

프로그래밍입문(2) 실습

12주차

C++ 상속의 이해

- 예제1) 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

- 예제1) 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

- 예제1)

- 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

- 예제1) 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

- 예제1) 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 }
```

상속

- 예제2) **Person** 클래스를 설계하라. **Person** 클래스는 이름, 주소, 전화 번호를 멤버 변수로 가진다. 하나 이상의 생성자를 정의하고 각 멤버 변수에 대하여 접근자와 설정자 함수를 작성하라. 이어서 **Person**을 상속받아서 **Customer**를 작성하여 보자. **Customer**는 고객 번호(**id**)와 마일리지(**mileage**)를 멤버 변수로 가지고 있다. 한 개 이상의 생성자를 작성하고 적절한 접근자 함수와 설정자 함수를 작성하라.

```
4 class Person{
5     string name;
6     string address;
7     string tel;
8 };
```

```
이름을 입력하세요 :김철수
주소를 입력하세요 :서울시
연락처를 입력하세요 :010-1101-2202
ID를 입력하세요 :110
마일리지를 입력하세요 :1500
-----고객 정보-----
이름: 김철수
주소: 서울시
연락처: 010-1101-2202
고객ID: 110
마일리지: 1500
```

상속

```
1 #include <iostream>
2 using namespace std;
3
4 class Person{
5     string name;
6     string address;
7     string tel;
8
9     public:
10    Person(string name1, string address1, string tel1){
11        name=name1;
12        address=address1;
13        tel=tel1;
14    }
15    void setName(string name1){
16        name=name1;
17    }
18    void setAddress(string address1){
19        address=address1;
20    }
21    void setTel(string tel1){
22        tel=tel1;
23    }
24    string getName(){
25        return name;
26    }
27    string getAddress(){
28        return address;
29    }
30    string getTel(){
31        return tel;
32    }
33 };
```


상속

```
35 class Customer : public Person{
36     int ID;
37     int mileage;
38
39     public:
40     Customer(string name1, string address1, string tel1, int ID1, int mileage1):Person(name1, address1, tel1){
41         ID=ID1;
42         mileage=mileage1;
43     }
44     void show_Customer(){
45         cout << "-----고객 정보-----" << endl;
46         cout <<"이름: " << getName() << endl;
47         cout <<"주소: " << getAddress() << endl;
48         cout <<"연락처: " << getTel() << endl;
49         cout <<"고객ID: " << ID << endl;
50         cout <<"마일리지: " << mileage << endl;
51     }
52 };
53
54 int main(){
55     string name1, address1, tel1;
56     int ID1, mileage1;
57
58     cout << "이름을 입력하세요 :"; cin >> name1;
59     cout << "주소를 입력하세요 :"; cin >> address1;
60     cout << "연락처를 입력하세요 :"; cin >> tel1;
61     cout << "ID를 입력하세요 :"; cin >> ID1;
62     cout << "마일리지를 입력하세요 :"; cin >> mileage1;
63
64     Customer c(name1, address1, tel1, ID1, mileage1);
65
66     c.show_Customer();
67
68 }
```

상속과 오버라이딩

예제3) 다음 조건을 만족하는 N1 클래스와 N1 클래스를 상속하는 N2 클래스를 작성하시오.

```
1 #include <iostream>
2 using namespace std;
3
4 class N1{
5     public:
6         void oper(){ cout<< "N1's oper()" << endl; }
7 };
8
9 class N2 : public N1{
10     public:
11         void oper(){ cout<< "N2's oper()" << endl; }
12 };
13
14 int main(){
15     N2 n2;
16     n2.oper();
17 }
```

가상함수

예제4) 예제2에서 나온 소스코드를 참고하여 다음과 같은 결과값이 나올 수 있도록 클래스를 수정하십시오.

```
14 int main(){
15
16     N1 *pn1, n1;
17     N2 n2;
18
19     pn1=&n1;
20     pn1->oper();
21
22     pn1=&n2;
23     pn1->oper();
24
25     return 0;
26 }
```

```
N1's oper()
N2's oper()
```

가상 소멸자

- 예제5)

```
1 #include <iostream>
2 using namespace std;
3
4 class N1{
5     public:
6         N1(){ cout << "N1()" << endl; }
7         ~N1(){
8             cout << " ~N1() " << endl;
9         }
10 };
11
12 class N2 : public N1{
13     public:
14         char *a;
15         N2(){
16             a= new char[300];
17             cout << "N2()" << endl;
18         }
19         ~N2(){
20             delete[] a;
21             cout << " ~N2() " << endl;
22         }
23 };
```

```
25 int main(){
26
27     cout << "1" << endl;
28     N2 *n2= new N2();
29     delete n2;
30
31     cout << "2" << endl;
32     N1* n1= new N2();
33     delete n1;
34
35     return 0;
36 }
```

Q & A