

C++ 상속에 대하여

1108 서민준

컨트롤 클래스?

1 // Visual Studio 2019에서 작성된 코드입니다.

해당 프로그램이 어떠한 기능을 제공하는지 구체적으로 알 수 있다.

```
2
3 #include <iostream>
4 class EmployeeHandler
5 {
6     private:
7         PermanentWorker* empList[50];
8         int empNum;
9
10    public:
11        EmployeeHandler() : empNum(0) {}
12
13        void AddEmployee(PermanentWorker* emp) { . . . }
14
15        void ShowAllSalaryInfo() const { . . . }
16
17        void ShowTotalSalary() const { . . . }
18
19        ~EmployeeHandler() { . . . }
20    };
```

상속이란?

```
class Person
{
private:
    int age; // 나이
    char name[50]; // 이름

public:
    Person(int myage, const char* myname)
        : age(myage)
    {
        strcpy(name, myname);
    }

    void WhatYourName() const
    {
        std::cout << "My name is " << name << std::endl;
    }

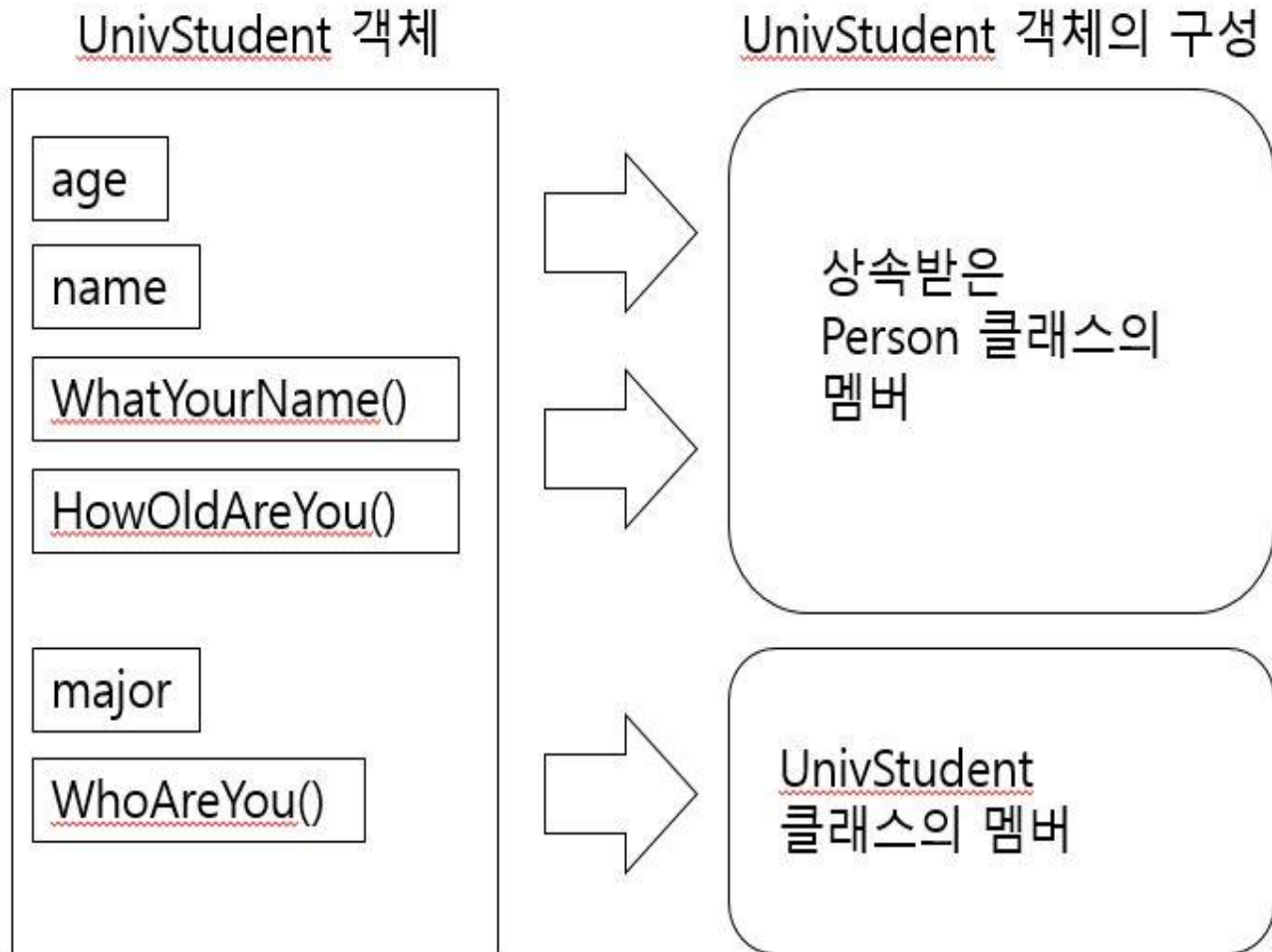
    void HowOldAreYou() const
    {
        std::cout << "I'm " << age << " years old." << std::endl;
    }
};
```

```
class UnivStudent : public Person // Person 클래스의 상속을 의미함
{
private:
    char major[50]; // 전공과목

public:
    UnivStudent(const char* myname, int myage, const char* mymajor)
        : Person(myage, myname)
    {
        strcpy(major, mymajor);
    }

    void WhoAreYou() const
    {
        WhatYourName();
        HowOldAreYou();
        std::cout << "My major is " << major << std::endl << std::endl;
    }
};
```

UnivStudent의 구성



상속의 생성자

```
UnivStudent(const char* myname, int myage, const char* mymajor)
    : Person(myage, myname)
{
    strcpy(major, mymajor);
}
```

Q1

UnivStudent 클래스의 생성자는 Person 클래스의 멤버까지 초기화해야 하는가?

Q2

UnivStudent 클래스의 생성자가 Person 클래스의 멤버를 어떻게 초기화해야 하는가?

private 멤버의 접근

UnivStudent 클래스의 멤버함수(또는 생성자) 내에서
Person의 private 멤버에 접근이 가능한가?

접근제한의 기준은 **클래스**!
따라서 **클래스 외부**에서는 private 멤버에는 접근이 **불가능**하다.

‘정보의 은닉’은 하나의 객체 내에서도 진행이 된다.

상속 용어 정리

Person	↔	UnivStudent
상위 클래스	↔	하위 클래스
기초(base) 클래스	↔	유도(derived) 클래스
슈퍼(super) 클래스	↔	서브(sub) 클래스
부모 클래스	↔	자식 클래스

유도 클래스의 생성과정 (1)

```
class SoBase
{
private:
    int baseNum;

public:
    SoBase() : baseNum(20)
    {
        std::cout << "SoBase()" << std::endl;
    }

    SoBase(int n) : baseNum(n)
    {
        std::cout << "SoBase(int n)" << std::endl;
    }

    void ShowBaseData() const
    {
        std::cout << baseNum << std::endl;
    }
};
```

```
class SoDerived : public SoBase
{
private:
    int derivNum;

public:
    SoDerived() : derivNum(30)
    {
        std::cout << "SoDerived()" << std::endl;
    }

    SoDerived(int n) : derivNum(n)
    {
        std::cout << "SoDerived(int n)" << std::endl;
    }

    SoDerived(int n1, int n2) : SoBase(n1), derivNum(n2)
    {
        std::cout << "SoDerived(int n1, int n2)" << std::endl;
    }

    void ShowDerivData() const
    {
        ShowBaseData();
        std::cout << derivNum << std::endl;
    }
};
```


유도 클래스의 생성과정 (2)

```
int main()
{
    std::cout << "case 1....." << std::endl;
    SoDerived dr1;
    dr1.ShowDer ivData();
    std::cout << "-----" << std::endl;

    std::cout << "case 2....." << std::endl;
    SoDerived dr2(12);
    dr2.ShowDer ivData();
    std::cout << "-----" << std::endl;

    std::cout << "case 3....." << std::endl;
    SoDerived dr3(23, 34);
    dr3.ShowDer ivData();
    std::cout << "-----" << std::endl;

    return 1;
}
```

실행결과

```
Microsoft Visual Studio 디버그 콘솔
case 1.....
SoBase()
SoDerived()
20
30
-----
case 2.....
SoBase()
SoDerived(int n)
20
12
-----
case 3.....
SoBase(int n)
SoDerived(int n1, int n2)
23
34
-----
```