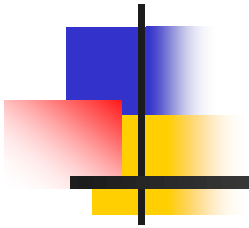
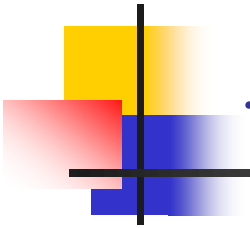


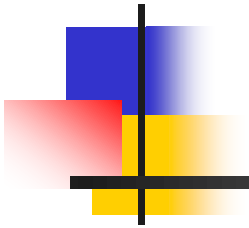
Java 8강 패키지

양 명 숙
[now4ever7@gmail.com]

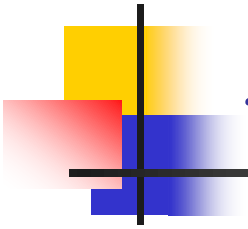


목차

- 클래스 패스
- 패키지



클래스 패스



클래스 패스(Class Path)의 지정

- 클래스 패스 – 클래스의 경로(클래스가 존재하는 경로)를 뜻함
  - 자바 가상머신은 프로그램의 실행과정에서 실행에 필요한 클래스를 찾을 때, 바로 이 클래스 패스를 기준으로 찾게 됨

- 환경변수 path - 확장자가 exe인 실행파일을 찾는 경로
- 환경변수 classpath - 확장자가 class인 클래스파일을 찾는 경로

환경변수에 대한 이해 – path

- C:₩myclass> calc.exe
- c:₩myclass> explorer.exe

환경변수 path의 정보에
calc.exe와 explorer.exe의 디렉터리정보가
등록되어있기 때문에 실행가능함

```
C:\myclass> echo %path%
```

환경변수path 확인 명령문

```
.;C:\WINDOWS\system32;C:\Program Files\Java\jdk1.6.0_10\bin;
```

출력결과

다음 세 경로에서 실행파일을 찾게 됨

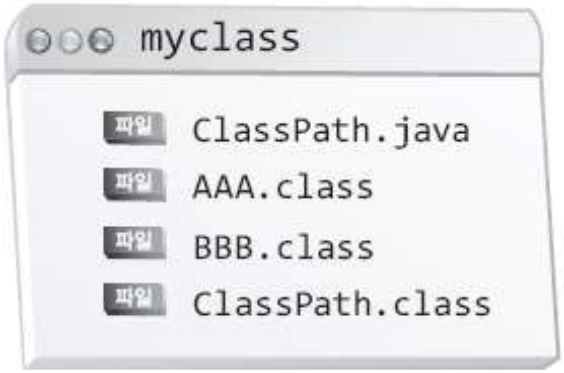
- 경로 1 → .
- 경로 2 → C:\WINDOWS\system32
- 경로 3 → C:\Program Files\Java\jdk1.6.0_10\bin

예제-ClassPath.java

```
AAA
BBB
```

```
class AAA
{
	public void printName()
	{
		System.out.println("AAA");
	}
}
class BBB
{
	public void printName()
	{
		System.out.println("BBB");
	}
}
class ClassPath
{
	public static void main(String args[])
	{
		AAA a=new AAA();
		a.printName();

		BBB b=new BBB();
		b.printName();
	}
}
```

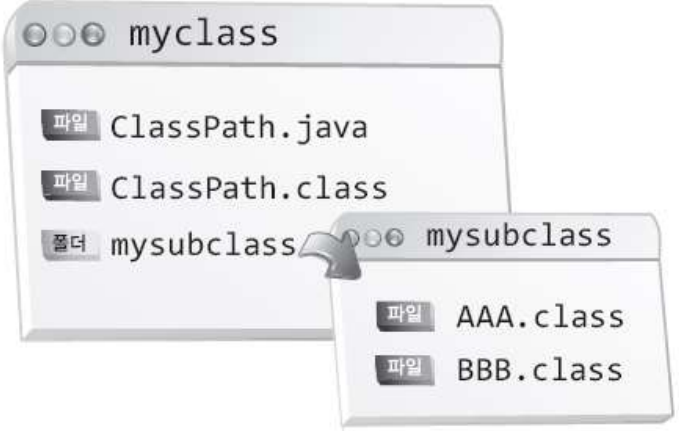
- main 메소드는 ClassPath.class에 존재
- 실행의 형태
  C:□myclass> java ClassPath
- 정상적인 실행결과 확인 가능



실행위치와 클래스파일의 위치

- 컴파일된 클래스 파일의 위치 이동

- AAA.class, BBB.class를 이동시켜서 실행
- 실행의형태
  C:□myclass> java ClassPath
- 클래스를 찾지못해서 에러 발생!
- AAA.class, BBB.class를 찾지 못함



- java.exe 가 실행된 디렉토리(현재 디렉토리)에서만 가상머신에 올려질 클래스를 찾기 때문
- mysubclass에서 클래스파일을 검색하라는 정보를 전달할 수 있어야 함

```
Exception in thread "main" java.lang.NoClassDefFoundError: AAA
        at ClassPath.main(ClassPath.java:21)
Caused by: java.lang.ClassNotFoundException: AAA
        at java.net.URLClassLoader$1.run(URLClassLoader.java:202)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(URLClassLoader.java:190)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:301)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:247)
        ... 1 more
```

환경변수 classpath 추가하기

- classpath의 설정 및 확인방법

```
C:\myclass> echo %classpath%
%classpath%
C:\myclass>set classpath=.;
C:\myclass>echo %classpath%
.;
```

classpath의 확인

classpath가 설정되지 않은 상태의 출력

현재 디렉터리를 classpath에 등록

문제의 해결을 위한 클래스패스의 설정

- 실행을 위한 classpath의 지정

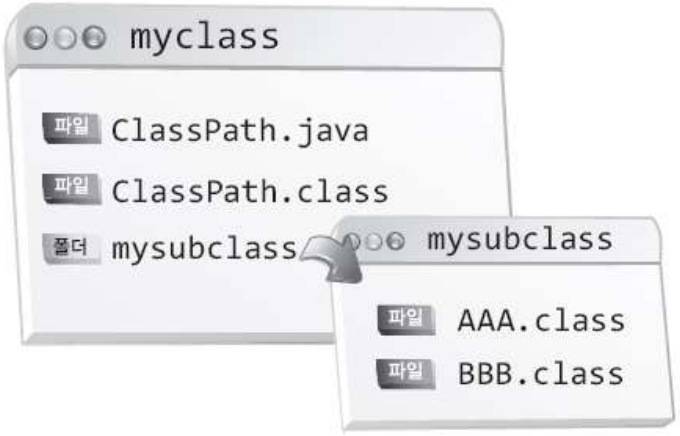
절대경로 지정방식

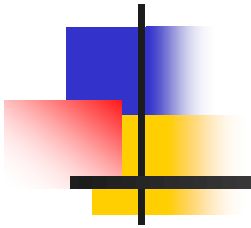
```
C:\myclass> set classpath=.;C:\myclass\mysubclass;

C:\myclass> set classpath=.;.\mysubclass;
```

상대경로 지정방식

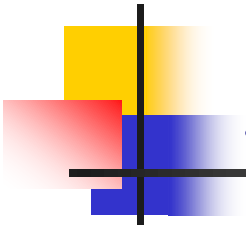
- 해당 명령 프롬프트 창에서만 유효한 환경변수
- 제어판-시스템-고급-환경변수 -> 모든 명령 프롬프트상에 적용

C:□myclass> java ClassPath





패키지



패키지(package)

- 패키지(package)
  - 클래스의 묶음
  - 패키지에는 클래스나 인터페이스를 포함시킬 수 있음
  - **서로 관련된 클래스들끼리 그룹 단위로 묶어** 놓음으로써 클래스를 효율적으로 관리할 수 있음
  - 같은 이름의 클래스일지라도 서로 다른 패키지에 존재하는 것이 가능하므로, 자신만의 패키지 체계를 유지함으로써 다른 개발자가 개발한 클래스 라이브러리의 클래스와 이름이 충돌하는 것을 피할 수 있음
  - 같은 이름의 클래스 일지라도 서로 다른 패키지에 속하면 패키지명으로 구별 가능

패키지(package)

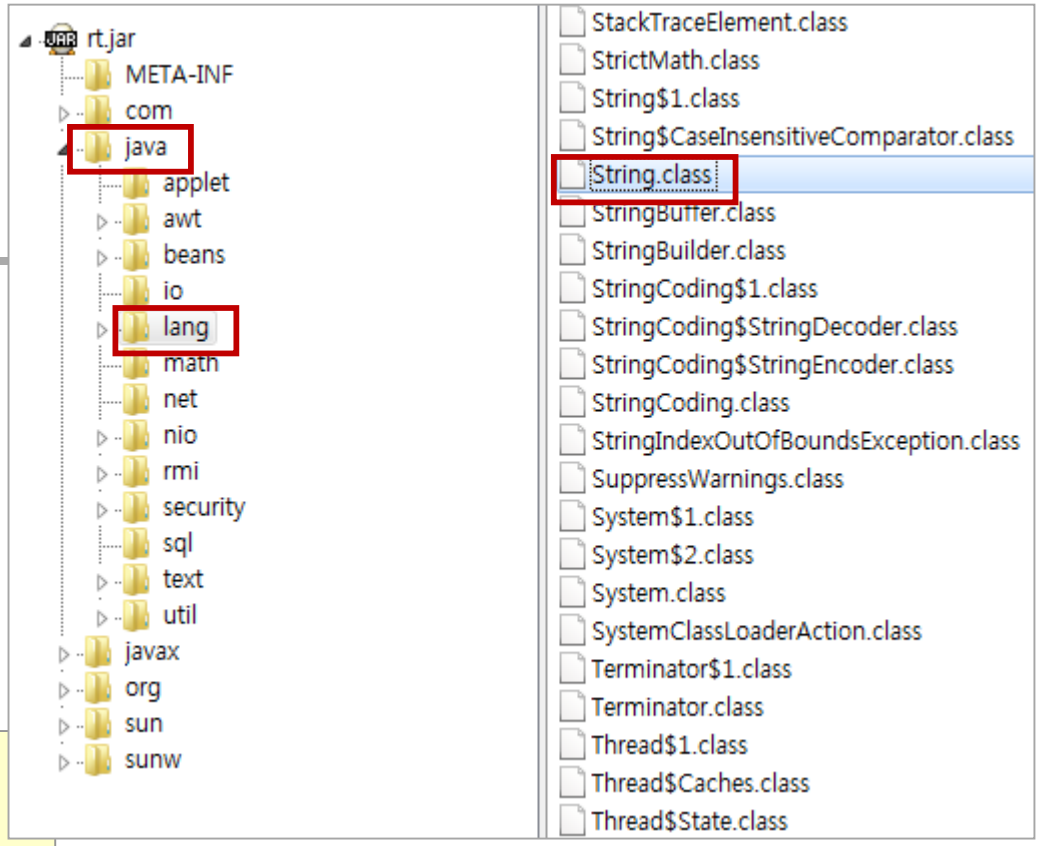
- 클래스 – 물리적으로 하나의 클래스 파일(.class)
- 패키지 – 물리적으로 하나의 디렉토리(폴더)
  - 어떤 패키지에 속한 클래스는 해당 디렉토리에 존재하는 클래스 파일(.class)이어야 함
  - 예) java.lang.String 클래스 – 물리적으로 java 디렉토리의 서브디렉토리인 lang에 속한 String.class 파일임

- 클래스의 실제 이름(full name)
  - 패키지명을 포함한 것
  - 예) String 클래스 => java.lang.String
    - Date 클래스 => java.util.Date

```
java.util.Scanner sc = new java.util.Scanner(System.in);
```

rt.jar

- String 클래스는 rt.jar 파일에 압축되어 있음
- jar 파일(*.jar) – 클래스와 관련 파일들을 압축한 것



C:□Java□jdk1.8.0_65□jre□lib

```
package java.lang;
public final class String extends Object
                implements Serializable,
                Comparable<String>, CharSequence
{
                public char charAt(int index){..}
                public int indexOf(int ch,int fromIndex){..}
                ....
}
```

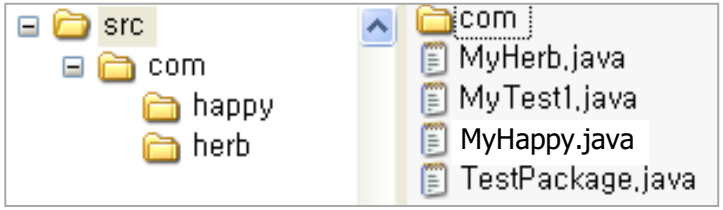
패키지의 선언

```
package 패키지명;
```

- 패키지 선언문은 반드시 소스파일에서 주석과 공백을 제외한 **첫 번째 문장이어야** 함
- 하나의 소스파일에 **단 한번만 선언**될 수 있음
- 해당 소스파일에 포함된 모든 클래스나 인터페이스는 선언된 패키지에 속하게 됨
- **모든 클래스는 반드시 하나의 패키지에 포함되어야** 함
  - 패키지를 선언하지 않고 사용하면, 기본적으로 제공하는 '이름없는 패키지 (unnamed package)'에 자동으로 속하게 됨

- 하나의 소스파일에는 첫 번째 문장으로 단 한 번의 패키지 선언만을 허용한다.
- 모든 클래스는 반드시 하나의 패키지에 속해야 한다.
- 패키지는 점(.)을 구분자로 하여 계층구조로 구성할 수 있다.
- 패키지는 물리적으로 클래스 파일(.class)을 포함하는 하나의 디렉토리이다.

```
java₩0626₩src>java com.TestPackage
```



예제

```
package com;

public class  MyTest1{
      public void display1(){
            System.out.println("MyTest1 Class");
      }
}
-------------------------------------
package com.happy;

public class  MyHappy{
      public void displayHappy(){
            System.out.println("MyHappy Class");
      }
}
-------------------------------------
package com.herb;

public class MyHerb{
      public void displayHerb(){
            System.out.println("MyHerb Class");
      }
}
```

package com;

```
import com.happy.*;
import com.herb.*;

public class TestPackage
{
    public static void main(String[] args)
    {
            MyTest1 m1=new MyTest1();
            m1.display1();

            MyHappy m2=new MyHappy();
            m2.displayHappy();

            MyHerb h=new MyHerb();
            h.displayHerb();
    }
}
```

예제

- java₩src₩ 소스 파일이 모두 있는 경우
- 컴파일
  - d:₩java₩src>javac -d . MyTest1.java
  - d:₩java₩src>javac -d . MyHappy.java
  - d:₩java₩src>javac -d . MyHerb.java
  - d:₩java₩src>javac -d . TestPackage.java
  - => -d 옵션 뒤에는 해당 패키지의 루트 디렉토리의 경로를 적어줌
  - . => 현재 디렉토리
- 패키지의 루트 디렉토리('com의 상위 디렉토리 : src')를 클래스 패스에 포함시켜야 함
  - d:₩java₩src>set classpath=.;d:₩java₩src;
- 실행
  - d:₩java₩src>java com.TestPackage

이름 없는 패키지

- 자바의 클래스는 반드시 하나의 패키지에 포함되어야 함
  - 별도의 패키지 선언이 존재하지 않는 파일에 정의되어 있는 클래스들은 "이름 없는 패키지" 로 묶이게 됨
  - 패키지 선언을 하지 않은 클래스들간의 패키지 접근권한을 부여하기 위함
- 패키지 컴파일을 위한 컴파일 옵션
  - 패키지의 구성과 동일한 형태로 폴더를 만드는 것이 귀찮다면 -d 옵션을 추가해서 컴파일

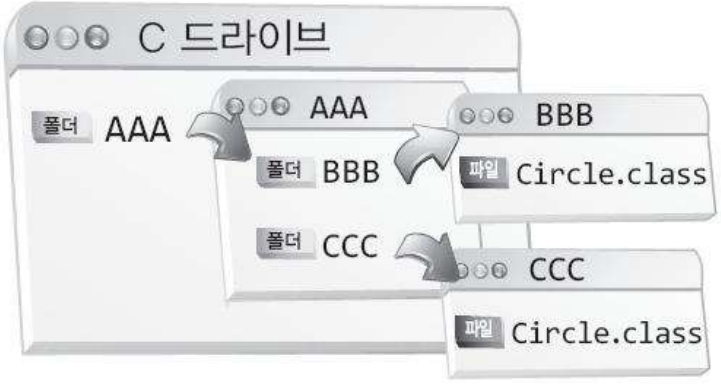
```
c:□myclass>javac -d  "패키지 생성 디렉토리"  "컴파일할 파일 이름"

=> 패키지에서 명시하는 디렉토리가 자동으로 생성됨
```

  - 패키지의 생성위치 지정– "패키지 생성 디렉토리" 이용

```
c:□myclass>javac -d . circle.java
        => 현재 디렉토리에 패키지가 생성됨

c:□myclass>javac -d  mydir circle.java
        => 현재 디렉토리의 서브 디렉토리인 mydir에 패키지가 생성됨
```

패키지의 상위 디렉토리가 클래스패스에 등록되어 있어야 함

패키지와 클래스패스의 관계



패키지도 클래스패스를 기준으로 검색
(클래스의 이름과 패키지의 이름은 클래스 패스를 기준으로 찾게 됨)

BBB와 CCC는 패키지로 선언되어야 함.

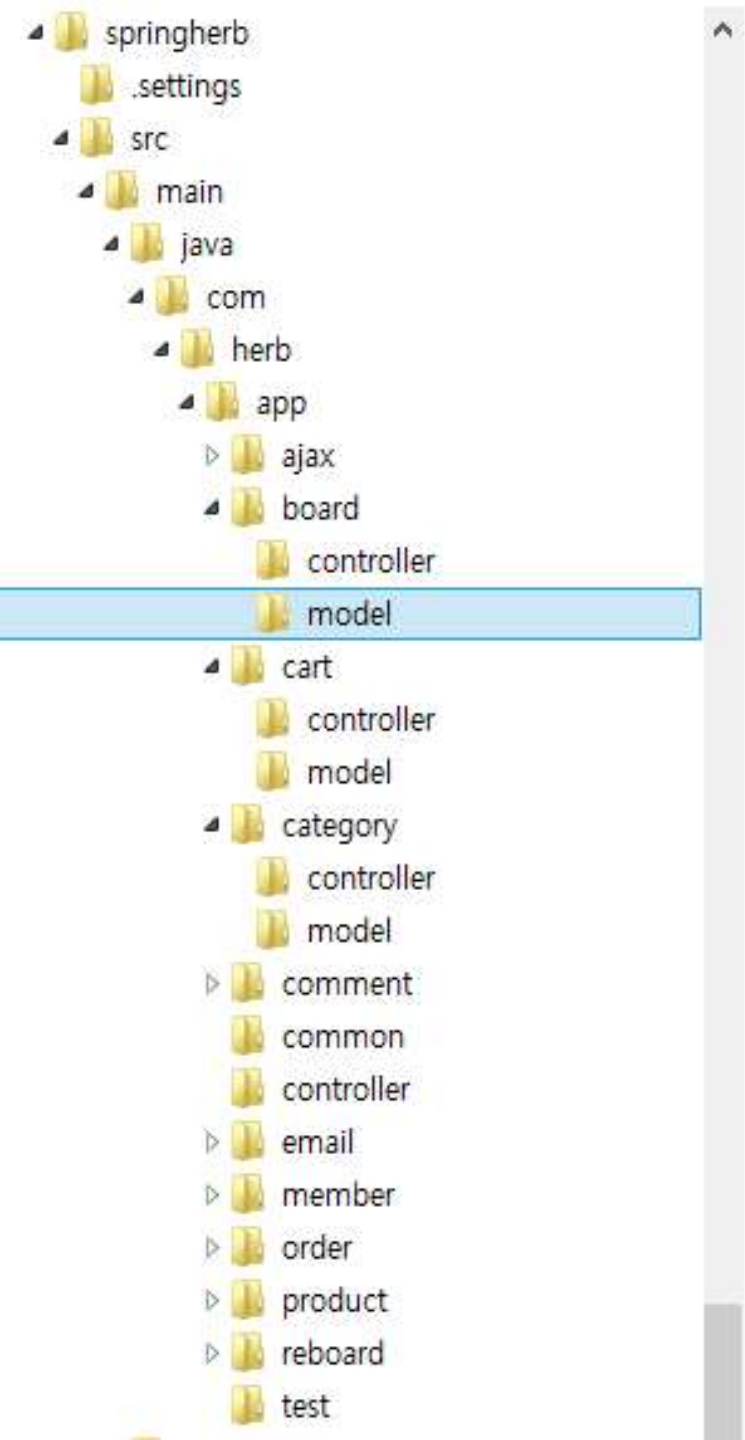
```
BBB.Circle c1=new BBB.Circle();
CCC.Circle c2=new CCC.Circle();
```

C:\AAA 디렉터리가 클래스패스에 등록되어 있어야함

AAA.BBB와 AAA.CCC는 패키지로 선언되어야 함.

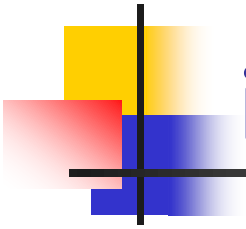
```
AAA.BBB.Circle c1=new AAA.BBB.Circle();
AAA.CCC.Circle c2=new AAA.CCC.Circle();
```

C:\디렉터리가 클래스패스에 등록되어 있어야 함



패키지 사용 예

```
package com.herb.app.board.model;

import java.sql.Timestamp;

public class BoardVO {
		private int no;
		private String name;
...
}
```



import 문

- import 문의 선언
  - import 패키지명.클래스명;
  - 또는
  - import 패키지명.*;
    - import 문에서 클래스의 이름 대신 '*'을 사용하는 것이 하위 패키지의 클래스까지 포함하는 것은 아님

• import 문은 package 문 다음에,
그리고 클래스 선언문 이전에 위치해야 함

1) package문
2) import문
3) 클래스 선언

패키지의 선언

"이 클래스는 orange.area 패키지에 묶겠다!"

이클래스를 orange\area 디렉터리에 저장하고,
이경로를 명시해서 인스턴스를 생성하겠다

```
package orange.area; // 패키지 선언

public class Circle
{

   . . . .

}
```

인스턴스의 생성방법

```
orange.area.Circle c1=new orange.area.Circle(1.5);
System.out.println("반지름이 1.5인 원의 넓이 : "+c1.getArea());
```

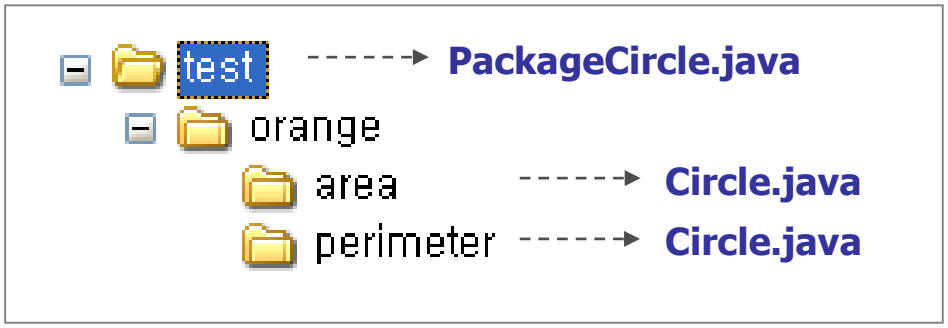
예제

```
package orange.area;

public class Circle
{
        double rad;
        final double PI;

        public Circle(double r)
        {
                rad=r;
                PI=3.14;
        }
        public double getArea()
        {
                return (rad*rad)*PI;
        }
}
```

```
package orange.perimeter;

public class Circle
{
        double rad;
        final double PI;

        public Circle(double r)
        {
                rad=r;
                PI=3.14;
        }
        public double getPerimeter()
        {
                return (rad*2)*PI;
        }
}
```

예제-PackageCircle.java

```
class PackageCircle
{
    public static void main(String[] args)
    {
        orange.area.Circle c1=new orange.area.Circle(1.5);
        System.out.println("반지름이 1.5인 원의 넓이: "+c1.getArea());

        orange.perimeter.Circle c2 = new orange.perimeter.Circle(2.5);
        System.out.println("반지름이 2.5인 원의 둘레: "+c2.getPerimeter());
    }
}
```



import 선언

```
import orange.area.Circle;
```

"orange.area 패키지의 Circle을 의미할 때에는 다 생략하고 Circle만 표시하겠다!"

- import 선언 이후의 두가지 인스턴스 생성방법

```
orange.area.Circle c1=new orange.area.Circle(1.5);
Circle c2=new Circle(2.5);
```

```
import orange.area.*;
```

- 모호함이 발생하는 대표적인 사례

```
import orange.area.Circle;
import orange.perimeter.Circle;
```

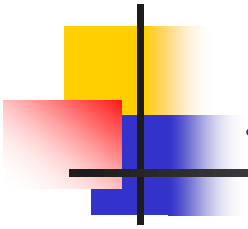
"orange.area 패키지로 묶여있는 클래스의 인스턴스 생성에서는 패키지의 이름은 생략하고 클래스의 이름만 명시하겠다."

접근 제한자 예제-protected

```
package com;

public class  MyTest1
{
		protected String name;
		int age;
		public String job;

		public void display1(){
			System.out.println("MyTest1 Class");
		}
}
```

```
package com.herb;

public class MyHerb
{
		protected String tel;
		int grade;
		public String addr;

		public void displayHerb(){
			System.out.println("MyHerb Class");
		}
}
```

```
package com;
import com.herb.*;
class ChildHerb extends MyHerb{
	public  void printInfo(){
		System.out.println("MyHerb  protected tel : " + tel);
		//System.out.println("MyHerb  default grade : " + grade);
		System.out.println("MyHerb  public addr : " + addr);
	}
}//class

public class TestPackage2{
	public static void main(String[] args) {
		MyTest1 m1=new MyTest1();  //동일 패키지
		m1.display1();
		System.out.println("MyTest1  protected name : " + m1.name);
		System.out.println("MyTest1  default age : " + m1.age);
		System.out.println("MyTest1  public job : " + m1.job);

		MyHerb h=new MyHerb();  //다른 패키지
		h.displayHerb();
		//System.out.println("MyHerb  protected tel : " + h.tel);
		//System.out.println("MyHerb  default grade : " + h.grade);
		System.out.println("MyHerb  public addr : " + h.addr);

		ChildHerb ch = new ChildHerb();
		//System.out.println("ChildHerb, MyHerb  default tel : " + ch.tel);  //error
		//System.out.println("ChildHerb, MyHerb  protected grade : " + ch.grade);  //error
		System.out.println("ChildHerb, MyHerb  public addr : " + ch.addr);
} }
```



개발툴 설치

- Eclipse 툴
  - 통합 개발 환경
  - http://www.eclipse.org
  - Eclipse IDE for Java EE Developers, wind64bit
  - zip 파일- 압축을 풀어주기만 하면 됨
- 이클립스에서
  - window 메뉴 - Preference - Java - Compiler : 버전 1.8 선택
  - Java -Installed JREs : jdk 1.8.0

  - 프로젝트 만들기
    - File 메뉴-New - Project - Java Project - project name 지정(basic)

이클립스 단축키

Alt + Up , Alt + Down : 해당 라인을 위 / 아래로 이동 시킨다.
Ctrl + Alt + Down : 커서가 위치한 라인을 복사해 밑줄에 생성해 준다.
Ctrl + Shift +X : 대문자로 변환
Ctrl + Shift + Y : 소문자로 변환

- Ctrl + Shift + F4 : 열려 있는 모든 창
- Ctrl + spacebar : 자동완성
- ALT+SHIFT+R : 텍스트를 일괄 변경
  - 변수 하나에 클릭 후 커서를 둔 상태로 단축키를 눌러주면 똑같은 변수 이름들에 테두리가 쳐지며 하나의 변경 만으로 일괄적으로 수정이 됨
- F11 : 디버그
- CTRL + F11:실행
- CTRL + I : 소스들의 "자동 들여쓰기"
  - 작성 된 소스들을 전체 선택 후, 단축키를 입력 해 주면, 코딩 중 놓친 들여쓰기를 친절하게 자동으로 처리
- CTRL + SHIFT + F : 자동 코드 정리
- Ctrl + M : editor 최대화
- Ctrl + Shift + M : add import
- Ctrl + Shift + O : add import
- Ctrl + / : 주석 처리 – 한 라인/블록에 대해 주석 처리 (추가 및 제거)
- Ctrl + Shift + / : 블록 주석(/* */)
- Ctrl + Shift + ₩ : 블록 주석 제거
- F3 : 메서드 정의로 이동, Alt+← : 이전 위치로, Alt+→ : 다음 위치로
- sysout + (Ctrl + Spacebar) : System.out.println() 문장 삽입

이클립스 단축키

- 새로 만들기 Ctrl+N
- Quick Access Ctrl+3
- 빨리 수정 Ctrl+1
- 한 줄 삭제 Ctrl+D
- 파일 아웃라인 Ctrl+O
- 상속 구조 Ctrl+T
- 선택 문자 찾기 Ctrl+K
- 행번호로 이동 Ctrl+L
- 파일명으로 찾기(Open Resource) Ctrl+Shift+R
- Open Type : Ctrl+Shift+T
- 프로젝트 텍스트 검색 Ctrl+H
- 단축키 목록 Ctrl+Shift+L

과제-성적 관리 프로그램

```
1. 성적 입력
2. 전체학생의 성적 조회
3. 학생별 성적 조회
4. 클래스(반)별 성적 조회
5. 학생 성적 수정
6. 종료
선택하세요 : 1

학생이름, 반, java, oracle, jsp 점수를 입력하세요
홍길동
j
90
80
87
```

```
1. 성적 입력
2. 전체학생의 성적 조회
3. 학생별 성적 조회
4. 클래스(반)별 성적 조회
5. 학생 성적 수정
6. 종료
선택하세요 : 2

=====================전체 학생 성적 조회 결과==========================
학생이름        클래스  java    oracle  jsp     총점    평균    학점
=======================================================================
홍길동          j       90      88      77      255     85.0    B
김길동          j       75      98      84      257     85.67   B
=======================================================================
```

```
1. 성적 입력
2. 전체학생의 성적 조회
3. 학생별 성적 조회
4. 클래스(반)별 성적 조회
5. 학생 성적 수정
6. 종료
선택하세요 : 3

검색하려는 학생이름을 입력하세요
홍길동

====================홍길동 학생 성적 조회 결과==========================
학생이름        클래스  java    oracle  jsp     총점    평균    학점
==========================================================================
홍길동          j       90      88      77      255     85.0    B
==========================================================================
```

```
선택하세요 : 4

검색하려는 클래스(반)를 입력하세요
j

====================j 클래스 성적 조회 결과==========================
학생이름        클래스  java    oracle  jsp     총점    평균    학점
==========================================================================
홍길동          j       90      88      77      255     85.0    B
김길동          j       75      98      84      257     85.67   B
==========================================================================
```

```
선택하세요 : 5

클래스(반)와 학생이름을 입력하세요
j
김길동

수정하려는 점수를 java, oracle, jsp과목 순으로 입력하세요
90
80
70
```