정리노트 #5

13조 (소재현, 전대원)

상속:

C++에서의 상속은 클래스 안에서 상속관계를 정의하는 것.

ㅇ 기본 클래스(base class) - 상속해주는 클래스. 부모 클래스

ㅇ 파생 클래스(derived class) – 상속받는 클래스. 자식 클래스

Ex (class Student : public Person) 이 코드의 경우

Student 가 파생 클래스가 되어서 물려받고,

Person 이 기본 클래스로써 부모 클래스가 됨.

예제 8-1 정리

```
class Point {
int x, y; //한 점 (x,y) 좌표값
public:
void set(int x, int y)
```

- Point 클래스는 부모 클래스가 되고 여기에서 입력해두었던 코드를 이 class ColorPoint : public Point 자식 클래스 코드에서 cp.set(3,4); 의 형태로 그대로 상속받아 사용 할 수 있다.

ㅇ 파생 클래스가 기본 클래스를 호출 하는 경우도 가능하다

Ex) void set(int x, int y) {

this->x= x; this->y=y;

}

void showPoint() {

cout << x << y;

}

color

void showColorPoint() {

cout << color << ":";

showPoint();

}

업 캐스팅(up-casting)

ㅇ 파생 클래스 포인터가 기본 클래스 포인터에 치환되는 것

다운 캐스팅(down-casting)

ㅇ 기본 클래스의 포인터가 파생 클래스의 포인터에 치환되는 것

접근 지정자 (복습)

private 멤버

ㅇ 선언된 클래스 내에서만 접근 가능

ㅇ 파생 클래스에서도 기본 클래스의 private 멤버 직접 접근 불가

public 멤버

ㅇ 선언된 클래스나 외부 어떤 클래스, 모든 외부 함수에 접근 허용

ㅇ 파생 클래스에서 기본 클래스의 public 멤버 접근 가능

protected 멤버

ㅇ 선언된 클래스에서 접근 가능

ㅇ 파생 클래스에서만 접근 허용

예제 8-4. private 상속 사례 PairPrograming

<전대원> : 컴파일 오류 1번부터 6번까지 모두라고 생각

Class Base (기본클래스)에서 private Base로 class Derived (파생 클래스) 상속 시 Class Base 를 모두 **Private** 로 private Base에 가져오게 된다

void showB() { cout << b; } 이 부분은 상속 받은 것이 아닌

class Derived 것이므로 컴파일 오류에 해당하지 않음.

단순 착각

<소재현> : 기본클래스에서 Public 으로 설정된 부분은

Private 로 처리하지 않아서 x.showA(); , x.showB(); 는 컴파일 오류 아님으로 풀이. >>> Private로 상속된 것은 모든 상속을 다 private 처리 하므로 **<전대원>** 과 다시 상속 유형을 정리하였음.

예제 8-5 PairPrograming

<전대원>, <소재현> : 예제 8-4 를 통해 public , protected, private의 상속 유형 정리후 Protected Base로 상속하였으나, int maint()에서 Protected 코드를 사용하지 않은점을 통해 컴파일 오류 다 잘 찾아냄.

다중 상속 : 두 가지 이상의 클래스를 상속 받을 수 있다

하지만 여러 컴파일 오류들을 초래할 수 있어 유의하여 사용해야하며 가상상속으로 기본 클래스 앞에 virtual로 선언하여 컴파일 오류를 막는다.