|  |  |
| --- | --- |
| 1 | 현실에 존재하는 대부분의 사물은 단독으로 존재하기 보다는 여러 개의 개별적 객체들이 모여,  하나의 객체를 이루는 경우가 더 일반적이다. 예를 들어 자전거는 바퀴, 핸들, 체인등 개별적 부품으로 이루어져 있고, 심지어 볼펜 조차도 스프링, 볼펜심 ,플라스틱 몸체 등으로 이루어져 있는 셈이다.  OOP에서는 이와 같이 특정 사물들을 부품으로 보유할 수 있는 관계를 ( Has a ) 관계라 한다. |
| 2 | 자동차의 Engine 을 표현하는 클래스를 디자인하세요 |
| 3 | 자동차의 Wheel 을 표현하는 클래스를 디자인 하세요 |
| 4 | 자동차의 Handle 을 표현하는 클래스를 디자인 하세요. |
| 5 | 자동차 몸체를 표현하는 클래스를 디자인 하세요. |
| 6 | 아래 클래스에 대한 설명으로 틀린 것을 모두 고르면?  **class Car{**  **int price;**  **Engine e;**  **Wheel w;**  **Handle h;**  **public Car(int p){**  **price=p;**  **}**  **}**  (1) 엔진, 휠, 핸들을 부품으로 보유한 자동차를 표현하고 있다.  (2) 자바언어는 기본자료형 외에 객체자료형을 지원하므로, Car 클래스의 멤버변수에 기본 자료형인  price 뿐만 아니라, Engine, Wheel, Handle 과 같은 객체자료형이 올 수 있는 것은 당연한 이치다.  (3) 이 클래스는 멤버 변수의 개수가 4개이다.  (4) Car클래스가 Engine, Wheel, Handle 을 멤버변수로 보유하고 있으므로, Car has a Engine, Car has a  Wheel, Car has a Handle 로 표현할 수 있을 것이다.  (5) price와 마찬가지로 ,e ,w ,h 변수 모두 개발자가 초기화 하지 않았을 경우 컴파일러에 의해 0으로  초기화 된다.  (6) Car 클래스를 메모리에 올리는 방법은 Car c= new Car() 이다. – 인수가 있는 생성자가 지정되어 있어 인수가 없는 default 생성자는 없으므로 인수를 넣어줘야 한다. |
| 7 | 사람을 표현하는 클래스를 디자인 하세요. |
| 8 | 백인을 표현하는 클래스를 디자인 하세요. |
| 9 | 흑인을 표현하는 클래스를 디자인 |
| 10 | 황인종을 표현하는 클래스를 디자인 하세요. |
| 11 | 빈 칸을 채우세요.  Ex) 오리 -------- 새  (1) 한국인 --------- ( )  (2) 승용차 -------- ( )  (3) 얼룩말 --------- ( )  (4) 손목시계 -------- ( )  (5) 종이컵 -------- ( )  (6) 운동화 --------- ( )  (7) 외과,내과 의사 ---- ( )  (8) 민법 -------------( )  (9) 피아노 -------------( )  (10) 동화책 -------------( ) |
| 12 | 현실에 존재하는 객체들 중 상위 객체로부터 파생된 경우는 흔히 볼 수 있다. 자바언어는 이러한 현실상의 객체간 관계를 상속이라는 개념으로 표현하고 있으며, 상속관계에 있는 객체는 ( is a )관계로 볼 수 있다. |
| 13 | 상속에서 상위 객체를 ( super )클래스라 하며, 자식객체는( sub )클래스라 부른다. |
| 14 |  |
| 15 | 다음 설명 중 올바른 것을 모두 고르면?  class Duck extends Bird{  }  (1) 새를 부모로 둔 오리 클래스를 정의한 것이다.  (2) extends를 is a 로 해석해도 문장에 무리가 없다.  (3) 클래스 선언을 Bird extends Duck 으로 순서를 바꾸어도 무리가 없다.-안된다  (4) 이 클래스가 올바로 컴파일 되려면 먼저 Bird 클래스가 컴파일 되어 있어야 할 것이다.  (5) Bird 를 부모클래스라 하며 자바에서는 parent 클래스 라고 표현한다.-super  (6) Duck 을 자식클래스라 하며 자바에서는 child 클래스 라고 표현한다.-sub |
| 14 | 다음 설명 중 맞는 것은?  class Bird{  String name=”그냥 새”;  boolean fly=true;  int legs=2;  }  class Duck extends Bird{  String name=”오리”;  public Duck(String n){  name=n;  }  public static void main(String[] args){  name=”도날드”; (가)에러  Duck d=new Duck(); (나) 에러  d.name=”도날드”; (다) Duck 의 name  d.fly=false; (라)  }  }  (1) (가)에서 Duck 클래스의 name 멤버변수의 값이 도날드로 바뀔 것이다.-에러  (2) (나) 에서 오리의 인스턴스가 1개 생성되어 heap 영역에 올라갈 것이다.-에러. 인수 없음.  (3) (나)에서 에러가 나지 않는다고 가정했을 때 (다)에서 Duck 의 name 변수값은 도날드로 변경될 것이다.-d의 name 이 변경.  (4) (라)의 문장은 제대로 수행될 것이다.-d가 에러 없이 생성되었을 경우 가능. |
| 15 | 다음 설명 중 올바르지 않은 것은?  (1) 상속이란 현실의 상,하위객체간의 관계를 OOP언어가 is a 관계로 표현한 기법이다.  (2) 상속은 extends 키워드로 구현하며, extends 키워드를 is a 로 바꾸어 이해해도 된다.  (3) 상속을 이용하면, 코드를 중복정의 하지 않아도 되며 개발시 재사용이 높아진다.  (4) 부모클래스의 모든 것이 자식클래스에게 물려지지만, 생성자만은 자식 클래스에게 물려지지 않는다.  (5) extends 키워드는 한 클래스내에서 여러 번 사용이 가능하며, 특별히 제한은 없다.-자바나 현실이나 다중 상속은 혀용하지 않는다. |
| 16 | 다음 설명 중 올바르지 않은 것은?  (1) 자식(sub)클래스는 상속을 통하여 부모(super)클래스의 멤버 변수 및 메서드를 물려받을 수 있다.-물려받지만 덮어 쓴다는의미는 아니다.  (2) 상속을 이용하면, 객체가 일반적으로 갖는 공통성 있는 특징,기능들을 부모클래스에 정의하고,  자식 클래스에서는 동일한 코드를 중복정의 하지 않아도 되므로, 개발시 소스의 재사용성이 높아진다.  (3) 현실에서도 다중 상속이 존재하지 않듯, OOP언어에서도 다중 상속은 허용되지 않는다.  (4) 부모클래스에서 멤버변수에 private 으로 접근 제한을 둘 경우, 이 멤버변수 조차도 자식이 접근할 수 있다.- 접근할 수 없다.  (5) 부모의 생성자가 상속되지 않는 이유는 생성자의 목적이 해당 객체가 태어날 때 그 객체만이 갖는 개성과  스타일 등 해당 객체만이 가질 수 있는 유일한 초기화 작업을 지원하기 위함이므로, 생성자를 자식객체가  물려받는다 하더라도, 사용할 의미가 없다고 보아야한다. |
| 17 | 아래 클래스에 대한 설명 중 맞는 것을 모두 고르면?  class Bird{  String name=”새”;  boolean fly;  public Bird(Boolean f){  fly=f;  }  }  class Duck extends Bird{  String sound=”quack”;  }  class UseTest{  public static void main(String[] args){  Bird b=new Bird(); (가)  b.sound=”JJack JJack”; (나)  Duck d=new Duck(); (다) : Duck 의 생성자 메서드를 만들고, super(true) 라고 Bird 의 생성자 지정해야 한다.  }  }  (1) (가)는 Bird 클래스를 메모리에 올리는 과정이므로 아무런 문제가 없다.-인수없음 에러  (2) (나)의 코드는 문제가 없이 실행될 것이다-에러. Bird 에는 sound 가 없다.  (3) (다)의 코드는 Duck 클래스를 메모리에 올리는 과정이므로 아무런 문제가 없다.-Bird 생성자에 인수  (4) 자식클래스는 부모클래스의 name,fly 변수를 상속받는다.  (5) 자식클래스는 부모클래스의 생성자를 상속받는다.-X  (6) 현실과 마찬가지로, 자식보다 부모의 존재가 시점상 우선해야 하므로, 상속 관계에 있는 자식 클래스를  메모리에 올리는 순간, 부모클래스의 인스턴스화 작업이 먼저 선행되어져야 한다. |
| 18 | 다음 설명 중 올바르지 않은 것은?  (1) 자바의 기본 자료형에는 문자,숫자,논리값이 있으며 특히 문자와 숫자 자료형은 서로 형변환이 가능  하다.  (2) 객체형도 자료형이다  (3) 위 (2)번이 맞는 주장이라면, 객체형에 대해서도 형변환의 원칙이 적용되어야 함이 맞다.  (4) 문자와 숫자형은 서로 숫자형이라는 공통성이 있기 때문에 같은 자료형이라고 말할 수 있지만, 객체  자료형은 같은 자료형이라는 개념이 존재할 수 없다.-있다 |
| 19 | 아래 클래스에 대한 설명 중 맞는 것을 모두 고르면?  class Bird{  String name=”난 새”;    public void fly(){  System.out.println(”새가 날아요!!”);  }  }  class Duck extends Bird{  String name=”난 오리”;  public void fly(){  System.out.println(”오리가 날아요!!”);  }  }  class UseTest{  public static void main(String[] args){  Bird b=new Bird();(가)  b.fly(); (나)  Duck d=new Duck(); (다)  d.fly(); (라)  Bird bb = new Duck(); (마)  System.out.println(bb.name);(바)  // 상속관계에서 부모와 자식의 인스턴스가 모두 올라갔을 때,  // 자식이 부모의 메서드를 오버라이드 한 경우, 업그레이드로 간주하여,  // 자식의 메서드를 우선하여 호출해 버림. 변수는 본인 것이 우선순위.  bb.fly(); (사) //메서드는 자식이 override 했을 경우 자식의 메서드를 적용한다.  }  }  (1) (가) 에서 Bird의 인스턴스 뿐만 아니라 Duck의 인스턴스로 함께 생성된다.  (2) (나) 에서의 출력 결과는 “오리가 날아요” 이다  (3) (다) 에 의해 생성되는 인스턴스의 갯수는 1개이다.  (4) (라) 에서의 출력 결과는 “오리가 날아요” 이다.  (5) (마) 에서 에러가 발생할 것이다.  (6) (바)에서의 출력 결과는 “난 오리” 이다.- 변수는 본인 것이 우선순위이므로 “난 새” 가 된다.  (7) (사)에서의 출력 결과는 “오리가 날아요” 이다. |
| 20 | 현실에 존재하는 여러 사물들은 단독적이기보다 서로 관계를 맺고 존재하는 경우가 대부분이다.  특히 OOP언어가 주목하고 있는 객체간의 관계는 2가지가 있는데, 하나는 객체간의 상,하위 관계를  ( is a )관계로 보고 ( extends )키워드로 구현한다.  또 다른 하나는 객체가 특정 객체들을 부품으로 보유하는 관계를 ( has a )보고, 객체자료형을  ( 멤버변수 )로 보유하여 구현한다. |