



# JAVA

GU, DA HAE

클 래 스 패 스  
패 키 지

J A V A

객

체

# 클래스 패스

Class Path

클래스 패스는 클래스가 존재하는 경로를 뜻한다.

자바 가상 머신(JVM)은 필요한 클래스를 찾을 때 클래스 패스를 기준으로 찾는다.

아무런 설정을 하지 않으면 JVM은 자신이 실행된 디렉터리에서만 클래스를 찾는다.

자바는 classpath라는 환경변수를 정의하여 클래스 패스를 사용한다.

[환경 변수 설정 방법]

```
> set classpath=.
```

```
> set classpath=.;D:\Wfolder1;
```

위의 방법은 해당 cmd에만 적용된다.

# JAVA

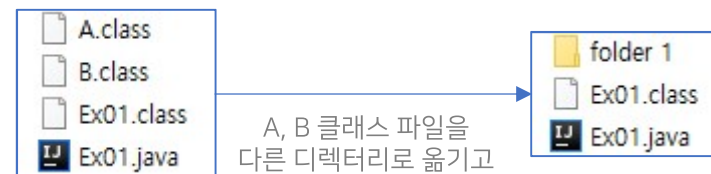
## 예제

```
class A{
    void printA() {
        System.out.println("A!!");
    }
}

class B{
    void printB(){
        System.out.println("B!!");
    }
}
```

```
public class Ex01{
    public static void main(String[] args){
        A a = new A();
        B b = new B();

        a.printA();
        b.printB();
    }
}
```



A, B 클래스 파일을  
다른 디렉터리로 옮기고  
다시 실행해 보자.

# 클래스 패스 설정

Class Path

자바는 classpath라는 환경변수를 정의하여 클래스 패스를 찾는다.

[환경 변수 설정 방법]

```
> set classpath=.
```

```
> set classpath=.;D:\Wfolder1;
```

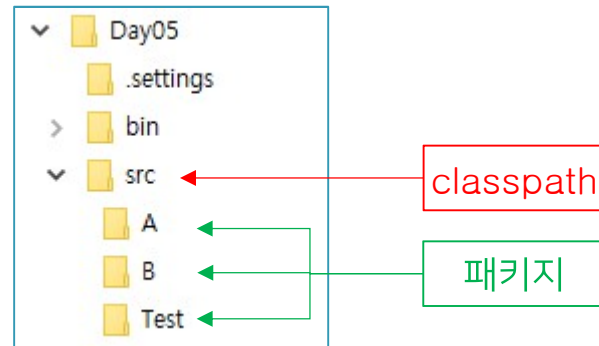
위의 방법은 해당 cmd에만 적용된다.

# 패키지

Package

classpath로 지정한 디렉터리의 **서브 디렉터리**를 패키지라 부른다.

자바의 **클래스 파일**은 반드시 특정 **패키지**에 포함되어야 한다.



[패키지의 클래스 생성 방법]

```
패키지명.클래스명 식별자 = new 패키지명.클래스명();
```

## 예제

```
class Circle{ // 원 둘레 계산 클래스
    double radius;
    double PI;
    Circle(double r){
        radius = r;
        PI = 3.141592;
    }
    double getPeri(){
        return 2*PI*radius;
    }
}
```

```
class Circle{ // 원 넓이 계산 클래스
    double radius;
    double PI;
```

```
    Circle(double r){
        radius = r;
        PI = 3.141592;
    }
    double getArea(){
        return PI*radius*radius;
    }
}

public class Ex01{
    public static void main(String[] args){
        Circle c = new Circle(2.5);
        System.out.println("원 둘레 : " +
            c.getPeri());
    }
}
```

# JAVA

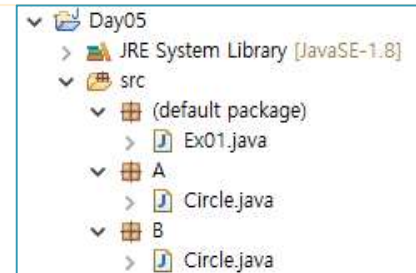
## 예제

package A;

```
public class Circle {  
    double radius;  
    double PI;  
  
    public Circle(double r){  
        radius = r;  
        PI = 3.141592;  
    }  
  
    public double getPeri(){  
        return 2*PI*radius;  
    }  
}
```

package B;

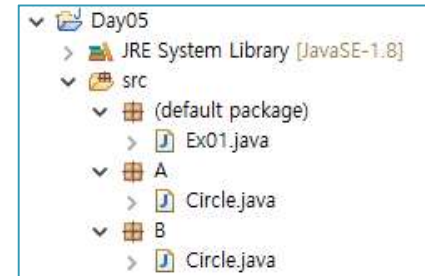
```
public class Circle {  
    double radius;  
    double PI;  
  
    public Circle(double r){  
        radius = r;  
        PI = 3.141592;  
    }  
  
    public double getArea(){  
        return PI*radius*radius;  
    }  
}
```





## 예제

```
public class Ex01{  
    public static void main(String[] args){  
        A.Circle c1 = new A.Circle(2.5);  
        System.out.println("Circle Perimeter : " + c1.getPeri());  
  
        B.Circle c2 = new B.Circle(3.8);  
        System.out.println("Circle Area : " + c2.getArea());  
    }  
}
```



클래스는 반드시 특정 패키지에 포함되어야 한다.  
그런데 Ex01의 경우 패키지를 지정하지 않고 선언했다.  
패키지를 지정하지 않으면 '이름 없는 패키지'로 묶이게 된다.

# import

패키지 경로명을 생략하기 위해 사용한다.

```
import 패키지경로.클래스명;  
import 패키지경로.*;
```

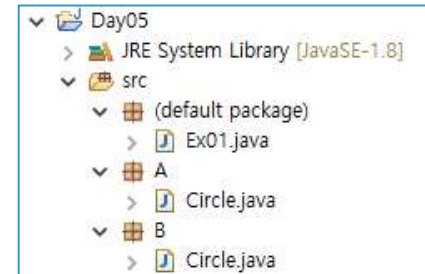
J A V A

## 예제

```
import A.Circle;

public class Ex01{
    public static void main(String[] args){
        Circle c1 = new Circle(2.5);
        System.out.println("Circle Perimeter : " + c1.getPeri());

        B.Circle c2 = new B.Circle(3.8);
        System.out.println("Circle Area : " + c2.getArea());
    }
}
```



J A V A

## 예제

```
import A.Circle;

import B.Circle;

public class Ex01{

    public static void main(String[] args){

        Circle c1 = new Circle(2.5);

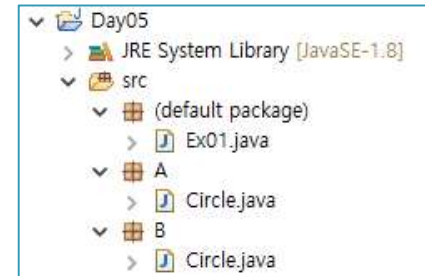
        System.out.println("Circle Perimeter : " + c1.getPeri());

        Circle c2 = new Circle(3.8);

        System.out.println("Circle Area : " + c2.getArea());

    }

}
```



ERROR