

11주차 cpp 9장 가상함수와 추상클래스

가상 함수(virtual function) 429p

- virtual 키워드로 선언된 멤버 함수
- 컴파일러에게 함수에 대한 호출 바인딩을 실행 시간까지 미루도록한다.

함수 오버라이딩(Overriding)

- 기본 클래스에서 가상함수의 존재감을 상실시킴
- 파생 클래스에서 오버라이딩한 함수가 호출되도록 동적으로 바인딩한다.
- 함수재정의는 곧 오버라이딩을 의미하며, 무조건 **동적으로 바인딩** 된다.
- 함수 재정의라고도 함
- 다형성의 한 종류

함수 오버라이딩 조건

- 가상 함수 이름, 매개변수 타입과 개수, 리턴 타입이 모두 일치해야함.

오버라이딩으로 인해 무시된 기본 클래스들의 가상함수를 호출하고싶다면 **범위지정연산자 (::)** 를 사용하여 호출 할 수 있다.

가상 소멸자

- 소멸자를 virtual 키워드로 선언하여 정의
- 소멸자 호출 시 동적 바인딩 발생.

동적바인딩

- 기본클래스의 객체 내에서 가상함수가 호출되어도 동적바인딩은 일어나지않음
- 객체 내에 오버라이딩 된 가상 함수가 있어야 **동적바인딩** 이 일어난다.
- 동적바인딩은 **파생클래스의 객체**에 대해, **기본클래스의 포인터로 가상함수가 호출**될때.

함수 오버로딩 조건

- 중복된 함수들의 이름이 동일
- 중복된 함수들은 **매개변수 타입이나 갯수**가 달라야 한다.
- 함수 중복에 **리턴 타입은 고려되지 않는다.**

토론한점 : 함수 오버로딩을 해야하는 이유는?

-> **하나의 이름**을 가지고 입력되는 인자들의 타입, 갯수의 차이로 같은 동작을 나타낼 수 있다.

예를 들어 굳이 정수형 덧셈, 실수형 덧셈 함수를 만들지 않고, add라는 동일한 이름의 함수를 선언하고, **함수를 호출하는 쪽에서는 인자의 종류에 따라 함수를 구분하여 호출하는 것 없이 하나의 함수만을 호출할 수 있는 편리함을 준다.**

순수 가상 함수 (pure virtual function)

- 함수의 코드가 없고 선언만 있는 가상함수

// 헤더파일로 클래스를 나누고 메인cpp파일에서 코드를 작성할것. 449p

추상클래스

- 최소 하나의 순수 가상 함수를 가진 클래스를 추상 클래스 라고 정의
- 실행코드가 없는 순수 가상함수를 가지고있는 추상 클래스는 불완전한 클래스이다.
- 추상클래스를 단순히 상속하게되는 다른 클래스도 추상클래스가 된다.
- 추상 클래스의 포인터는 선언 가능

추상클래스의 목적

- 인스턴스를 생성할 목적이 아닌, 상속에서의 기본 클래스 역할을 하기위함
-> 추상 클래스의 모든 멤버 함수를 순수 가상 함수로 선언할 필요 없음

추상클래스의 구현방법

- 상속받은 추상 클래스의 모든 순수 가상함수들을 오버라이딩하여 구현하여야 함

논의한점 : 가상 함수와 함수 오버로딩의 용도와 차이

-> 가상 함수와 함수 오버로딩은 둘 다 다형성을 구현하는 데에 사용됨.가상 함수는 주로 상속과 연관되어 있어서, 기본 클래스와 파생 클래스 사이에서 동적 바인딩을 통해 오버라이딩이 가능한걸로 이해함.

-> 반면에 함수 오버로딩은 같은 이름의 함수를 여러 버전으로 정의함, 이때 각 함수 버전은 서로 다른 매개 변수를 가지고 있어야 하며 . 함수 오버로딩은 주로 같은 기능을 수행하지만 다양한 입력을 처리하고자 할 때 사용됨. 호출 시에 인자의 타입이나 개수에 따라 적절한 버전이 선택될 것임.