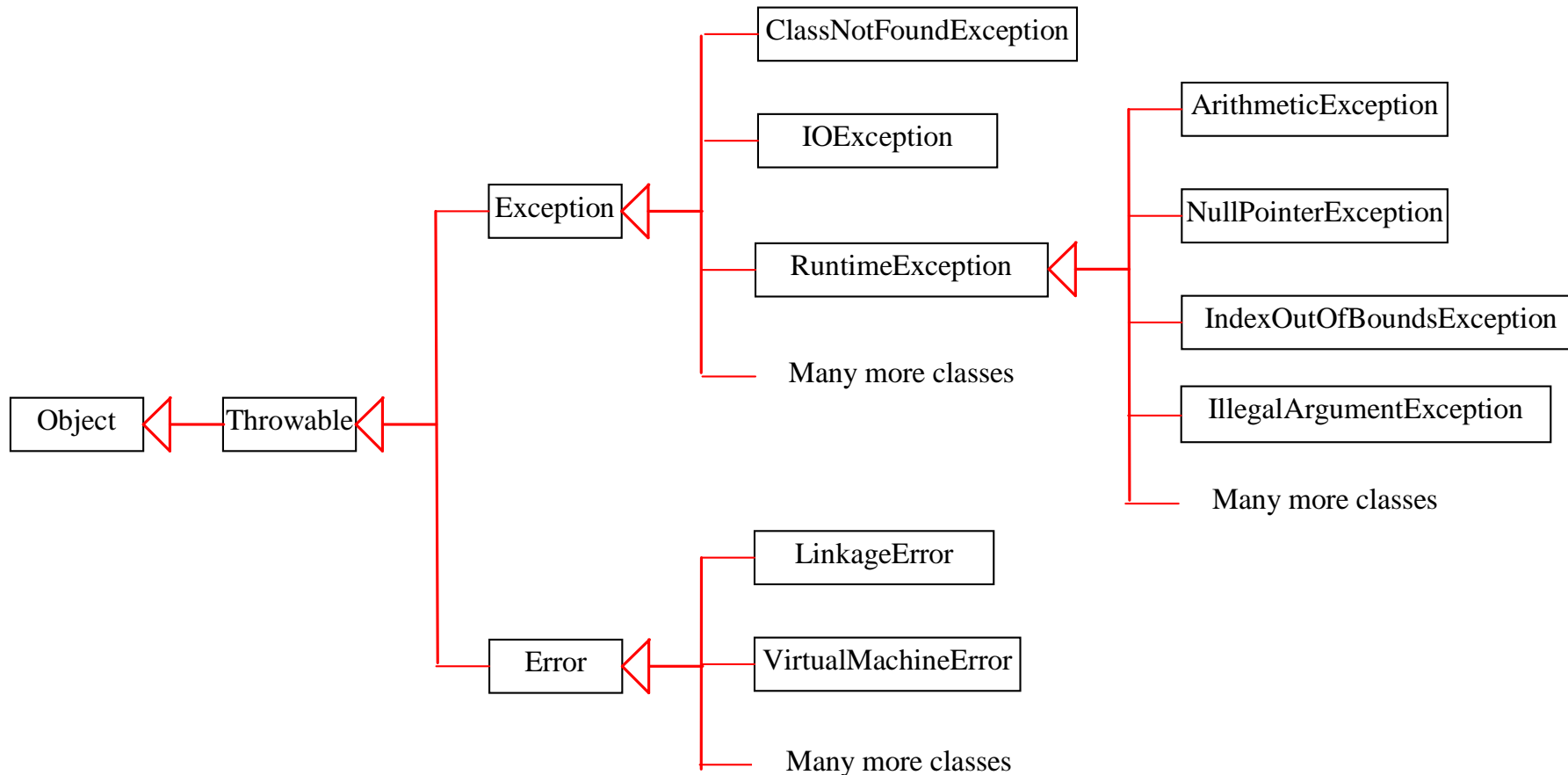
Outline

- Discussed
 - Approaches to handle errors (what-if and exceptions)
 - Concept of Exception
 - The Java throwable class hierarchy
 - system errors, runtime exceptions, checked errors, unchecked errors
 - Methods of declaring, throwing, catching exception, and rethrowing exceptions
 - Exception, call stack, stack frame, and stack trace
 - Some best practice
 - File system path (to identify file)
 - Concept of text file (Java API classes and text file)
- Exception and simple text/character File I/O
 - Reliable processing text file (patterns and exceptions)

Lesson Outline

- Review relevant concepts
 - Creating custom exceptions
 - Designing file structures for application programs
 - Reading, processing, and writing text files
- Demo: custom exceptions and processing text files
- Exercise:
 - Complete an exercise

Review: Exception Hierarchy



Review: Custom Exceptions

- Subclass a Throwable class
 - Which class to subclass?
 - Is it meaningful (the name? the type of error to deal with?)
 - Is it checked or unchecked exception?

```
public class InvalidRadiusException extends IllegalArgumentException {  
...  
}
```

vs.

```
public class InvalidRadiusException extends IllegalStateException {  
...  
}
```

Review: Design Text File Structures

- There are a variety ways to design text file structures
- Ask questions:
 - What constitutes a record?
 - How many records?

Defining Record

- Examples

- Based on lines, such as a line is a record

- Based on sentinels

```
BEGIN RECORD
```

```
John Doe
```

```
2900 Bedford Ave
```

```
END RECORD
```

- Based on counters

```
2
```

```
John Doe
```

```
2900 Bedford Ave
```

Number of Record

- Based on sentinel, such as the end of file (no more record to read)
- Based on counter, e.g.,
2
John Doe, 2900 Bedford Ave
Jane Doe, 2900 Bedford Ave

Review: Reading and Writing Files

- Java classes
 - Reading file: `java.util.Scanner`
 - Writing files: `java.io.PrintWriter`
- and several related classes and concepts
 - `java.io.File`, `java.nio.file.Path`
 - Text files and character encodings

Problem 1. Demo

Suppose a school stores their student data in a text file where the text file contains unbound lines of text, each line consists of the name, the gender, the GPA, and an unbound list of hobbies of the student, all separated by a comma, e.g.,

```
John Doe,M,3.939,Basketball,Volleyball,Tennis
```

```
Jane Doe,F,4.000,Tennis,Softball,Computer games
```

```
Amy Doe,L,4.000,Tennis,Softball
```

Write a program that identifies the candidates for a sports team (such as basketball team). The candidates must be have a GPA no less than a threshold and whose hobby include the sport.

The user is expected to use the program as the example below:

```
java ListCandidates students.txt 3.000 basketball candidates.txt
```

The program prints out a summary on the standard output, and write the list of the students to the candidates.txt that has the same format as the input file.

The program also uses exceptions (including custom exceptions) to handle errors

Problem 2. Your Exercise

- (Based on Question 12.25) A university posts its employee's salaries at <http://liveexample.pearsoncmg.com/data/Salary.txt> that you should download and save it locally on your computer. Each line in the file consists of a faculty member's first name, last name, rank, and salary.
- Write a program that takes the salary.txt file as input, and
 - that checks the existence of the output file, salarystats.txt, and if it exists and is newer than the data, displays the information described below on the standard output;
 - if the output file does not exist or older than the input file, process the input file and obtain the following
 - the total salary for assistant professors, associate professors, and full professors, and faculty (all ranks), respectively,
 - the average salary for assistant professors, associate professors, and full professors, and faculty (all ranks), respectively, and
 - and display these on the standard output and also write these to the output file.
- The program also uses exceptions (including custom exceptions) to handle errors (following the demo)