

•가상함수

Virtual키워드로 선언된 함수

Virtual키워드

-동적 바인딩 지시어

-컴파일러에세 함수에 대한 호출 바인딩을 실행시간
까지 미루도록 지시.

•오버라이딩

파생클래스에서 기본클래스의 가상함수와 동일한
이름의 함수 선언

-기본 클래스 가상함수 존재감 상실 시킴

-파생클래스에서 오버라이딩한 함수가 함수가 호출
되도록 동적바인딩

-함수재정의라고도 부르고, 다형성의 한 종류

•오버라이딩 특징

-가상함수이름, 매개변수 타입과 개수 리턴 타입 모
두 일치해야함

-오버라이딩시 virtual 키워드 생략 가능

-접근지정자 private, public, protected 자유롭게
사용 가능

•오버라이딩목적

-파생클래스에서 재정의할 함수에 대한 인터페이스

재공

•함수재정의, 오버라이딩 비교

함수재정의: 기본 클래스의 객체, 파생클래스의 객체에 있는 이름이 같은 멤버함수가 동등한 호출기회 가짐.

오버라이딩: 기본클래스객체 멤버 함수는 존재감 상실하고, 파생클래스 객체 멤버함수 호출.

•동적바인딩

-기본클래스에 대한 포인터로 가상함수를 호출할 때 객체내에 오버라이딩한 파생클래스 함수를 찾아 실행

•기본클래스에 가상 함수를 호출하고 싶을때

- :: 범위 지정연산자 사용(기본클래스이름::가상함수())

•가상 소멸자

-virtual키워드로 선언

-소멸자 호출시 동적바인딩 발생

•순수 가상함수

-함수에 코드가 없고 선언만 있는 가상함수

-선언방법: virtual 멤버함수=0;

•추상클래스

-최소 한개 이상의 순수 가상함수가 있는 클래스

-온전한 클래스x, 객체 생성불가

-포인터 선언은 가능

•추상클래스 목적

-파생클래스에서 구현할 가상함수의 원형을 보여주
큰 인터페이스 역할

•추상클래스 상속

-추상클래스를 상속한 파생클래스는 자동 추상클래스

-추상클래스를 상속받아 순수 가상 함수를 오버라이딩해 객체 생성 가능.

☆☆추상클래스 연습하기☆☆

추상클래스 Calculator에 선언된 순수가상함수
calc를 파생클래스를 아래 실행결과가 나오게 만들기

```
#include <iostream>
using namespace std;

class Calculator {
    void input() {
        cout << "정수 2 개를 입력하세요>> ";
        cin >> a >> b;
    }
protected:
    int a, b;
    virtual int calc(int a, int b) = 0; // 두 정수의 합 리턴
public:
    void run() {
        input();
        cout << "계산된 값은 " << calc(a, b) << endl;
    }
};

int main() {
    Adder adder;
    Subtractor subtractor;
    adder.run();
    subtractor.run();
}
```

정수 2 개를 입력하세요>> 5 3

-계산된 값은 8

정수 2 개를 입력하세요>> 5 3

-계산된 값은 2

```
class Adder:public Calculator {
protected:
    int calc(int a, int b){
        return a+b;
    }
}
```

```
class Subtract:public Calculator {
protected:
    int calc(int a, int b){
        return a-b;
    }
}
```

