|  |  |
| --- | --- |
| 1 | 다음 설명 중 맞는 것은?  class Animal{  String name="난 동물";  int color;  public Animal(String color){  this.color=color;  }  public void run(){  System.out.println(“동물이 뛰어요”);  }  }    class Tiger extends Animal{  String name;  public Tiger(String name){  ( ) //(A)  name=name;  }  public void run(){  System.out.println(“호랑이가 뛰어요”);  }  }  class Rabbit extends Animal{  String name=name;  public Rabbit(String name){  this.name=name;  }  public void run(){  System.out.println(“토끼가 뛰어요”);  }  }  class UseAnimal{  public static void main(String[] args){  Tiger tiger=new Tiger(); //(B)  Animal animal=new Tiger(“난 호랑이”);//(C)  animal.run(); //(D)  Rabbit rabbit = new Rabbit(“난 토끼”);  animal = rabbit;  animal.run(); //(E)  }  }  (1) (B) 라인에서 아무런 문제가 없다  (2) (A)라인에 들어갈 적절한 코드는 super() 이다  (3) (C)에서 Tiger 의 인스턴스 생성에 앞서 , 부모인 Animal 클래스의 인스턴스가 먼저 힙에 생성되어질  것이다.  (4) (D)에서 호출되는 메서드는 Animal 클래스의 메서드 이므로 실행 결과는 “동물이 뛰어요” 이다  (5) (E)에서 실행되는 결과는 Rabbit 클래스의 메서드가 호출되므로 “토끼가 뛰어요” 이다.  (6) 이 예제에서 Animal 이라는 자료형은 호랑이, 토끼보다 상위에 존재하는 부모 객체이므로 , 하위 자료형  들을 가리킬 수 있으나, 실제 동작은 부모 클래스의 메서드가 호출된 게 아니라, 하위 객체의 메서드가  호출되고 있으므로, 다형성을 구현한 것이다. |
| 2 | 아래의 괄호에 들어갈 공통 단어는?  생물학, 의학에서는 ( ) 이라는 용어가 있다.  ( )이란 같은 種(종) 일 지라도 서로 틀린 개성과 다양한 모습을 갖는 객체의 특징을 의미하는데, 자바언어 에서의 ( )이란 이와 크게 다르지 않다.  자바에서는 특정 자료형을 갖는 객체가 실제 동작 방식에 있어서는 여러 가지 객체의 모습을 띄고 동작될 수 있는 기능을 지원한다.  좀더 구체적으로는 부모 객체의 데이터 타입으로 자식 객체를 가리킨 후, 실제 메서드 호출은 자식 객체의  메서드를 호출하는 기능을 지원하는데, 이 기법을 이용하면 부모 클래스의 자료형 하나로 하위의 객체들을  모두 가리킬 수 있어 상황에 따라 유연한 코드를 작성할 수 있게 지원한다. 객체지향 언어에서는 이런 특징  을 ( )이라 한다. |