|  |  |
| --- | --- |
| 1 | 현실에 존재하는 모든 사물은 단독으로 존재할 수 없으며 여러 객체들로 구성되어 있다.  이와 마찬가지로 제대로 된 기능을 갖춘 프로그램의 경우엔 단순히 하나의 클래스로 구성되어 있지 아니하며, 여러 개의 객체들 간 상호 의존성을 갖으며 하나의 프로그램으로 구성된다.  객체 지향 언어인 자바는 이러한 객체들 간의 상호 의존성 관계를 크게 2가지로 보고 있는데, 각각  ( )관계와 ( )관계이다 |
| 2 | 아래의 경우에 해당되는 관계를 채워 넣으세요.   |  |  | | --- | --- | | 현실의 상황 | OOP의 관계 | | 자전거는 바퀴, 핸들, 체인 등 각종 부품으로 구성되어 있다 |  | | 오리는 새이다 |  | | 새끼 고양이는 어미 고양이를 닮았다 |  | | 볼펜은 볼펜심, 몸체, 스프링 등으로 이루어져 있다. |  | |
| 3 | 다음의 괄호에 알맞는 말을 채워 넣으세요.  현실에 존재하는 객체들 중 상위 객체로부터 파생된 경우는 흔히 볼 수 있다. 즉 현실에서의 부모, 자식 관계  를 OOP언어에서도 표현할 수 있는데, 이러한 관계를 ( ) 관계라 하며, extends 키워드로 표현한다  이때 부모 클래스를 ( ) 클래스라 하며, 자식 클래스를 ( ) 클래스라 한다. |
| 4 | 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.  (4) 이 클래스가 올바로 컴파일 되려면 먼저 Bird 클래스가 컴파일 되어 있어야 할 것이다.  (5) Bird 를 부모클래스라 하며 자바에서는 parent 클래스 라고 표현한다.  (6) Duck 을 자식클래스라 하며 자바에서는 child 클래스 라고 표현한다. |
| 5 | 다음의 괄호에 알맞는 말을 채워 넣으세요.  자바에서는 객체가 특정 객체들을 부품으로 보유하는 관계를 ( ) 라 하며, 부품으로 보유할 객체 자료형  을 클래스 내의 ( )로 선언하여 구현한다. |