**접근 제어**

|  |  |
| --- | --- |
| 1 | 다음 설명 중 틀린 것은?  (1)~~개발자가 다루려는 대상 클래스가 보안상 중요한 경우라 할지라도, 이 클래스의 멤버 변수는 다른~~  ~~외부의 클래스들이 접근할 수 있도록 무조건 허용해줘야 한다는 것이 자바 언어 개발시 권장 사항이다.~~  (2) 자바는 private, protected, default, public 이라는 접근 제한자를 지원한다  (3) public 으로 선언된 변수, 메서드, 클래스는 제한없이 외부의 클래스가 언제든지 접근이 가능하다  (4) default 접근 제한자는 같은 디렉토리에 있는 클래스들간에만 접근을 허용하며, ~~개발자가 클래스 앞에~~  ~~default 라는 키워드를 명시하여 선언해 줘야 한다.~~ (명시시 오류)  (5) ~~protected로 선언된 경우, 상속 관계의 객체들만 접근이 가능하다~~ (자식 및 같은 패키지도 가능)  (6) public은 모두에게 제한없이 공개된다. public을 적용하면 보안을 적용하지 않겠다는 의미이다  (7) 접근 제한자는 클래스, 메서드, 변수에 적용할 수 있다. |
| 2 | 다음은 java의 접근 제한자를 보안 수준 순으로 나열한 것입니다. 빈 칸을 채우세요  public < ( protected ) < ( default ) < private |
| 3 | 다음 중 틀린 것은?  (1) 클래스내에 선언된 멤버변수가 중요한 내용일 경우도 있으므로, 일반적으로 멤버변수는 보호됨이  원칙이다.  (2) 멤버변수를 private으로 보호하면, 외부의 그 어떠한 클래스도 접근이 불가능하다.  (3) 데이터는 private 보호하되, 외부의 클래스가 이 데이터를 사용할 수 있도록 하려면 메서드는  공개되어 있어야 한다.  (4)(3)번의 이유로 인하여 클래스 작성시 멤버변수는 private 보호하고, 멤버변수마다 1:1 대응되는  getter와 setter 메서드를 제공하는 클래스 정의 기법을 가리켜 OOP에서는 은닉화 또는 캡슐화  (=encapsulation)라고 한다  (5) ~~getter/setter 작성 시, 매개변수의 개수 등은 개발자가 임의로 작성하면 되며, 별도의 명명규칙 같은~~  ~~것은 없다.~~ |
| 4 | 다음은 현실의 ‘계좌’ 를 클래스로 표현한 것이다. 틀린 설명은?  **public** **class** **Account** {  **int** balance=200; //(가)  **String** bank="기업은행";  **String** accountNum="12-456-789-10";    **public** **int** getBalance() { //(나)  **return** balance;  }  **public** **void** setBalance(**int** balance) { //(다)  **this**.balance=balance;  }  }  (1) ~~(가)의 balance 속성은 별도의 접근제한을 지정하지 않았으므로 아무런 접근제한자가 적용되지 않은~~  ~~상태이다.~~  (2) (가) 의 balance 속성은 default 접근제한이 적용되고 있으므로 같은 디렉토리에 있는 특정 클래스의  접근이 허용되고 있다.  (3) balance를 다른 디렉토리에 있는 외부의 어떤 클래스가 참조하려면 직접 접근이 불가능하므로,  (나)의 getter 메서드를 통해 balance값을 얻어갈 수 있을 것이다.  (4) private 으로 멤버 변수를 제한해두면 외부의 클래스가 접근하는 것 자체가 불가능 해지며, 이렇게  변수 자체를 ~~아예 사용 못하게 하는 것이 자바 보안의 목적이다.~~  (5) 캡슐화는 클래스내의 변수에 대한 외부의 직접 접근을 불허하여 데이터를 보호하고자 함이며, 이렇게  보호된 데이터는 오직 개발자가 정의해 놓은 메서드를 통해서만 간접적으로 접근할 수 있도록 한  OOP의 클래스 정의 기법이다. |
| 5 | 아래의 클래스를 은닉화 하세요.  Public class Customer{  Private String name;  Private int age;  Private boolean isMarry;  } |