

나는 정말
JAVA를
공부한 적이 있나?

Chapter 08. 클래스 패스와 패키지

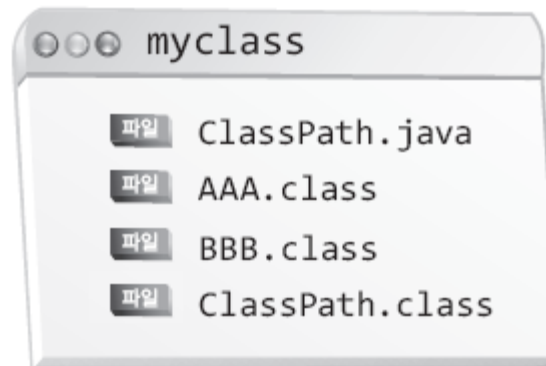


08-1. 클래스 패스의 지정

■ 실행위치와 클래스 파일의 위치

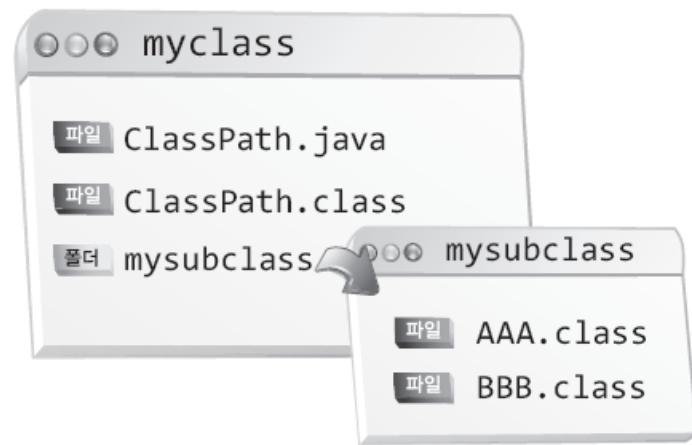
- main 메소드는 ClassPath.class에 존재한다.
- 실행의 형태는 다음과 같다.
C:\myclass> java ClassPath
- 정상적인 실행결과 확인 가능

C:\myclass



- AAA.class, BBB.class를 이동시켜서 실행
- 실행의 형태는 다음과 같다.
C:\myclass> java ClassPath
- 클래스를 찾지 못해서 에러 발생!
- AAA.class, BBB.class를 찾지 못한다.

C:\myclass



mysubclass에서 클래스 파일을 검색하라는 정보를 전달할 수 있어야 한다!

■ 환경변수에 대한 이해

```
C:\myclass> calc.exe
c:\myclass> explorer.exe
```

환경변수 path의 정보에
calc.exe와 explorer.exe의 디렉터리 정보가
등록되어 있기 때문에 실행가능하다.

환경변수 path 확인 명령문

```
C:\myclass> echo %path%
.;C:\WINDOWS\system32;C:\Program Files\Java\jdk1.6.0_10\bin;
```

출력결과



다음 세 경로에서 실행파일을 찾게 된다.

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

■ 환경변수 classpath 추가하기

- 환경변수 path는 확장자가 exe인 실행 파일을 찾는 경로
- 환경변수 classpath는 확장자가 class인 클래스 파일을 찾는 경로

classpath의 설정 및 확인방법

C:\myclass> echo %classpath% classpath의 확인

%classpath% classpath가 설정되지 않은 상태의 출력

C:\myclass>set classpath=.; 현재 디렉터리를 classpath에 등록

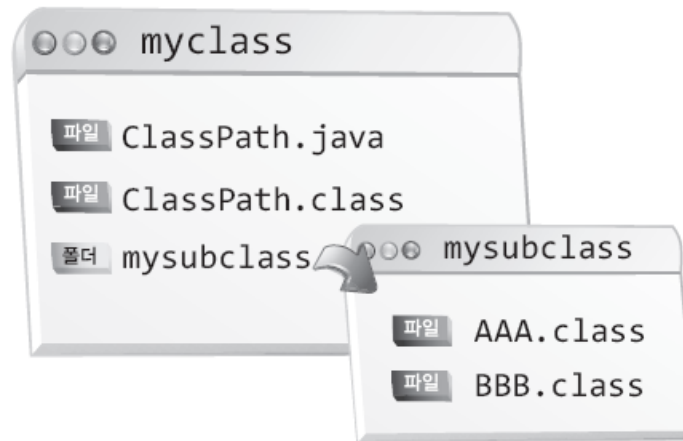
C:\myclass>echo %classpath%

.;

■ 문제의 해결을 위한 클래스 패스의 설정

C:\myclass

C:\myclass> java ClassPath

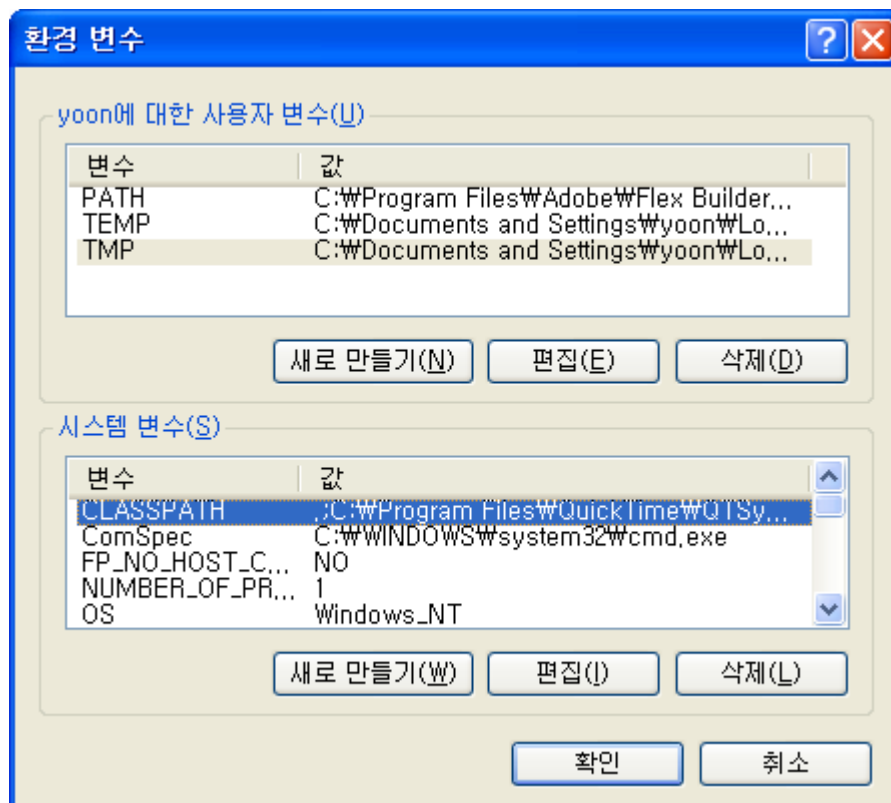


실행을 위한 classpath의 지정

C:\myclass> set classpath=.;C:\myclass\mysubclass; 절대경로 지정방식

C:\myclass> set classpath=.;.\mysubclass; 상대경로 지정방식

■ classpath 추가의 또 다른 방법



실무적인 접근방법이 아닌,
단지 자바 학습에만 사용 가능한
방법임을 기억하자!

- 지금 당장은 매우 편한 방법이다.
- 그러나 classpath의 추가를 자동화할 수 없다.
- 100대의 컴퓨터에 프로그램을 설치해야 한다면?

■ 배치파일 경험하기

배치파일 시나리오

1. 예제 ClassPath.java를 컴파일
2. 그림 8-2의 구조로 실행하기 위해서 mysubclass 디렉터리 생성
3. AAA.class와 BBB.class를 mysubclass 디렉터리로 이동(복사)
4. mysubclass 디렉터리를 classpath로 추가
5. 실행



확장자가 bat인 배치파일

배치파일 실행 결과

```
javac ClassPath.java
md mysubclass
copy AAA.class .\mysubclass\AAA.class
copy BBB.class .\mysubclass\BBB.class
del AAA.class
del BBB.class
set classpath=.;.\mysubclass
java ClassPath
pause
```

```
C:\myclass>javac ClassPath.java
C:\myclass>md mysubclass
C:\myclass>copy AAA.class .\mysubclass\AAA.class
1개 파일이 복사되었습니다.
C:\myclass>copy BBB.class .\mysubclass\BBB.class
1개 파일이 복사되었습니다.
C:\myclass>del AAA.class
C:\myclass>del BBB.class
C:\myclass>set classpath=.;.\mysubclass
C:\myclass>java ClassPath
AAA
BBB
C:\myclass>pause
계속하려면 아무 키나 누르십시오 . . .
```




08-2. 패키지의 이해

■ 문제의 도입: 팀장 회의

- perimeter팀장 : A사에 원의 넓이 및 둘레를 구하는 기능의 클래스를 제공하기로 하였습니다.
- area팀장 : 음! A사로부터 클래스에 대한 세부 요구사항이 있었나요?
- perimeter팀장 : 아뇨, 개발자 위주의 사용 편의성과 활용을 위한 문서화만 요구하였습니다.
- area팀장 : 그럼! 둘레를 구하는 기능의 클래스는 perimeter팀에서 개발을 하세요. 넓이를 구하는 기능의 클래스는 저희 팀에서 개발을 하겠습니다.
- perimeter팀장 : 그럼 두 개의 클래스를 디자인하자는 뜻인가요?
- area팀장 : 네! 아무래도 우리 팀의 개발자는 넓이에 강하고, perimeter팀의 개발자는 둘레에 강하니, 그렇게 개발을 하는 것이 좋을 듯 합니다.

둘레는 perimeter 팀이! 넓이는 area 팀이!

■ 문제의 발견: 클래스 디자인

perimeter 팀의 결과물

```
class Circle
{
    double rad;
    final double PI;

    public Circle(double r)
    {
        rad=r;
        PI=3.14;
    }

    public double getPerimeter()
    {
        return (rad*2)*PI;
    }
}
```

area 팀의 결과물

```
class Circle
{
    double rad;
    final double PI;

    public Circle(double r)
    {
        rad=r;
        PI=3.14;
    }

    public double getArea()
    {
        return (rad*rad)*PI;
    }
}
```

클래스의 이름이 중복되었다!

따라서 아래에서 보이듯이 두 클래스의 구분이 불가능!

```
Circle c1 = new Circle(1.5); // 넓이를 위한 Circle 인스턴스 생성
Circle c2 = new Circle(2.5); // 둘레를 위한 Circle 인스턴스 생성
```

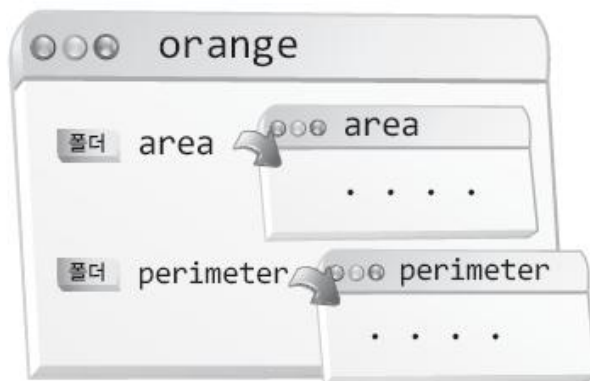
디렉터리를 달리해도 불가능!

■ 문제의 해결: 해결방안의 모색

이것을 가능하게 하기 위한 문법이 패키지이다!

- 클래스 파일을 저장할 디렉토리를 달리한다.
- 클래스의 디렉토리 위치를 지정해서 인스턴스를 생성한다.

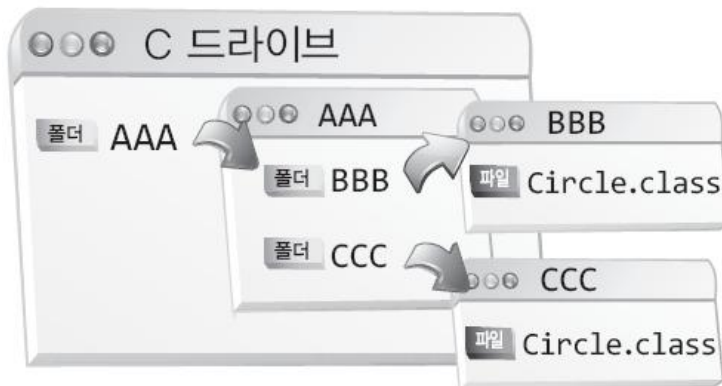
1. 클래스 파일의 저장 위치를 구분



2. 클래스 파일의 저장 위치를 구분

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

■ 패키지과 클래스패스의 관계



패키지도 클래스패스를 기준으로 검색

BBB와 CCC는 패키지로 선언되어야 함.

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

C:\AAA 디렉터리가 클래스 패스에 등록되어 있어야 한다.

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 클래스 패스에 등록되어 있어야 한다.

■ 패키지의 선언

“이 클래스는 orange.area 패키지에 묶겠다!”



이 클래스를 orange\area 디렉터리에 저장하고,
이 경로를 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());
```

■ import 선언

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



의미하는 바

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



import 선언 이후의 두 가지 인스턴스 생성방법

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

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



의미하는 바

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

모호함이 발생하는 대표적인 사례

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

