

COMP 110-001

Package & Review of Classes

Yi Hong

June 04, 2015

Today

- Package in Java
- Review of classes

What is Package?

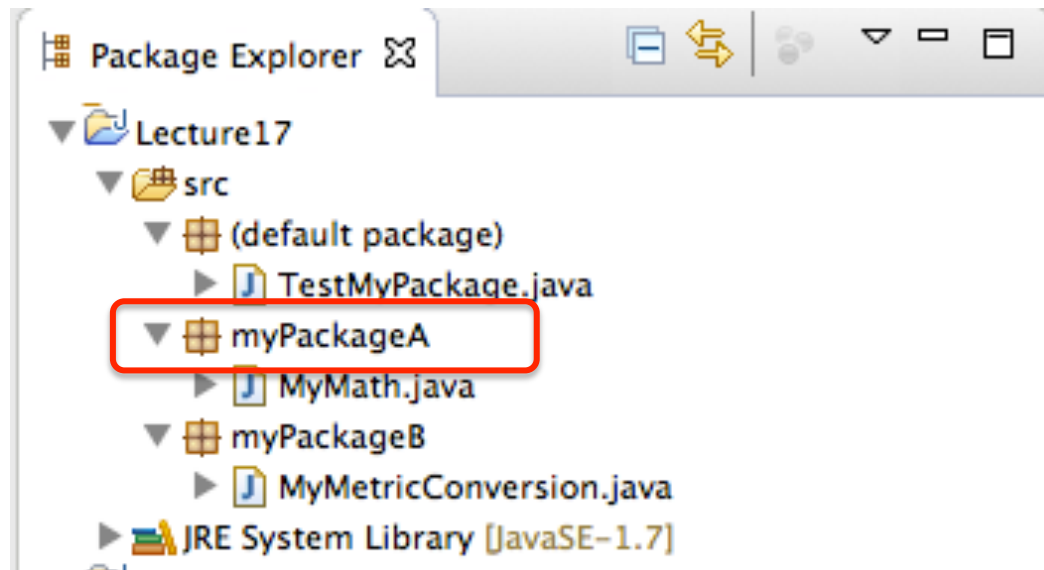
- A collection of classes grouped together **into a folder**
 - group related classes into one unit
 - We want to divide a big project into multiple components (at many levels)
 - The highest level is a package
 - Package -> Class -> Method

The package Statement

- Each class is in a separate file named after the class
- Each file in the package must begin with a package statement, ignoring blank lines and comments
- Syntax for a class in a package
 - <Blank lines or comments>
 - package Package_Name;
 - <A class definition>
 - E.g.: package java.util;

Create a Package

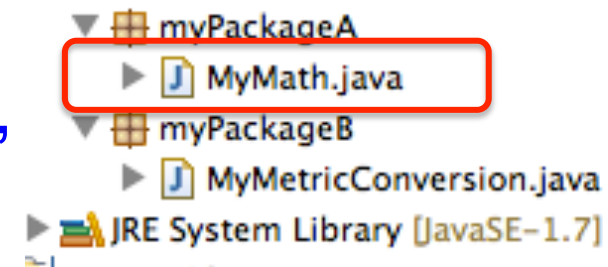
- Create a new package in Eclipse
 - Right click your project in Package Explorer
 - New -> Package
 - Enter Package name (e.g.: myPackageA)



Group Classes into a Package

- Put Java class files into the package folder

- In Eclipse, src/packageName/
- If you are creating a new class, just right click on package and choose New -> Class.



- Each class file must have this line at the beginning of the file:

`package Package_Name;`

```
1 package myPackageA;  
2  
3 public class MyMath {  
4
```

Existing Packages in Java

- Java has provided you many packages
- We have used them in previous labs / assignments

java.lang — basic language functionality and fundamental types

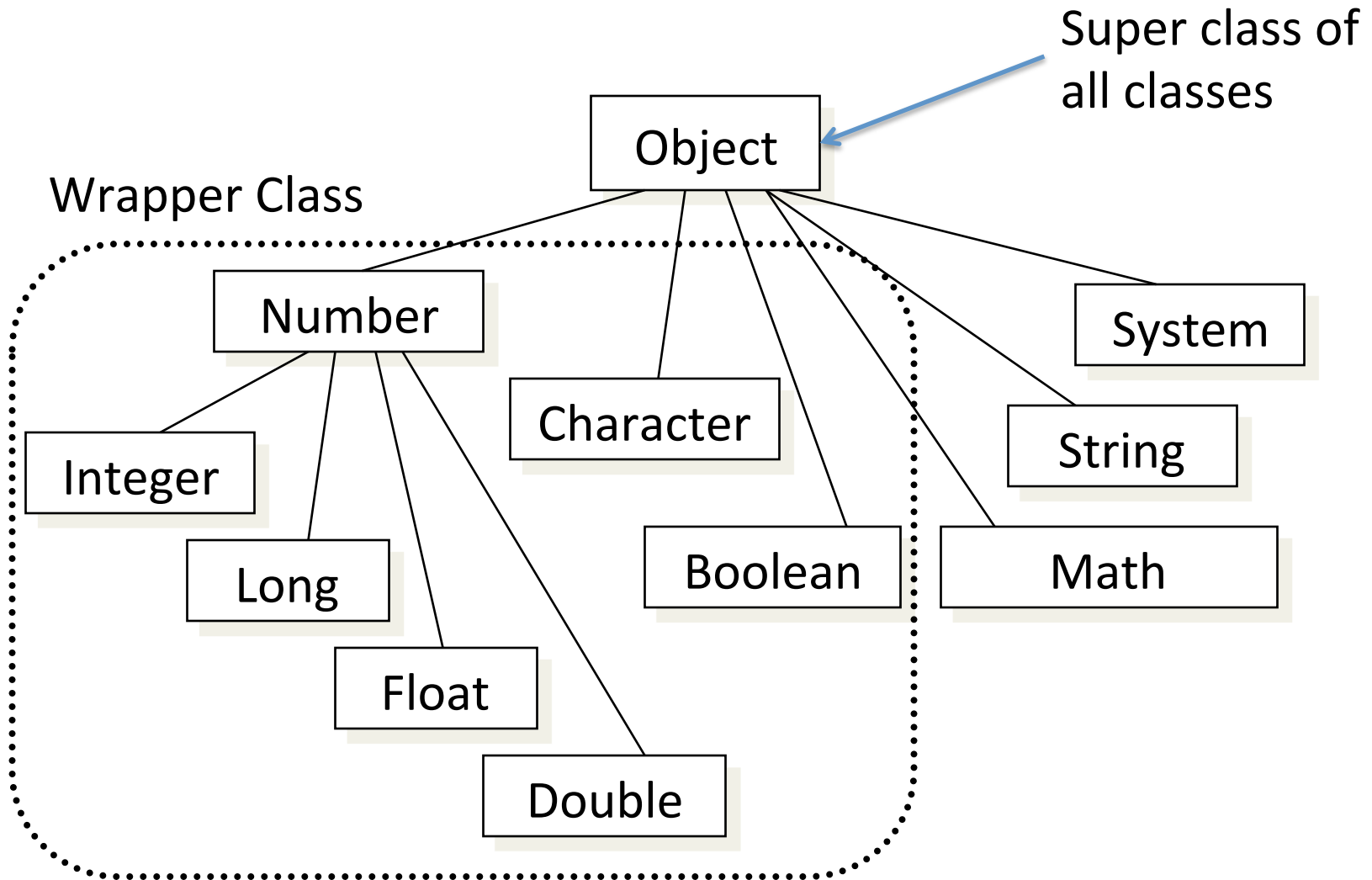
all wrapper classes (**Integer**, **Double** ...) are in this package

also **String**

also **System** (yeah. The **System** in **System.out.println**)

and the **Math** class - (all the static methods, **abs()**, **log()**, **pow()**
....)

Classes in java.lang Package



Existing Packages in Java

java.util — classes for representing and manipulating data collection

- We have used **Scanner**, **Random**, **ArrayList** in this package

java.awt and **java.swing** — classes for building graphical user interface

- We have used them in lab 3 and lab 5.

java.io — file operations (read / write files)

- We will discuss this one soon

The `import` Statement

- To use a class in a package, place an `import` statement that names the package at the start of the file
 - Classes in the same package can use each other directly

- Syntax

```
import Package_Name.Class_Name_Or_Asterisk;
```

```
E.g.: import java.util.Scanner;
```

```
import java.io.*;
```

Using Classes in a Package

- The import statement should be **AFTER** the package statement if you have the package statement:

```
package myPackageA;  
import java.util.*; // use * to import everything in a package  
public class Class1 {  
  
    .....  
  
}
```

- **One exception:** java.lang is always automatically imported
 - That's why we can use System.out.println()....

Summary on Package

- Package corresponds to folder
- Classes in one package must have the package name declared in the first line
- To use classes in other packages, use import statement
- import statement has to be after package declaration if any

Review of Classes via An Example

- Implement a class named Rectangle defined by two vertices (the bottom-left vertex and the top-right vertex with 2D points (x, y) , x, y are integers)
 - Provide three different constructors, `Rectangle()`, `Rectangle(int x1, int y1, int x2, int y2)`, and `Rectangle(Point p1, Point p2)`
 - Design a method to compute the area
 - Design a method to compare whether two rectangles are the same
 - Design a static method to check whether two rectangles are overlapped

- Demonstrate in class, and check the sample code on the course website
 - `Point.java`, `Rectangle.java`, and `RectangleDriver.java`

Next Class

- Arrays