

c++ 과제 3번

1. 클래스 상속

- 상속은 객체 지향 프로그래밍의 핵심 개념 중 하나로, 클래스 간에 코드를 재사용할 수 있도록 한다. 쉽게 말하자면, 한 클래스가 다른 클래스의 속성과 메서드를 물려받는 것을 말한다.

- 장점

상속의 주요한 장점 중 하나는 코드 재사용성이다.

ex. 'Vehicle'라는 클래스가 있다고 가정한다. 그리고 'Car'라는 새로운 클래스를 만들려고 하는데 'Vehicle' 클래스의 모든 속성이 필요하다면, 상속받을 수 있다. 이때 추가적으로 'Car' 클래스만의 특징을 추가하거나 변경(오버라이딩)할 수 있다.

2. 부모 클래스(기반 클래스)와 자식 클래스(파생 클래스)

- 부모 클래스

- 부모 클래스는 다른 클래스들에서 공통적으로 사용할 수 있는 속성과 기능을 제공하는 클래스이다.

- 부모 클래스는 여러 자식 클래스들에게 이 속성과 기능을 물려줄 수 있다.

- 부모 클래스는 모든 자식 클래스가 가져야 할 기본적인 부분을 정의하는 역할을 한다.

- 자식 클래스

- 자식 클래스는 부모 클래스의 속성과 기능을 상속받은 클래스이다.

- 자식 클래스는 부모 클래스에서 제공하는 기능을 그대로 사용할 수 있고, 필요에 따라 추가적인 기능을 더하거나, 부모 클래스의 기능을 변형할 수 있다.

- 부모와 자식 클래스의 관계

- 부모 클래스의 기능을 상속받은 자식 클래스는 부모 클래스의 모든 공용 멤버에 접근할 수 있다.

- 상속을 통해 코드를 재사용할 수 있고, 자식 클래스는 추가적인 기능을 확장해 사용할 수 있다.

3. Friend, static 그리고 const

- Friend

- private에 접근하게 해주는 키워드이다.

- 클래스의 friend 함수나 클래스를 선언할 때 사용한다.

- 해당 클래스의 private 및 protected 멤버에 접근 가능하다.

- 클래스의 friend 함수는 해당 클래스의 모든 멤버에 접근 가능하다.

- friend 선언은 클래스의 캡슐화를 깨뜨리므로 사용에 주의해야한다. (캡슐화 : 클래스의 멤버를 외부에서 쉽게 접근하지 못하게 정보를 은닉)

- static

- 객체가 아닌 클래스에 속하는 변수/함수를 위한 키워드이며 정적 멤버 변수와 정적 멤버 함수를 정의한다.
- 정적 멤버 변수는 모든 객체가 공유하는 변수이며, 클래스의 인스턴스와 관계없이 클래스 이름으로 직접 접근 가능하다.
- 정적 멤버 함수는 클래스의 인스턴스와 무관하게 호출될 수 있으며, 객체의 상태를 변경하지 않는 곳에 사용 가능하다.

- const

- 변수, 함수, 매개변수, 함수 반환 유형 및 함수 멤버를 상수로 선언할 수 있다.
- 변수가 const로 선언되면 변수의 값을 변경할 수 없다.
- 멤버함수가 const로 선언되면 해당 함수가 객체의 상태를 변경하지 않음을 나타낸다.

4. 상속(단일상속, 다중상속, 가상상속)

- 단일 상속

- 한 클래스가 오직 하나의 부모 클래스만 상속받는 형태이다.
- 자식 클래스는 부모 클래스의 모든 멤버 변수와 멤버 함수를 상속받아 사용할 수 있다.

- 다중 상속

- 하나의 클래스가 둘 이상의 부모 클래스로부터 상속받는 형태이다.
- 여러 클래스의 기능을 하나의 자식 클래스가 모두 사용할 수 있게 하지만, 설계가 복잡해질 수 있다.

- 가상 상속

- 다중 상속 구조에서 발생할 수 있는 "다이아몬드 문제"를 해결하기 위한 기능이다. 다이아몬드 문제를 여러 경로를 통해 부모 클래스를 상속받을 때, 자식 클래스에서 그 부모 클래스의 멤버가 중복되어 모호성이 발생하는 상황이다.

5. 가상 함수(Virtual Functions)

- 다형성을 구현하기 위해 사용되는 함수이다.
- 기본 클래스에서 선언되고 파생 클래스에서 재정의될 수 있는 함수이다.
- 가상 함수를 사용하면 런타임 시점에 객체의 실제 타입에 따라 호출할 함수가 결정된다. 이를 통해 기본 클래스 포인터 또는 참조를 통해 파생 클래스의 함수를 호출할 수 있게 된다.

6. 순수 가상 함수와 추상 클래스

- 순수 가상 함수

- 순수 가상 함수는 구현이 없는 가상 함수로 파생 클래스에서 반드시 재정의해야 하는 함수이다.

- 순수 가상 함수는 기본 클래스에서 구현되지 않고, 선언만 되어있으며 함수 선언 끝네 "=0"을 붙여 선언한다.

- 추상 클래스

- 추상 클래스는 하나 이상의 순수 가상 함수를 포함한 클래스이다.
- 추상 클래스는 인스턴스를 생성할 수 없으며 주로 다른 클래스들이 상속받아 구체적인 기능을 구현할 수 있도록 설계한다. (인스턴스: 클래스에서 생성된 객체를 의미)
- 추상 클래스에서는 *가상 소멸자*를 제공하는 것이 매우 중요하다. 이는 파생 클래스의 객체가 소멸될 때 파생 클래스의 소멸자가 올바르게 호출되도록 보장한다.

if. 가상 소멸자가 없다면?

- 기본 클래스의 소멸자만 호출되고, 파생 클래스의 소멸자가 호출되지 않아 메모리가 제대로 해제되지 않을 수 있다.