**접근제어 및 패키지**

|  |  |
| --- | --- |
| 1 | 다음 설명 중 틀린 것은?  (1) 클래스안에 선언된 멤버변수들은 데이터를 담는 역할을 하므로, 업무의 특성상 보안이 중요한 경우  다른 객체의 접근을 적절히 제한할 방법이 필요하다.  (2) 자바의 접근제어는 총 4가지가 있으며 보안 등급이 가장 높은 접근제어자는 private 이다.  (3) 모든 접근을 허용하는 접근 제어자는 public 이다.  (4) 개발자가 아무것도 접근 제어를 명시하지 않을 경우 기본적으로 적용되는 접근제어는 default 이다.  - default : 같은 디렉토리(패키지)에 있는 것만 접근 가능.  (5) 접근제어자를 보안이 낮은 순으로 나열하면 public < default < protected < private 이다.  - public < protected < default < private |
| 2 | 다음 설명 중 틀린 것은?  (1) 같은 디렉토리내의 파일명은 OS차원에서도 중복을 허용하지 않으므로 클래스를 하나의 디렉토리에  계속 누적해 나가다보면 클래스 파일명의 중복 문제가 발생하게 된다.  (2) 클래스명의 중복을 원천적으로 방지하기 위해서는 서로 다른 디렉토리로 구분하여 저장하면 된다.  (3) 클래스들을 디렉토리 단위로 구분하여 놓되, 관련성 있는 클래스들끼리 모아놓는 방법이 더 효율적일  것이다. 자바의 이러한 디렉토리 분류 방법을 패키지라 한다.  (4) 클래스 파일명이 동일하더라도 패키지가 다를 경우엔 중복의 문제는 발생하지 않는다.  (5) 서로 다른 디렉토리에 존재하는 클래스간의 접근은 불가능하다. – protected 이상이면 가능. |
| 3 | 다음 설명 중 틀린 것은?  (1) 패키지를 사용하게 되면 파일명의 중복도 피할 수 있을 뿐 아니라, 관련성 있는 클래스끼리 분류도 할  수 도 있다.  (2) 사용하고자 하는 클래스가 다른 디렉토리에 있을 경우 그 위치를 알아야 사용이 가능하므로, 개발자는  프로그래밍시 해당 클래스의 위치를 지정해야 한다.  (3) 다른 디렉토리에 있는 클래스의 경로를 표현하기 위해서 개발자는 OS에서 지원하는 절대경로를 사용  할 수 있으며, 윈도우 일 경우 D:/word/Dog.class 와 같이지 표현해야 한다.  (4) 위(3)번은 자바의 슬로건이 가장 잘 반영한 개발 방법이다.  (5) 자바가 OS에 종속적이지 않으려면 특정 클래스의 위치를 표현하는 방법은 OS의 특성을 타지 않는  방식으로 표현해야 할 것이다. |
| 4 | 아래 클래스에 대한 설명 중 올바르지 않은 것은?  package pack1;  class Car{  int price=300;  }  (1) javac –d 옵션으로 컴파일을 수행하면 pack1이라는 패키지가 자동으로 생성되면서 그 안에 Car.cla  ss를 생성하여 준다.  (2) 만일 이미 pack1 이라는 패키지가 존재한다면 중복 생성되지 않는다.  (3) 이 클래스는 외부에서 절대로 접근이 불가능한 클래스이다.- public 을 써줘야 쓸 수 있다.  (4) price 멤버변수에 접근 제한자가 지정되지 않으면 자동으로 private 으로 지정된 것과 같다.-default 접근제한다 기 본 |
| 5 | 다음의 빈칸에 알맞는 말을 채우세요.  환경변수 중 ( path )는 일반 파일들을 쉽게 접근하기 위해 사용되는 반면, 환경변수 중 ( classpath ) 는  일반 파일이 아닌 ( class )파일의 경로를 지정하기 위한 방법에 사용된다. |
| 6 | 위 4번의 클래스가 실제 하드디스크 D:/myjava/pack1 에 들어있고 클래스 패스에 d:/myjava까지 등록  한 상태라고 가정할 때 Car 클래스에 대한 import 문의 올바른 표기 방법은?  (1) import D:\ minjava  (2) import minjava \pack1  (3) import pack1;  (4) import D:\minjava \pack1\Car  (5) import pack1.Car  (6) import pack1.\* => 가능은 하나 이렇게 쓰지 말자. |
| 7 | 다음 설명 중 올바르지 않은 것은?  (1) classpath 환경 변수의 이름은 이미 정해진 변수명이므로, 이 변수명 이외의 다른 이름은 사용할 수  없다.  (2) 패키지명은 소문자로 하되, 주로 개발 밴더사의 도메인을 거꾸로 기재하여 배포한다.  (3) 위(2)번과 같이 하는 이유는 패키지명의 중복을 피하고 유일성을 확보하기 위함이다.  (4) 패키지안에 들어있는 클래스에 대해 보안을 유지하려면 클래스 선언시 private으로 지정해야 한다.-클래스가 아닌 변수에 지정한다. |
| 8 | 다음 설명 중 틀린 것은?  (1)자바언어에서는 클래스, 멤버변수, 메서드에 접근 제한을 지정할 수 있다.  (2)자바의 접근 제한자를 보안이 강한 순서데로 나열하면 private > default > protected > public  순이다.  (3) 접근제한자를 지정하지 않을 경우 기본 접근제한자인 default 접근제한자가 적용되며, 같은 패키지내  에서의 접근만을 허용한다.  (4)대부분은 멤버변수의 접근제한을 public 으로 지정하며, 멤버메서드는 private 으로 보안을 유지한다.-반대 |
| 7 | 다음 설명 중 틀린 것은?  (1) 클래스의 멤버변수는 결국 데이터를 보관하는 역할을 하므로 보안상 중요한 경우 private 으로 접근  을 제한한다.  (2) private 으로 선언된 멤버 변수는 같은 인스턴스내의 멤버메서드라 할지라도, 접근이 불가능하다.-접근가능  (3) private 으로 멤버 변수를 묶어두면 외부의 클래스에서 변수에 대해 접근 자체가 불가능해 지며,  이렇게 변수 자체를 사용 못하게 하는 것이 자바 보안의 목적이다.-사용하는게 목적이다.  (4) (3)번의 경우처럼 변수 자체에 대한 사용을 금지시킬 경우, 클래스를 정의한 목적 자체가 무의미해  지므로, 변수 사용에 대한 접근 허용 여부를 개발자가 임의로 조절할 수 있도록 하기 위한 방법이  필요하다.-은닉화  (5) 위(4)번과 같은 문제를 해결하기 위해, 자바와 같은 객체지향 언어에서는 특정 변수에 대해 1:1로  대응하는 getter 및 setter 메서드를 정의하여 사용함이 일반적이다.  (6) 클래스의 멤버 변수는 private 으로 지정하여 외부의 접근을 막아두고, 이에 대한 읽기, 쓰기 권한을  가진 메서드를 제공하여 데이터의 보안을 유지하는 클래스 정의 기법을 가리켜 은닉화,캡슐화라 한다. |
| 8 | 아래의 클래스를 은닉화 하세요.  class Customer{  String name;  int age;  boolean isMarry;  } |
| 9 | 아래의 클래스를 은닉화 하세요.  class Company{  int memberCount;  String location;  } |