

프로그래밍입문(2) 실습

10주차

상속의 기본

- 예제1)

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 class Car{
6     int getHP(){
7         return 100;
8     }
9 };
10
11 class SportsCar : public Car{
12     public:
13     int getHP(){
14         return 300;
15     }
16 };
17
18 int main(){
19     SportsCar sc;
20     cout << "마력 :" << sc.getHP() << endl;
21     return 0;
22 }
```

상속의 기본

- 예제2)

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  class Human{
6      string name;
7      int age;
8
9      public:
10     Human(string n, int a): name(n), age(a){ }
11     string getName(){
12         return name;
13     }
14     int getAge(){
15         return age;
16     }
17     void setName(string n){
18         name = n;
19     }
20     void setAge(int a){
21         age = a;
22     }
23     void print(){
24         cout << "이름 : " << name << endl;
25         cout << "나이 : " << age << endl;
26     }
27     ~Human(){
28         cout << "Human 생성자 호출" << endl;
29     }
30 };
```

상속의 기본

- 예제2)

```
32 ▾ class Student : public Human{
33     string major;
34
35     public :
36 ▾     Student(string name, int age, string major) : Human(name, age){
37         this->major=major;
38     }
39 ▾     string getMajor(){
40         return major;
41     }
42 ▾     void setMajor(string m){
43         major=m;
44     }
45 ▾     void print(){
46         Human::print();
47         cout<< "전공 :" << major << endl;
48     }
49 ▾     ~Student(){
50         cout << "Student 소멸자 호출"<< endl;
51     }
52 };
53
54 ▾ int main(){
55     Student s1("명진", 21, "컴퓨터"), s2("미현", 22, "경영");
56
57     s1.print();
58     s2.print();
59 }
```

Public 상속

- 예제3)

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 class Person{
6     int age;
7     string name;
8     public:
9         int GetAge() const{
10             return age;
11         }
12         string GetName() const{
13             return name;
14         }
15         Person(int _age=1, string _name="noname"){
16             age=_age;
17             name=_name;
18         }
19 };
```

Public 상속

- 예제3)

```
21 ▾ class Student: public Person{
22     string major;
23     public:
24 ▾     Student(string _major){
25         |     major=_major;
26         |     }
27 ▾     string GetMajor() const{
28         |     return major;
29         |     }
30 ▾     void ShowData() const{
31         |     cout << "이름: " << GetName() << endl;
32         |     cout << "나이: " << GetAge() << endl;
33         |     cout << "전공: " << GetMajor() << endl;
34         |     }
35     };
36
37 ▾ int main(){
38     Student Kim("computer");
39     Kim.ShowData();
40     return 0;
41 }
```

C++ 상속의 이해

- 예제4) 다음 클래스는 은행 계좌 정보를 담을 수 있도록 정의되어 있는 **Account** 클래스이다.

```
class Account{
    private:
        string acc_num; // 계좌번호
        int balance;    // 계좌잔액

    public:
        Account(string num, int bal){
            acc_num = num;
            balance = bal;
        }
};
```

C++ 상속의 이해

- 예제4) 문제 충족 조건

- >> 이전 슬라이드의 클래스를 **public** 상속하는 **KBAccount** 클래스를 정의하라.
- >> **KBAccount** 클래스는 **Account** 클래스가 지니고 있는 멤버 변수 이외에 고객별 이체 한도 정보를 담고 있는 멤버 변수(**trans_limit**)를 가져야 한다.
- >> 다음 슬라이드에 제시하고 있는 **main** 함수의 예와 실행 결과를 참조해서 정의하라.

C++ 상속의 이해

- 예제4)

- main 함수의 예

```
45 int main(){
46     //계좌번호 : 1234-5678
47     //초기 입금액 : 5000원
48     //이체 한도 : 100000원
49     KBAccount acc("1234-5678", 5000, 100000);
50     acc.ShowData();
51     return 0;
52 }
```

C++ 상속의 이해

- 예제4) Solution

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 class Account{
6     private:
7         string acc_num; // 계좌번호
8         int balance;    // 계좌잔액
9
10    public:
11    Account(string num, int bal){
12        acc_num = num;
13        balance = bal;
14    }
15    const string getAcc_num(void) const;
16    int getBalance(void) const;
17 };
18
19 const string Account::getAcc_num() const{
20     return acc_num;
21 }
22
23 int Account::getBalance() const{
24     return balance;
25 }
```

C++ 상속의 이해

- 예제4) Solution

```
27 class KBAccount:public Account{
28     double trans_limit;
29
30     public:
31         KBAccount(string a_num, int a_bal, double limit):Account(a_num, a_bal){
32             trans_limit=limit;
33         }
34         void ShowData();
35 };
36
37 void KBAccount::ShowData(){
38     cout << "계좌 번호 : " << getAcc_num() << endl;
39     cout << "계좌 잔액 : " << getBalance() << "원" << endl;
40     cout << "이체 한도 : " << trans_limit << "원" << endl;
41 }
42
43 int main(){
44     //계좌번호 : 1234-5678
45     //초기 입금액 : 5000원
46     //이체 한도 : 100000원
47     KBAccount acc("1234-5678", 5000, 100000);
48     acc.ShowData();
49     return 0;
50 }
```

Q & A