**[ 10 ] 패키지(package)**

1. 패키지의 이해

패키지란?

프로그래밍에서 여러 클래스를 관리하기 위해 기능적으로 영향을 미칠 수 있는 클래스끼리 묶어 놓고, 접근 범위 안에 효과적으로 호출하기 위해서 사용하는 개념이다.

윈도우에는 윈도우탐색기(window+E)가 있다. 윈도우 탐색기는 컴퓨터에 보관되어 있는 파일을 관리하기 위해 폴더를 만들고, 그러한 폴더를 효율적으로 다루기 위해 존재한다. 즉 폴더 구조를 한눈에 볼 수 있도록 만들어진 프로그램이다.

JAVA에도 많은 ~.java파일들이 존재 합니다. 많은 정도가 예상하기도 힘들 정도로 많이 존재합니다. 또한 우리가 프로그래밍을 하면서 계속적으로 .java파일을 만들 것입니다. 이러한 많은 파일을 효율적으로 관리하기 위해 JAVA에서는 ‘패키지’라는 구조를 사용 합니다.

우선 JAVA(jdk)에 존재하는 패키지를 살펴 보도록 하겠습니다. 이미 만들어 놓은 기능들입니다.

1. 아래 주소접속하여 자바(jdk)에서 제공하는 클래스(3,000개정도)들이 어떤 것들이 있는지 봅시다.
2. <http://www.oracle.com/technetwork/java/javase/documentation/jdk8-doc-downloads-2133158.html> 에 접속하여 다운받거나 온라인으로 <http://www.oracle.com/technetwork/java/api-141528.html> (<https://docs.oracle.com/javase/8/docs/api/index.html>)에 들어가 API들을 볼 수 있다.
3. JAVA API(Application Programming Interface) : 자바에서 기본적으로 제공되는 클래스의 설명들을 볼 수 있다. 패키지를 모르면 "All Classes"를 선택하여 쉽게 찾아볼 수 있다. 자바 사전이라고 생각하고, 자주 보는 습관을 갖자.
4. 모든 클래스들을 패키지로 관리되어 있는 것을 볼 수 있다. 해당 패키지에는 해당클래스를 볼 수 있습니다.
5. 패키지 : **관련있는 클래스들을 패키지로 그룹지어 관리하는 것(클래스를 모아 놓은 폴더)이라는 것**을 확인할 수 있습니다.

ex. Scanner 클래스는 java/util폴더에 있는 Scanner클래스

(java.util.Scanner)

ex. Math클래스는 java/lang폴더에 있는 Math클래스 : java.lang 패키지의 클래스는 import 생략.

(java.lang.Math)

1. 우리가 만든 패키지

JAVA(jdk)에 패키지가 있듯이 우리도 클래스를 만들 때 패키지로 관리하면 효율적인 작업을 할 수 있습니다.

이제 직접 패키지를 만들어 보도록 하겠습니다.

☞ 실제 폴더 구조를 보도록 합시다

2. 정답은 없지만, 효율적으로 해야 하는 패키지의 일반적인 설계방식

① 초보 개발자 뿐 만이 아니라, 오래된 개발자의 경우에도 패키지를 만들 때 패키지 이름에 대해서 다소 고민을 하는 경향이 있습니다. 패키지 이름만 봐도 어떤 클래스들이 모여있는지 알 수 있도록 만들고 싶은데 그게 쉽지 않은 것이죠. 패키지 이름은 유일한 이름이여야 합니다. 이유는 우리회사에서 개발은 소스(.java파일)를 다른 곳에 이식하려 할 때 이름이 중복되면 되기 때문입니다. 그래서 주로 패키지 이름은 유니크한 도메인 주소를 역방향으로 하여 만듭니다. 예를 들어 회사 도메인이 www.tj.com 이라면, ‘com.tj.\*’라고 하고 뒤에 기능별 또는 개발팀에서 정한 규칙에 의해 패키지명을 만듭니다.

② 물론 패키지명은 개발 중 언제든지 수정이 가능, 패키지 간의 클래스 이동도 가능. 따라서 패키지명을 만들 때 너무 스트레스를 받지 말고, 쉽게 쉽게 생각하면 될 거 같습니다.

③ 명명규칙은 변수와 같이 숫자로 시작할 수 없고, \_나 $를 제외한 특수문자를 사용할 수 없다. 모두 소문자로 작성하는 것이 일반적이다.

3. import의 이해 : 다른 패키지의 클래스를 가져오자~

패키지를 만들고 클래스를 만든 후 서로 다른 패키지의 클래스를 사용하려고 할 때는 반드시 import 키워드를 사용해서 해당 클래스를 임포트 해야 합니다. 하지만 우리가 사용하는 이클립스는 자동으로 임포트되도록 지원합니다.

(cf) 동일한 패키지의 클래스는 임포트가 필요없음.

(cf) import com.tj.ex.\*; //com.tj.ex.\* 패키지 안에 있는 모든 것. 패키지내에 너무 많은 클래스가 있을 경우엔 메모리 환경에 안좋은 영향을 미치겠지만 요즘은 메모리 환경이 좋아 이런 방식을 많이 사용합니다.

<예제 : 학사관리>

/\* 학사관리 프로그램을 구현합니다

* 필요한 클래스를 머릿속에 생각해 보자
* 우선 main메소드가 있는 학사관리 시스템 클래스가 있고, 학생을관리하는 시스템이므로 학생에 대한 클래스가 있어야 한다. 그리고 학생클래스에는 학생의 학번, 이름, 나이, 전공 데이터를 포함하고, 데이터는 언제든지 수정할 수 있도록 한다.
  + StudentTestMain
  + Student – (데이터) 학번, 이름, 나이, 학과

(메소드) modify(기능번호,바꿀데이터) //변경기능 1은 학번, 2는 이름, 3는 나이, 4는 학과

infoString() //출력할 스트링 리턴 \*/

**package** com.tj.ex3\_student;

// (데이터) 학번, 이름, 나이, 학과

// (메소드) modify(fn, data) //변경기능 1은 학번, 2는 이름, 3는나이, 4는 학과

// infoString()