설계 사양서

멀티미디어공학과
Tree조
0791081-장진우
0791057-김현원

해결방법 및 프로그래밍 설명

▷ 입력받은 수입, 지출 값을 통해 계산.

▷ 입력받은 날짜, 내역, 계산된 수입, 지출, 밸런스 값 출력.

```
class Account // 클래스선언
{
        string s_Date; // 멤버변수선언
        string s_Item;
        static int i_Tot; // static 멤버변수

public:
        int i_Inc;
        int i_Exp;
        int i_Bal;

        Account(); // 생성자
        void set_Account(string date, string item, int income, int expense); //
멤버함수원형
        void out_Data();
};

Account::Account() // 생성자정의
{
        i_Bal = 0;
}

void Account::set_Account(string date, string item, int income, int expense) //
멤버함수정의
{
        i_Bal = i_Tot;
        s_Date = date;
        s_Item = item;
        i_Inc = income;
        i_Exp = expense;
        i_Bal += income; // i_Bal = i_Bal + income;
        i_Bal -= expense; // i_Bal = i_Bal - expense;
        i_Tot = i_Bal;
}

void Account::out_Data()
{
        cout <<"|"<< setw(10) << fixed << s_Date;
        cout <<"|"<< setw(16) << fixed << s_Item;
        cout <<"|"<< setw(16) << fixed << i_Inc;
        cout <<"|"<< setw(16) << fixed << i_Exp;
        cout <<"|"<< setw(16) << fixed << i_Bal << "|"
}
```

▷ 전체 입력할 건수 입력받음.(동적메모리 할당)

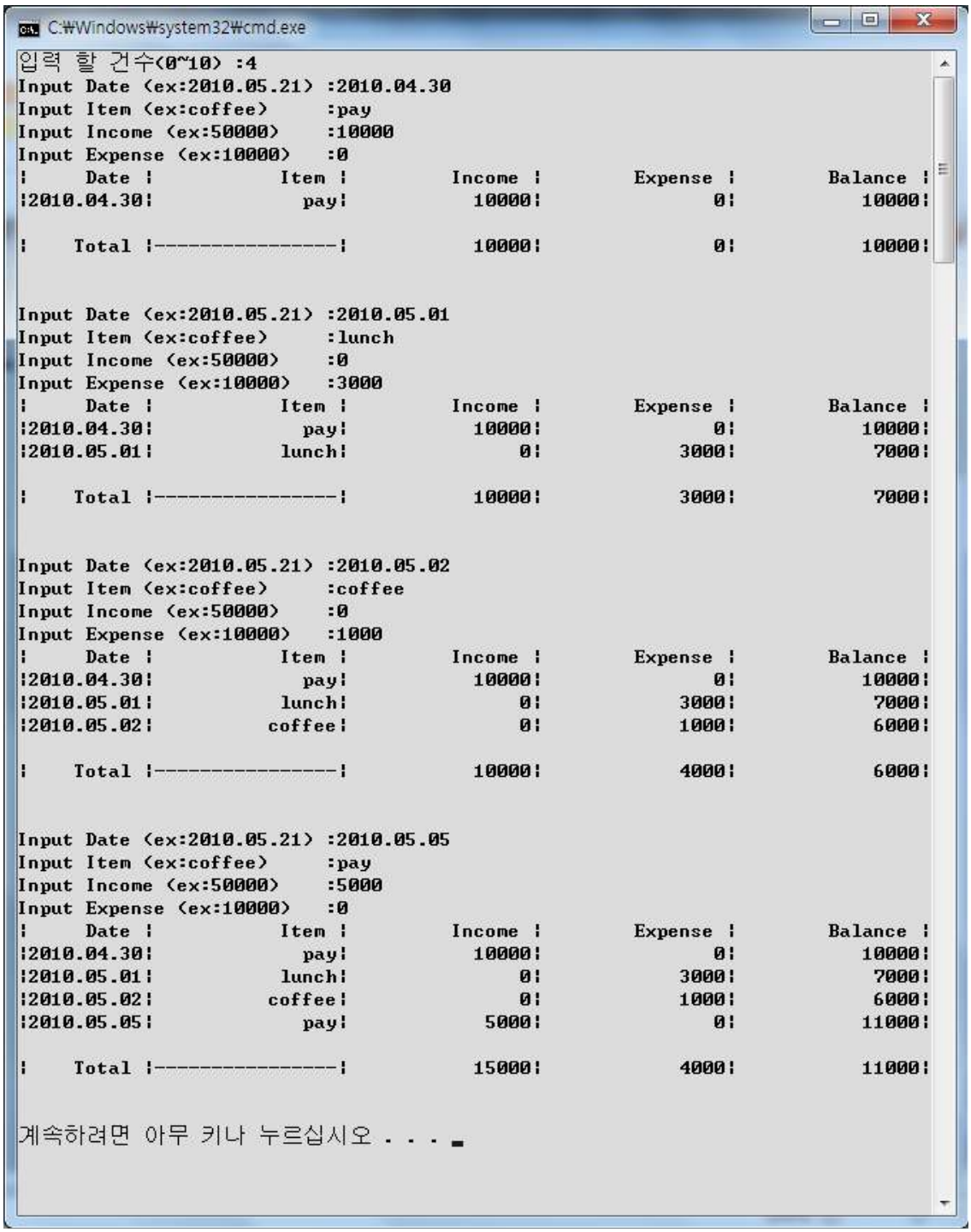
▷ 건수에 따라 차례로 각 데이터 입력받음.(for loop로 반복)

▷ 전체 계산된 값 출력.

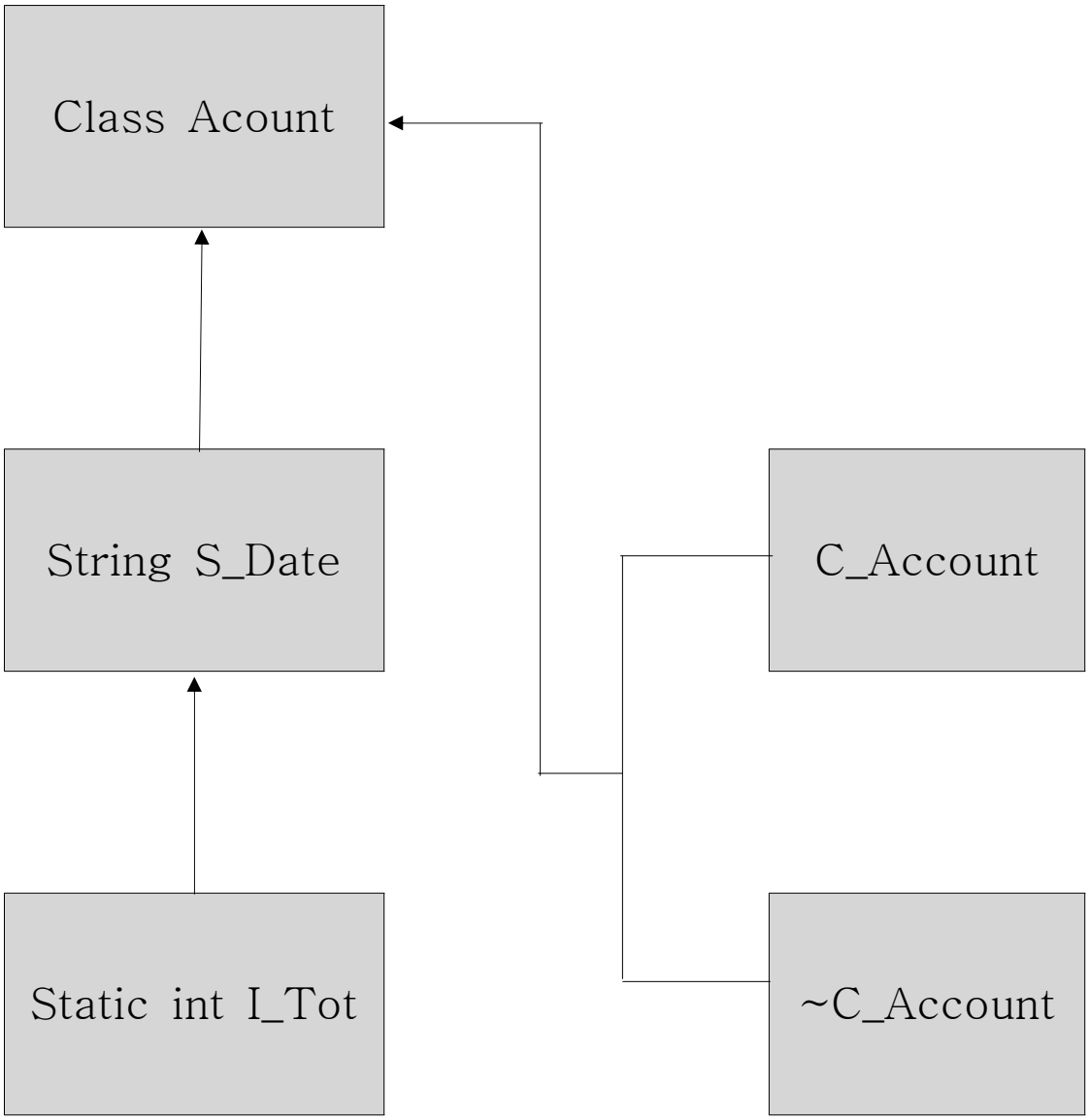
```
int Account::i_Tot = 0; // static 멤버변수초기화

class c_Accounts // 클래스선언
{
        Account *A_Acc; // 포인터 객체
        int i_max;
        int i_count;

public:
        c_Accounts(); // 생성자
        int get_Max();
        void set_Max();
        void set_Data();
        void out_Data();
        ~c_Accounts(); // 소멸자
};

c_Accounts::c_Accounts()
{
        i_count = 0;
}

int c_Accounts::get_Max()
{
        return i_max;
}
void c_Accounts::set_Max()
{
        cout << "입력 할 건수(0~10) :";
        cin >> i_max; // 건수 입력

        A_Acc = new Account[i_max]; // 동적메모리 할당
}

void c_Accounts::set_Data()
{
        string date, item;
        int income, expense;

        if(i_count >= i_max)
                return

        cout << "Input Date (ex:2010.05.21) :"
        cin >> date;
        cout << "Input Item (ex:coffee)     :"
        cin >> item;
        cout << "Input Income (ex:50000)    :"
        cin >> income;
```

```
        cout << "Input Expense (ex:10000)   :"
        cin >> expense;

        A_Acc[i_count++].set_Account(date, item, income, expense);
}

void c_Accounts::out_Data()
{
        int to_Inc = 0, to_Exp = 0, to_Bal = 0;

        cout << "|     Date |           Item |         Income |        Expense |
      Balance |"

        for(int i=0; i<i_count; i++) // for loop: i가 i_count보다 작을 때까지 반
복
        {
                A_Acc[i].out_Data(); // 입력값, Balance 계산값 출력
                to_Inc += A_Acc[i].i_Inc; // 모든 Income 계산
                to_Exp += A_Acc[i].i_Exp; // 모든 Expense 계산
                to_Bal += A_Acc[i].i_Bal; // 모든 Balance 계산
        }
        cout << "₩n|    Total |----------------|" // Total 값 출력
        cout << setw(16) << fixed << to_Inc << "|"
        cout << setw(16) << fixed << to_Exp << "|"
        cout << setw(16) << fixed << to_Inc-to_Exp << "|₩n₩n"
}

c_Accounts::~c_Accounts()
{
        delete[] A_Acc; // 동적메모리 해제
}
```

실험 결과


C:₩Windows₩system32₩cmd.exe
입력 할 건수(0~10) :4
Input Date (ex:2010.05.21) :2010.04.30
Input Item (ex:coffee) :pay
Input Income (ex:50000) :10000
Input Expense (ex:10000) :0
| Date | Item | Income | Expense | Balance |
|2010.04.30| pay| 10000| 0| 10000|
| Total |----------| 10000| 0| 10000|
Input Date (ex:2010.05.21) :2010.05.01
Input Item (ex:coffee) :lunch
Input Income (ex:50000) :0
Input Expense (ex:10000) :3000
| Date | Item | Income | Expense | Balance |
|2010.04.30| pay| 10000| 0| 10000|
|2010.05.01| lunch| 0| 3000| 7000|
| Total |----------| 10000| 3000| 7000|
Input Date (ex:2010.05.21) :2010.05.02
Input Item (ex:coffee) :coffee
Input Income (ex:50000) :0
Input Expense (ex:10000) :1000
| Date | Item | Income | Expense | Balance |
|2010.04.30| pay| 10000| 0| 10000|
|2010.05.01| lunch| 0| 3000| 7000|
|2010.05.02| coffee| 0| 1000| 6000|
| Total |----------| 10000| 4000| 6000|
Input Date (ex:2010.05.21) :2010.05.05
Input Item (ex:coffee) :pay
Input Income (ex:50000) :5000
Input Expense (ex:10000) :0
| Date | Item | Income | Expense | Balance |
|2010.04.30| pay| 10000| 0| 10000|
|2010.05.01| lunch| 0| 3000| 7000|
|2010.05.02| coffee| 0| 1000| 6000|
|2010.05.05| pay| 5000| 0| 11000|
| Total |----------| 15000| 4000| 11000|
계속하려면 아무 키나 누르십시오 . . .

패키지 다이어그램


Class Acount
String S_Date
Static int I_Tot
C_Account
~C_Account