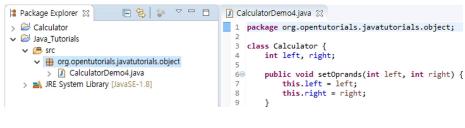
## **Package**

## **Package**

- 클래스의 이름 중복 문제를 해결하기 위한 약속
- 서로 다른 내용의 Java.txt라는 두 파일은 이름이 같아 한 컴퓨터에 공존할 수 없다.
- 그래서 고안된 것이 Directory이고, 두 Java.txt 파일을 각각 A와 B라는 디렉토리에 저장하면 된다.
- 클래스가 많아짐에 따라서 같은 이름을 가진 클래스가 생겨날 가능성이 높아지게 되는데 이름의 충돌을 방지하기 위해 고안된 것이 패키지라고 할 수 있다.



package org.opentutorials.javatutorials.object;



- 패키지의 이름은 클래스를 만드는 개발자가 소속된 도메인 이름을 사용한다.
- 디렉토리 경로와 패키지의 이름이 일치하고 패키지는 디렉토리 안에 존재한다.
- Src 디렉토리 하위에

> J A.java

> J B.java

> A JRE System Library [JavaSE-1.8]

- org₩opentutorials₩javatutorials₩object₩CalculatorDemo4
- com\oracle\object\CalculratorDemo4
- 똑같은 이름의 CalculatorDemo4가 공존할 수 있게 된다.

## Use Package

i. class A와 B는 같은 Package안에 있으므로 컴파일 OK! (컴파일 시 같은 Package안의 클래스를 Load한다.)

```
☐ Package Explorer 
☐

                                                 ☐ CalculatorDemo4.java ☐ A.java ☒ ☐ *B.java
> 🖾 Calculator
                                                 package org.opentutorials.javatutorials.packages.example1;
Java_Tutorials
                                                      public class A {
  4
     > # org.opentutorials.javatutorials.object
                                                   5
                                                      }

    matter org.opentutorials.javatutorials.packages.example1

                                                   6
        > J A.java
        > D B.java
   > A JRE System Library [JavaSE-1.8]

☐ Package Explorer 
☐
                                                package org.opentutorials.javatutorials.packages.example1;
> 🐸 Calculator

    Java_Tutorials

                                                     public class B {
  public static void main(String[] args) {
     > # org.opentutorials.javatutorials.object
                                                             A a = new A(); // class A의 인스턴스회

    g.org.opentutorials.javatutorials.packages.example1
```

}

8

ii. class A와 B는 서로 다른 Package에 있으므로 컴파일 불가!

```
☐ Package Explorer 
☐

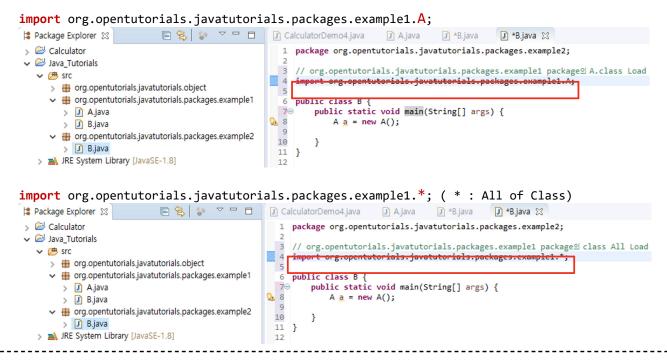
☑ CalculatorDemo4.java ☑ A.java ☑ *B.java
package org.opentutorials.javatutorials.packages.example2;
public class B {
  🗸 进 src
                                                           public static void main(String[] args) {
     > # org.opentutorials.javatutorials.object
                                                   2 5
                                                                A = \text{new } A();

    # org.opentutorials.javatutorials.packages.example1

                                                                // ERROR : Class A를 사용하려고 하니 example2 package에 존재하지 않음
                                                     6
       > J A.java
                                                     8
                                                        }
        > D B.java
                                                     9
       org.opentutorials.javatutorials.packages.example2
       > J B.java
   > A JRE System Library [JavaSE-1.8]
```

Import 선언

- 특정 Package안에 있는 Class를 Load 할 수 있는 기능
- 현재 Package와는 다른 Package의 class를 Load하겠다.



## 로드된 패키지들간의 중복 회피

i. class D 에서 class B를 인스턴스하는데 import된 example1, example2에 동일한 이름의 B.class가 있어서 ERROR class B의 이름이 중복되기 때문에 애매함(ambiguous)의 문제가 발생한다.

```
☐ Package Explorer 
☐
                              (-) Q | - - -

☑ CalculatorDemo4.java

                                                                                         📝 *B.java 🚺 *B.java 💢
                                                                             A.java
> 😂 Calculator
                                                           package org.opentutorials.javatutorials.packages.example3;
                                                        2⊖ import org.opentutorials.javatutorials.packages.example1.*;
import org.opentutorials.javatutorials.packages.example2.*;
   🗸 进 src
     > # org.opentutorials.iavatutorials.object
                                                        5
                                                           public class D {

    # org.opentutorials.javatutorials.packages.example1

                                                         6
                                                                public static void main(String[] args) {
        > J A.java
                                                                   B b = new B();
        > J B.java
                                                       9

    # org.opentutorials.javatutorials.packages.example2

                                                       10
         > J B.java
                                                       11 }

    # org.opentutorials.javatutorials.packages.example3

                                                       12
        > D.java
   > A JRE System Library [JavaSE-1.8]
```

ii. 해결방법 - example1의 B.class라고 명시적으로 지정하기

```
☑ CalculatorDemo4.java
                       A.java

    ★B.java

                                              package org.opentutorials.javatutorials.packages.example3;
 2⊖ import org.opentutorials.javatutorials.packages.example1.*;
 import org.opentutorials.javatutorials.packages.example2.*;
    public class D {
  70
        public static void main(String[] args) {
            B b = new B();
 9
                       🗽 The type B is ambiguous
 10
 11 }
                       2 quick fixes available:

    Explicitly import 'org.opentutorials.javatutorials.packages.example1.8'

                        4- Explicitly import 'org.opentutorials.javatutorials.packages.example2.8'
                                                                    Press 'F2' for focus

☑ CalculatorDemo4.java

                                                                   A.java

→ *B.java
  1 package org.opentutorials.javatutorials.packages.example3;
😘 20 import org.opentutorials.javatutorials.packages.example1.*;
🕠 3 import org.opentutorials.javatutorials.packages.example2.*;
  5 public class D {
  6
          public static void main(String[] args) {
  7⊝
<u>)</u> 8
              org.opentutorials.javatutorials.packages.example1.B 💆
 9
              = new org.opentutorials.javatutorials.packages.example1.B();
 10
 11
 12 }
 13
```