**오버로딩**

|  |  |
| --- | --- |
| 1 | 다음 설명 중 틀린 것은?  (1)자바에서는 하나의 클래스내에 메서드의 이름을 중복 정의해서는 안된다. X  (2)메서드의 이름은 중복되어도 인수의 자료형과 개수를 달리하면 메서드간 구분이 가능하므로  이러한 코드 작성법은 자바에서 허용된다.O  (3)중복된 메서드명은 하나의 클래스내에 2개까지만 허용된다.X  (4)한 클래스 내에서 중복된 이름를 허용하는 자바의 메서드 정의 기법을 오버로딩(Overloading)이라  한다.O  (5)개발자는 변수나 메서드명을 부여하기 전 상당한 고민을 하여 직관성을 부여한다, 하지만 이렇게  어렵게 지어놓은 메서드명을 중복금지라는 원칙 때문에, 기능상 별차이가 없음에도 불구하고  전혀 다른 이름을 부여하기 위해 고민해야 한다면 프로그램 개발 시 상당히 효율성이 떨어질 것이다  또한 상당히 비슷한 기능을 갖는 메서드들임에도 불구하고, 이름을 달리 정의하게 될 경우, 메서드  이름만으로도 그 수행 내용을 예측할 수 있었던 직관성을 잃게 되어 많은 불편함을 초래할 수 있다.  따라서 오버로딩을 사용하면 개발자는 메서드명의 중복을 피하기 위해 명명작업 때문에 고민하지  않아도 되며, 단지 인수의 자료형과 개수를 달리 하여 기존의 이름을 유지할 수 있다 O |
| 2 | **OCJP 시험문제 - OverLoading**  Given:  1. public class A {  2. public void doit() {  3. }  4. public String doit() {  5. return “a”;  6. }  7. public double doit(int x) {  8. return 1.0;  9. }  10.}  What is the result?  A. An exception is thrown at runtime.  B. Compilation fails because of an error in line 7.  C. Compilation fails because of an error in line 4.O  D. Compilation succeeds and no runtime errors with class A occur. |
| 3 | 아래의 클래스에 대한 설명 중 맞는 것은?  **class** Doll{  **static** String *name*="피카츄";  **int** price;    **public** Doll(**int** p){  price=p;  }  **public** **static** **void** main(String[] args){  Doll.*name*="테디베어"; //(가)  Doll d=**new** Doll(); //(나)  d.*name*="뽀로로"; //(다)  }  }  (1) 이 클래스의 멤버 변수는 2개이다.O  (2) 개발자가 생성자를 정의하지 않았으므로 디폴트 생성자만 존재한다.X  (3) (가) 에서 아직 인형의 인스턴스가 생성되지 못한 상태에서 name변수를 접근하려 했으므로,  컴파일 에러가 발생한다.X  (4) (나) 는 인형을 메모리에 올리는 코드이므로 잘못된 부분이 없다.X  (5) static 으로 선언된 name 변수는 (다) 와 같은 접근방법으로 사용이 불가능하다.X  (6) 이 클래스는 개발자가 생성자를 정의하였으므로, (나)에서 인형의 인스턴스를 생성하려할 때 Doll()  은 존재하지 않는 메서드이므로, 컴파일 에러가 날 것이다.O |
| 4 | 아래의 클래스에 대한 설명 중 맞는 것은?  **class** Movie{  String title;  **int** price;  **public** Movie(**int** p){  price=p;  }  **public** Movie(String t){  title=t;  }  **public** **static** **void** show(){  System.out.println(“movie start!!”);  }  **public** **static** **void** main(String[] args){  Movie mv1=**new** Movie(); //(가)  Movie.*show*(); //(나)  }  }  (1) 이 클래스는 생성자가 2개 정의되었으므로 컴파일 에러가 발생할 것이다.X  (2) (가)에서 Movie 클래스의 인스턴스가 오류없이 생성될 것이다.X  (3) Movie 클래스에는 컴파일러에 의한 디폴트 생성자가 존재한다.X  (4) 생성자 오버로딩이 적용된 클래스이며, 디폴트 생성자는 존재하지 않는다.O  (5) (나)의 Movie.show() 메서드는 멤버메서드이므로 클래스명으로 접근할 수 없다.X |