

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 48강 - C++의 클래스 상속

학습 목표

C++의 클래스 상속

- 1) C++의 클래스 상속을 이용해 객체 지향 프로그래밍의 장점을 활용할 수 있는 방법에 대해 이해할 수 있습니다.

C++의 클래스 상속

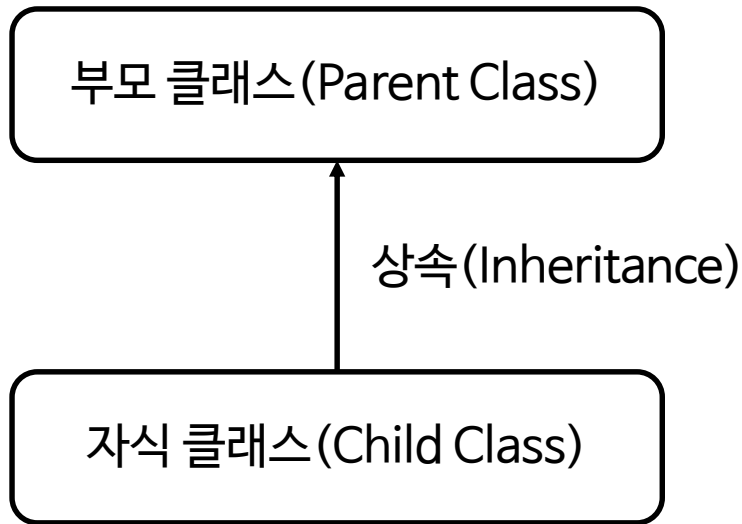
상속

상속(Inheritance)은 객체 지향 프로그래밍의 주요한 특성 중 하나입니다. 현실 세계에서의 상속의 개념을 프로그래밍으로 그대로 가져와 사용할 수 있습니다. 이를 통해 프로그램의 논리적 구조를 계층적으로 구성할 수 있습니다.

흔히 자식이 부모의 속성을 물려 받듯이, 자식 클래스(Child Class)가 부모 클래스(Parent Class)의 속성을 그대로 물려 받아 사용할 수 있습니다. 그러므로 상속을 활용하여 소스코드의 재사용성을 늘일 수 있습니다.

C++의 클래스 상속

상속



C++의 클래스 상속

상속

자식 클래스는 파생 클래스(Derived Class)라고도 불리우며, 부모 클래스(Parent Class)의 모든 속성을 물려 받습니다. 자식 클래스는 콜론(:)을 활용하여 부모 클래스와 연결될 수 있습니다.

C++의 클래스 상속

상속: 부모 클래스 정의하기

```
#include <iostream>
#include <string>

using namespace std;

class Person {
private:
    string name;
public:
    Person(string name): name(name) { }
    string getName() {
        return name;
    }
    void showName() {
        cout << "이름: " << getName() << '\n' ;
    }
};
```

C++의 클래스 상속

상속: 자식 클래스 정의 및 사용하기

```
class Student: Person {  
private:  
    int studentID;  
public:  
    Student(int studentID, string name) : Person(name) {  
        this->studentID = studentID;  
    }  
    void show() {  
        cout << "학생 번호: " << studentID << '\n';  
        cout << "학생 이름: " << getName() << '\n';  
    }  
};  
  
int main(void) {  
    Student student(1, "나동빈");  
    student.show();  
    system("pause");  
}
```

C++의 클래스 상속

상속

자식 클래스의 인스턴스를 만들 때 가장 먼저 부모 클래스의 생성자가 호출됩니다. 이후에 자식 클래스의 생성자가 호출됩니다. 또한 자식 클래스의 수명이 다했을 때는 자식 클래스의 소멸자가 먼저 호출된 이후에 부모 클래스의 소멸자가 호출됩니다.

C++의 클래스 상속

오버라이딩

오버라이딩(Overriding)은 부모 클래스에서 정의된 함수를 무시하고, 자식 클래스에서 동일한 이름의 함수를 재정의하는 문법입니다. 오버라이딩을 적용한 함수의 원형은 기존의 함수와 동일한 매개변수를 전달 받습니다.

C++의 클래스 상속

오버라이딩: 자식 클래스 정의 및 사용하기

```
class Student: Person {
private:
    int studentID;
public:
    Student(int studentID, string name) : Person(name) {
        this->studentID = studentID;
    }
    void show() {
        cout << "학생 번호: " << studentID << '\n' ;
        cout << "학생 이름: " << getName() << '\n' ;
    }
    void showName() {
        cout << "학생 이름: " << getName() << '\n' ;
    }
};

int main(void) {
    Student student(1, "나동빈");
    student.showName();
    system("pause");
}
```

C++의 클래스 상속

다중 상속

다중 상속(Multipple Inheritance)는 여러 개의 클래스로부터 멤버를 상속 받는 것을 말합니다.

C++의 클래스 상속

다중 상속: 부모 클래스 하나 더 정의하기

```
class Temp {  
public:  
    void showTemp() {  
        cout << "임시 부모 클래스입니다.\n";  
    }  
};
```

C++의 클래스 상속

다중 상속: 자식 클래스 사용해보기

```
class Student: Person, public Temp {
private:
    int studentID;
public:
    Student(int studentID, string name) : Person(name) {
        this->studentID = studentID;
    }
    void show() {
        cout << "학생 번호: " << studentID << '\n';
        cout << "학생 이름: " << getName() << '\n';
    }
    void showName() {
        cout << "학생 이름: " << getName() << '\n';
    }
};

int main(void) {
    Student student(1, "나동빈");
    student.showName();
    student.showTemp();
    system("pause");
}
```

C++의 클래스 상속

다중 상속의 한계

- 1) 여러 개의 부모 클래스에 동일한 멤버가 존재할 수 있습니다.
- 2) 하나의 클래스를 의도치 않게 여러 번 상속받을 가능성이 있습니다.

배운 내용 정리하기

C++의 클래스 상속

- 1) C++의 클래스 상속은 객체 지향 프로그래밍의 중요한 키워드입니다.
- 2) 상속의 원리를 활용하여 소스코드의 재사용성을 증대 시킬 수 있습니다.