클래스패스와 환경변수

**클래스 이름과 패키지**

다른 사람이 만든 같은 이름을 가진 .class파일(.java파일을 컴파일한) 들은 패키지로 구별 가능

회사이름으로 구분

전세계 유일한 이름을 클래스 이름 앞에 붙여준다.(도메인 명)

[www.gabdol.com](http://www.gabdol.com)이라면 모든 클래스의 이름을 com.gabdol.A와 같은 식으로 만들면 된다.

com.gabdol이 클래스 A의 패키지 이름이 되는 것

만약, 갑돌이네 회사가 두개의 자바 소프트웨어 개발을 했는데, 둘 다 A.java라는 클래스가 있다면 이 또한 구분 🡪 com.gabdol두에 임의의 회사가 알아서 패키지 이름을 만들면 된다. 예로 소프트웨어 이름이 sw1, sw2라면 com.gabdol.sw1.A, com.gabdol.sw2.A식으로 클래스 이름 사용

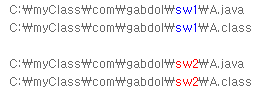
패키지 사용

1. 이름 충돌 문제 해결
2. 클래스 분류도 쉬움  
   예로, 고객 관련 클래스들의 패키지 이름 : com.gabdol.client  
   상품 관련 클래스들의 패키지 이름 : com.gabdol.product으로 정하면 클래슫르이 성격을 패키지 이름으로도 쉽게 파악 가능

**클래스의 위치**

패키지 이름에 따라 클래스의 위치가 유일하게 구별될 수 있어야 한다.

폴더의 위치를 다르게 두면 클래스 충돌 X



**패키지 선언과 임포트(package or import)**

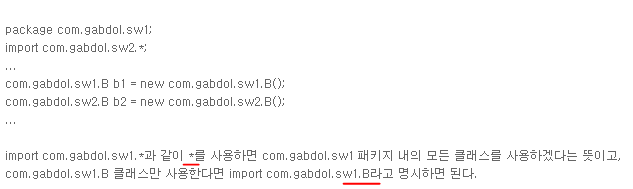
com.gabdol.sw1.A의 클래스의 패키지 : com.gabdol.sw1

이 정보는 당연히 소스 코드 첫 머리에 package com.gabdol.sw1; 으로 쓰면 된다.

다른 패키지에 있는 클래스 사용

1. 클래스의 이름 모두 써줌  
   com.gabdol.sw2.B b = new com.gabdol.sw2.B();  
   🡪 클래스 B가 com.gabdol.sw2.패키지 내에 있는 것이 드러나므로 컴파일 시 문제 X
2. 패키지를 임포트(import)하는 것  
   package com.gabdol.sw1;  
   import com.gabdol.sw2.\*;

B b = new B();

// com.gabdol.sw1.A클래스가 com.gabdol.sw2 패키지 내의 클래스들을 사욯하겠다는 뜻(이렇게 하면 자바 컴파일러는 com.gabdol.sw1패키지와 임포트한 com.gabdol.sw2 패키지 내에서 찾게 된다. 만일에 모두 B라는 클래스를 가지고 있다면 컴파일러는 어떤 걸 쓸 줄 모르므로 에러 발생)   


**클래스패스**

갑돌이가 C:\myClass\com\gabdol\sw1\A.java파일을 컴파일 하려 한다.

sw1폴더까지 이동 후에 javac A.java 시도

* 당연히 클래스 B를 찾을 수 없다는 에러 메시지 출력

com.gabdol.sw1.B클래스가 컴파일된 바이트코드인 B.class가 어디에 있는 지 자바 컴파일러는 알 길이 없기 때문이다. 🡪 이 때 필요한 것이 클래스패스 이다.

갑돌이가 다음 처럼 하면 컴파일이 성공적으로 이루어진다.

javac –classpath C:\myClass A.java

여기에 –classpath C:\myClass 부분 : 자바 컴파일러에게 C:\myClass폴더를 기준으로 패키지를 찾아라는 뜻

즉, com.gabdol.sw1.B클래스는 C:\myClass 폴더를 시자긍로 C:\myClass\com\gabdol\sw1폴더에서 찾고,

com.gabdol.sw2.B클래스는 C:\myClass\com\gabdol\sw2 폴더에서 찾는 것

(예시)

갑돌이는 언정이에게 건네 받은 클래스들을 C:\otherClass라는 폴더에 저장하고 있었는데, 이 중에 일부 클래스를 A.java에서 사용할 일 발생

언정이가 건네 준 클랫느느 com.eonjeong.lee이라는 패키지에 포함되어 있고, 이 클래스들은 C:\otherClass\other\com\eonjeong\lee 폴더에 있다면, 언정이가 준 클래스들은 C:\otherClass\other 폴더를 시작으로 찾아야 한다!!!

따라서, 언정이의 클래스를 사용하는 A.java파일을 컴파일 하기 위해서는..

javac –classpath “C:\myClass;C:\otherClass\other” A.java

이렇게 하면 C:\myClass 폴더에 저장되어 있는 com.gabdol.sw1.A, com.gabdol.sw2.B클래스와 C:\otherClass\other 폴더에 저장되어 있는 com.eonjeong.lee.\* 클래스들을 자바 컴파일러가 제대로 찾을 수 있게 되는 것이다!

A.java를 컴파일해서 얻어진 클래스를 직접 실행하려면 자바 가상 머신(JVM)을 구동해야 하는데, 이 때에서 당연히 A.class가 사용하는 다른 클래스를 JVM이 찾을 수 있어야 한다!! 따라서, 역시 컴파일할 때와 마찬가지로 클래스패스가 필요하다. A.class를 실행하려면 다음처럼 해야한다.

java –classpath “C:\myClass; C:\otherClass\other” com.gabdol.sw1.A

실행할 떄에는 실행하고자 하는 클래스의 패키지 이름까지 명시해야 한다.(com.gabdol.sw1.A)  
왜냐하면, JVM이 클래스를 실행할 떄에는 지정된 이름이 패키지이름을 포함한 것으로 생각하기 때문이다. 만일, 다음과 같이 java –classpath “C:\myClass; C:\otherClass\other” A 실행하면 JVM은 C:\myClass\A.class 혹은 C:\otherClass\other\A.class를 찾으려 한다. 클래스패스란 여러 폴더에 산재한 클래스들의 위치를 지정해서 패키지에 따라 클래스를 제대로 찾을 수 있게 해주는 값이다.

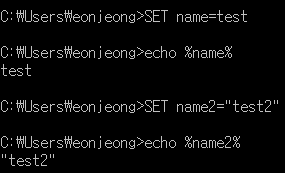
**환경변수**

매번, 컴파일할 떄와 실행할 때에 classpath를 옵션에 클래스패스를 명시하는 것은 불편하다.

한번의 설정으로 이런 문제를 해결하기 위해서 환경변수를 사용

환경변수 : 말 그대로 변수에 어떤 값을 저장시키되 이를 환경에 저장하는 것

환경에 저장 : 그 환경 내의 모든 프로그램이 그 변수의 값을 알 수 있게 된다는 것



다른 명령창에다 하면

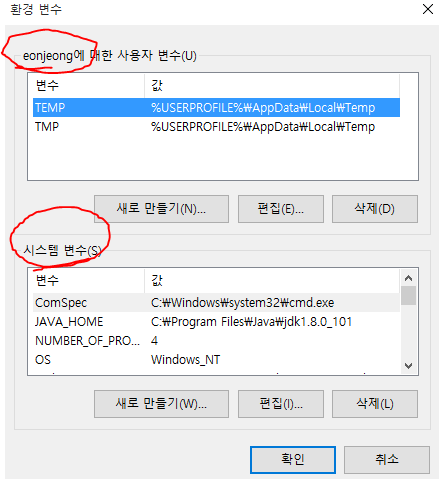
C:\> echo %myname%

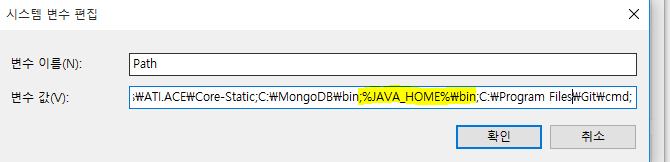
%myname%

오로지, 변수를 설정한 그 명령창 내에서만 그 변수는 의미가 있게 되는 것

온갖 프로그램에서 공유하기 위해서는 다른 방법 사용

윈도우면 고급/환경변수 메뉴



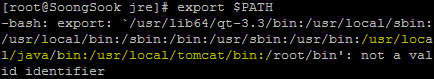


위 쪽은 특정 로그인 사용자에 대한 호나경변수 설정

아래 쪽은 모든 사용자에게 적용되는 호나경 변수

관리자(Administrator) 권한을 갖고 있다면 아래 시스템 변수의 값을 설정 가능

(리눅스)



현재의 패스가 걸린 디렉토리 전부 출력

export 명령어 : 전체 시스템상의 설정들 출력해 볼 수 있다.

패스가 걸리지 않은 디렉토리에 있는 파일 실행시키려면 절대 디렉토리 적어주어 실행 혹은 디렉토리 파일이 있다면 ./를 붙여서 ./xxx라고 실행

만일 패스를 추가해주고 싶다면 자신의 홈디렉토리의 .bash\_profile을 에디트하면 된다.

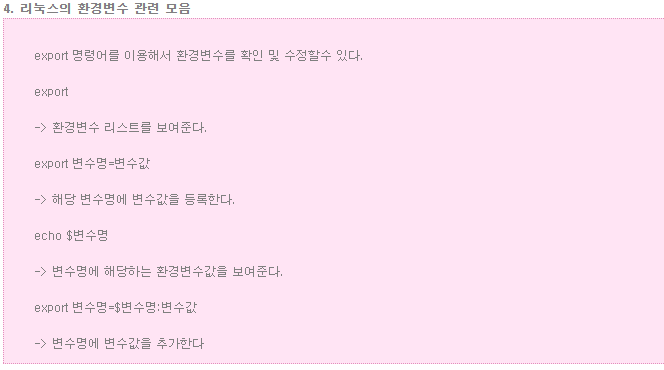
vi ~/.bash\_profile

‘PATH=$PATH:$HOME/bin’부분에서 콜론(:)으로 구분하여 덧붙이면 된다.

예로 /usr/local/bin을 추가하고 싶다면

PATH=$PATH:$HOME/bin:/usr/local/bin 이렇게 해주고 저장을 하고 나온 후

source ~/.bash\_profile이라고 치면 설정이 바로 적용된다.



jar파일은 클래스들을 묶어 놓은 것

갑돌이는 자신이 만든 com.gabdol이하의 모든 클래스를 하나의 파일인 gabdol.jar파일로 묶을 수 있다.

이렇게 gabdol.jar팡리을 C:\gg\cc 폴더 아래에 두었다면 이 압축 파일 내의 클래스를 역시 자바 컴파일러와 JVM이 찾을 수 있도록 해야 한다.

이때는 파일 이름까지 포함해서 클래스패스를 추가해야 한다.

set CLASSPATH=C:\gg\cc\gabdol.jar;C:\myClass;C:\otherClass\other