Object Oriented Programming

Kookmin University

Department of Computer Science

Announcements

- Final exam in Dec. 4th.
- Interesting read
    - http://www.businessinsider.com/9-tech-skills-that-pay-over-120000-2016-10
    - In the programming Job postings, the portion of required language
        - Java: 13%, SQL: 13%
        - Javascript: 11%
        - C#: 7%, C++: 7%, C: 5%
        - PHP: 5%, Perl: 5%, PL/Sql (Oracle database): 5%

Naming Convention

- Different languages have different conventions
- For Java
  - methodsAreNamedWithLowerCamelCase
  - variablesAreAlsoLowerCamelCase
  - CONSTANTS_ARE_IN_ALL_CAPS_WITH_UNDERSCORES
  - ClassNameIsCapitalizedAsUpperCamelCase
  - package.name.is.lowercase.and.separated.by.dots
- Choose short words but avoid abbreviations (msg VS. message)

Setting Code Formatter For Eclipse

- https://stackoverflow.com/questions/3754405/best-eclipse-code-formatters

The eclipse configurable Java guide can be found here. Installing the guide is quite simple, Save the style xml to a file and import it using the following dialogue box:

Window -> Preferences -> Java -> Code Style -> Formatter

Finally select the active profile as *GoogleStyle*.

Why Multiple Inheritance Is Not Allowed

- Because the language designer decided not to…
- Comments from C# (another representative OOP language) creator
  - Duplicate methods in multiple parent classes can result in confusion/conflict
  - Benefit of multiple inheritance is small comparing to additionally incurred complexity
  - The feature of multiple inheritance can be achieved using Interface
  - Multiple inheritance injects a lot of complexity – casting, layout, dispatch, field access, comparison, …

Benefit of Using Interface

- A child class can implement (inherit) multiple parent interface classes
  - Note that a child class can inherit only one parent class by default
- Use the multiple inheritance feature to have an object more degree of Polymorphism
- Code : SmartPhone, MusicPlayable, PhotoTakable
- Imagine how to implement a digital camera, MP3 player
  - Code: DigitalCamera, Mp3Player

Interface Seems Similar to Abstract

- Differences between Interface and Abstract Class
  - Interface contains only method declaration
  - Abstract class can contain both method declaration and implementation

  - A class can implement multiple interfaces – more polymorphism
  - A class can extend only one abstract class

  - All methods declared in interface should be implemented
  - All methods declared in abstract class do not have to implemented

Use Interface or Abstract Class

- Multiple inheritance is necessary
  - Interface: multiple inheritance possible
- Need some default implementations with method declaration only
  - Abstract can have both concrete implementation and declaration
- Type declaration
  - Interface is preferable as it generally contains distinct concrete implementation in child classes and multiple inheritance
- Is-a VS. Can-do
  - General suggestion: use abstract class for Is-a hierarchy and use interface for Can-do (Circle, Square – Shape and Fly - Flyable)

Interface VS. Abstract Class

- Interface generally defines capability - note multiple implementation of Interface
  - Flyable, Runnable, Eatable, …
- Define horizontal capability (feature) using Interface. In vertical relationships, use abstract method
  - Abstract: Animal – Cat, Dog, Tiger
  - Interface: Eatable, Walkable, …
- Combination of abstract and interface class
  - Abstract class implements multiple interface while providing some common methods implementation
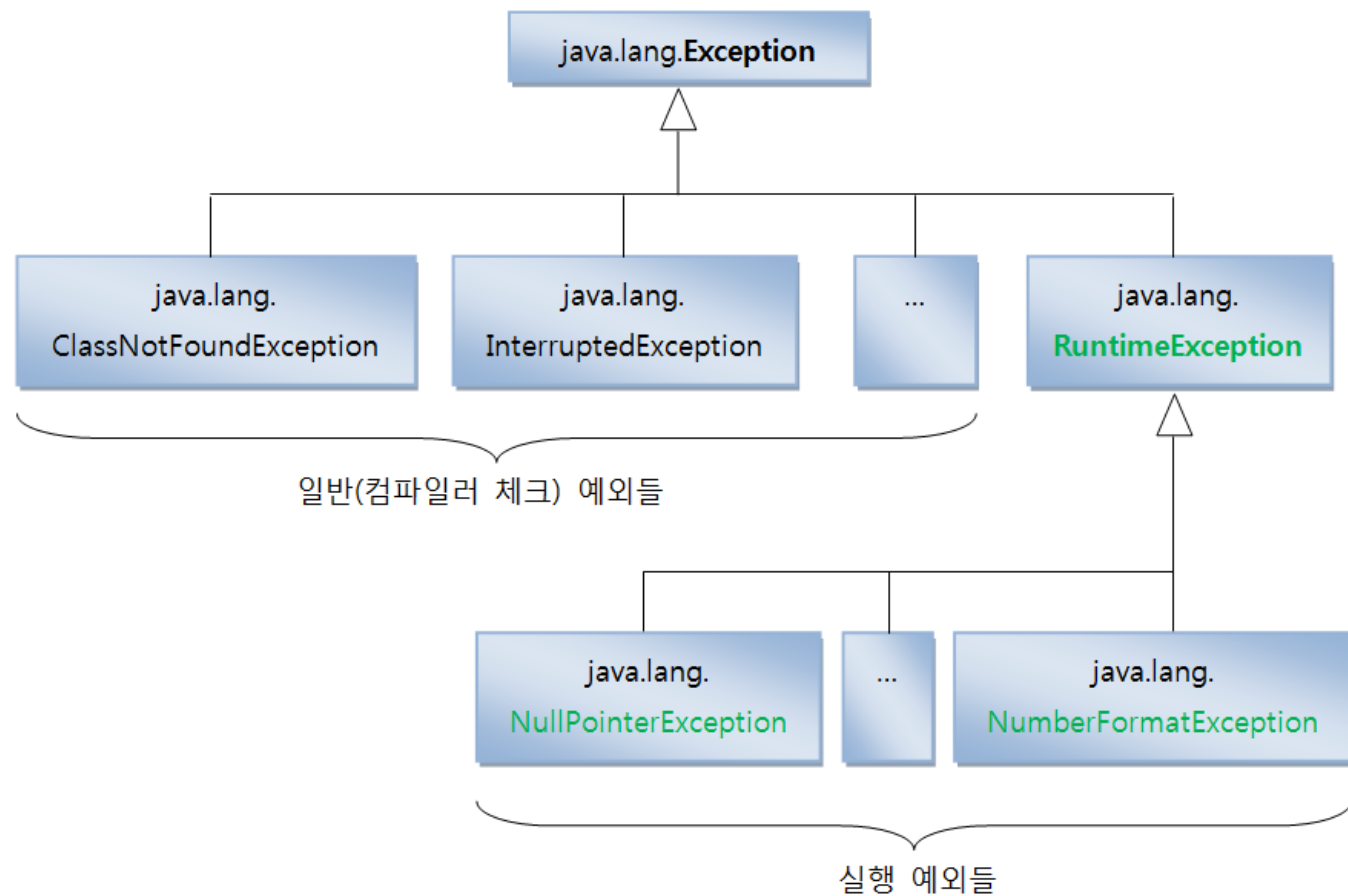
Inheriting Interface

- Interfaces can be inherited
  - Parent interface can be inherited to child classes
  - A class that implements a child class should implement all methods declared in the parent class
  - InterfaceExample
    - How does it work?
    - Is it a good code?

Exception

- During code execution, how to handle if an error has occurred?
  - Examples: user input error, code bug
- We need to have a way to deliver an error message to a method caller and take an appropriate action
- Exception
  - A class that defines an error
  - An object is created and it is passed to a caller method
- RuntimeException
  - An exception that happens during program execution

Types of Exception

이것이자바다 422, 423

RuntimeException

- RuntimeException happens during Java program execution – not during the compile time
- Examples
  - NullPointerException - it happens when an object with a null pointer is accessed
  - ArrayIndexOutOfBoundsException – happens when an array is indexed out of its capacity

```java
public void nullPointerException() {
    String name = null;
    System.out.println(name.toString());
}
```

```java
public void arrayIndexException() {
    String[] names = new String[5];
    System.out.println(names[6]);
}
```

Handling Exception

- try-catch-finally
- try block
  - A code block that might contain causes for exception
- catch block
  - A code block to handle an exception
- finally block
  - A code block that is executed after try or catch block execution
  - It contains codes that should be executed regardless of exception happening

Exception Handling Syntax

```
try {
    // code body
} catch (Exception e) {  // get an exception as a variable
    // handle an Exception using a variable e
} finally {  // optional code
    // wrap up the code
}
```
• Code – Examples

Handling Multiple Exceptions

- Using single catch block
  - catch (Exception1 | Exception2 | Exception3 e)
- Using multiple catch blocks
  - catch (Exception1 e1) {}
  - catch (Exception2 e2) {}
  - catch (Exception3 e3) {}

Pass Exceptions to Caller

- When a method does not want to handle exception and pass it to a caller method, add throws clause in the method declaration

- Syntax
  - METHOD_DECLARATION throws Exception1, Exception2 {}

- Such method should be handled with try,catch block

- Or a caller method should pass the exception

User Defined Exception and Throw It

- Create a class similar to other class

- Inherit RuntimeException class

- If you want to throw an exception, call
  - throw new EXCEPTION_NAME();

References

- 이것이자바다 – 한빛미디어 2015
- http://javarevisited.blogspot.kr/2013/05/difference-between-abstract-class-vs-interface-java-when-prefer-over-design-oops.html
- http://stackoverflow.com/questions/995255/why-is-multiple-inheritance-not-allowed-in-java-or-c