

객체지향프로그래밍

Assignment Report 3-2

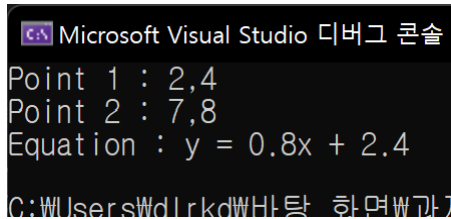
2019202050 이강현

1. [Virtual Function] 아래의 main함수가 결과화면과 같이 동작하게 class Equation과 class TwoPoint를 구현하시오. class Equation은 TwoPoint를 상속받으며 두 class에는 print와 생성자외의 함수는 없다. 과제 제출시 main함수를 포함해서 제출

문제 설명: 가상함수를 이용하여 상속관계에 있는 두 클래스의 함수에 접근을 용이하게 한다.

모두 TwoPoint 클래스 포인터변수로 클래스변수를 가리키므로 가상함수를 사용하여야 원하는 값을 얻을 수 있다.

필요한 개념: 가상함수 사용법, 클래스의 상속, 생성자의 활용



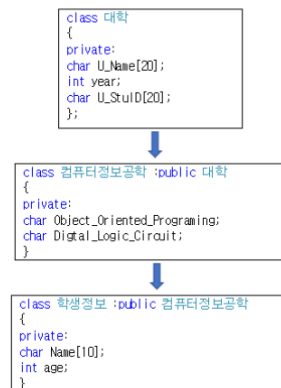
Microsoft Visual Studio 디버그 콘솔

```
Point 1 : 2,4  
Point 2 : 7,8  
Equation :  $y = 0.8x + 2.4$ 
```

C:\Users\Wd\l\kd\W\H\타... 화면...<실행 결과>

고찰: 생성자와 print함수 이외의 멤버함수를 가지면 안되므로 print라는 함수를 상속받은 클래스에서도 사용해야만 했다. 하지만 같은 클래스 포인터를 사용하기 때문에 포인터의 주소값을 바꿔주더라도 TwoPoint클래스의 print만 출력되는 문제가 생겼는데 가상함수로 선언하니 원하는 값을 얻을 수 있었다. 클래스의 상속관계가 더욱 복잡해지고 많아지면 비슷한 동작을 하는 함수들이 많아질텐데 그때 사용하면 편리할 것 같다.

2. [Inheritance] Table1의 멤버변수들을 가지는 Figure 1의 3개의 Class를 구현하고 Table 2의 3개의 command를 가지는 학생정보관리 프로그램을 구현하시오. 멤버함수는 자유롭게 구현할 수 있지만 생성자를 통해 멤버변수들의 값을 저장해야 하며 반드시 Class의 멤버변수들을 이용해야 한다.



<Figure 1>

멤버함수	저장 정보
U_Name	대학명
Year	학년
U_StuID	학번

Object_Oriented_Programing	OOP의 성적
Digital_Logic_Circuit	DLC 성적
Name	이름
age	나이

<Table 1>

Command	사용법	설명
INSERT	INSERT	새로운 학생 정보를 입력 정보는 결과화면과 같이 학교, 학번, 학년, OOP 학점, DLC 학점, 이름, 나이를 입력받아 저장한다. 최대 100 개의 정보를 저장하며 그 이상의 정보가 들어올 경우 무시한다.
FIND	FIND 이름 학번	저장된 정보 중 이름과 학번이 맞는 정보가 있는지 검색하고 맞는 정보가 있을 경우 학교, 학번, 학년, OOP 학점, DLC 학점 순으로 출력한다
EXIT	EXIT	Program 종료

<Table 2>

학생정보는 학생정보 class 에 저장되어 있으며. 학생정보** type 으로 학생정보 class를 관리한다

문제 설명: 대학, 과, 학생정보순으로 상속되어 있는 클래스를 만들고 학생정보 포인터를 이용하여 데이터에 접근한다.

필요한 개념: 클래스의 상속, 생성자의 활용

```
Microsoft Visual Studio 디버그 콘솔
INSERT
학교 :광운대학교
학번 :2019202050
학년 :2
객체지향프로그래밍 학점 :A
디지털논리회로 학점 :B
이름 :이강현
나이 :23
INSERT
학교 :건국대학교
학번 :2022283938
학년 :1
객체지향프로그래밍 학점 :B
디지털논리회로 학점 :A
이름 :홍길동
나이 :20
FIND
이강현 2019202050
학교: 광운대학교
학번: 2019202050
학년: 2
객체지향프로그래밍 학점: A
디지털논리회로 학점: B
FIND 홍길동 2022283938
학교: 건국대학교
학번: 2022283938
학년: 1
객체지향프로그래밍 학점: B
디지털논리회로 학점: A
EXIT
```

<실행 결과>

고찰: 클래스의 상속이 줄줄이 되어있는 형태의 문제인데 자식클래스에서 부모클래스를 생성자를 통해 초기화하고 부모클래스의 멤버함수를 상속받아 멤버변수에 접근하는 방식을 사용했다. 생성자를 통해 자식클래스에서 부모클래스를 초기화하는방법을 알았고 상속개념에 대해 더 확장된 이해를 할 수 있었다.

3. [Inheritance]물품 판매 시뮬레이터를 구현하시오. 프로그램은 Figure 1과 같이 구현된 item class를 상속받는 Pen, Pencil, Eraser Class를 이용해 물품의 값, 재고, 할인율을 관리한다. 프로그램의 Command는 Table 1과 같으며 프로그램은 Command를 입력 받아 동작할 때 마다 매출을 출력한다. Pen, Pencil, Eraser Class는 멤버변수를 가지고 있지 않고 item의 멤버변수만 사용하며 item은 Figure 1외에 추가하지 않는다. 프로그램 시작시 모든 물품의 재고는 100개로 가정한다.

```
class item
{
private:
    int price;
    int stock;
    int discount;
public:
    item(int price, int stock) { this->price = price; this->stock = stock; discount = 0; }
    int getprice() { return price; }
    int getstock() { return stock; }
    int getdiscount() { return discount; }
    void addstock(int num) { this->stock += num; }
    void sell_item(int num) { this->stock -= num; }
    void setdiscount(int num) { discount = num; }
};
```

<Figure 1>

Command	사용법	설명
Sell	0 물품명 개수	<p>물품명에 맞는 물품을 개수*판매단위 만큼 판매하며 매출을 물품의 값 * 판매단위 * 개수*할인율만큼 더하고 재고를 개수*판매단위 만큼 뺀다.</p> <p>Pen 판매단위 : 3 개 가격 : 개당 200</p> <p>Pencil 판매단위 : 12 개 가격 : 개당 100</p> <p>Eraser 판매단위 : 1 개 가격 : 개당 500</p> <p>만약 개수*판매단위가 재고보다 많을 경우 명령을 무시한다.</p>
AddStock	1 물품명 개수	물품명에 맞는 물품의 재고를 개수만큼 더한다.

Discount	2 물품명 할인율	물품명에 맞는 물품의 할인율을 입력 받은 값으로 바꾼다
Print	3	모든 물품의 가격, 재고, 할인율을 출력
Quit	4	프로그램 종료

문제 설명: item이라는 클래스를 여러 클래스에서 모두 상속받은 후 적절하게 상속받은 클래스에서 item클래스의 멤버함수들을 이용하는 문제이다.

필요한 개념: 클래스의 상속

```
=====
-----Pen-----
Price : 600
Stock : 100
Discount : 0%
-----
-----Pencil-----
Price : 1200
Stock : 100
Discount : 0%
-----
-----Eraser-----
Price : 500
Stock : 100
Discount : 0%
=====
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 0 Pen 10
=====
Sales : 6000
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 0 Pencil 5
=====
Sales : 12000
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 0 Eraser 10
=====
Sales : 17000
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 3
=====
-----Pen-----
Price : 600
Stock : 70
Discount : 0%
-----
-----Pencil-----
Price : 1200
Stock : 40
Discount : 0%
-----
-----Eraser-----
Price : 500
Stock : 90
Discount : 0%
=====
=====
Sales : 17000
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 1 Pen 50
=====
Sales : 17000
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 1 Pencil 100
=====
Sales : 17000
```

```

Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 1 Eraser 10
=====
Sales : 17000
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 3
=====
-----Pen-----
Price : 600
Stock : 120
Discount : 0%
-----
-----Pencil-----
Price : 1200
Stock : 140
Discount : 0%
-----
-----Eraser-----
Price : 500
Stock : 100
Discount : 0%
=====
=====
Sales : 17000
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 2 Pen 50
=====
Sales : 17000
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 0 Pen 10
=====
Sales : 20000
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 3
=====
-----Pen-----
Price : 600
Stock : 90
Discount : 50%
-----
-----Pencil-----
Price : 1200
Stock : 140
Discount : 0%
-----
-----Eraser-----
Price : 500
Stock : 100
Discount : 0%
=====
=====
Sales : 20000
Enter Any Command(0 : Sell, 1 : AddStock, 2 : Discount, 3 : Print, 4 : Quit): 4

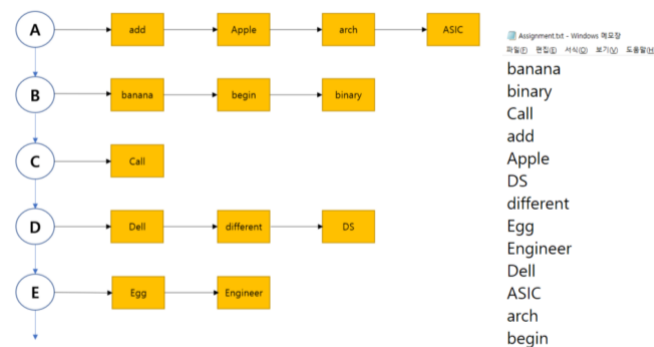
```

<실행 결과>

고찰: 위에서부터 순서대로 상속하여 내려주던 2번문제와는 다르게 1개의 클래스가 여러 개의 클래스에게 상속해주는 문제였다.

물품 판매라는 주제에 상속이란 개념은 다양한 품목에 동일하게 적용할 수 있는 가격, 할인을, 재고라는 정보를 물품들이 개별적으로 만들어야 하는 어려움을 해결해 주었고 실제 코드를 작성할 때 편리하게 사용할 수 있는 개념임을 보여줬다.

4. [Linked List] 2D Linked List를 이용한 단어장 프로그램을 구현하시오. 프로그램은 Assignment.txt로부터 단어를 읽어 Figure 1과 같이 알파벳순으로 1차원 Linked List를 구현하고 단어의 시작 알파벳과 같은 알파벳의 Node에 사전식 배열로 정렬되어 Linked List로 저장된다. 프로그램은 시작 시 Assignment.txt로부터 단어를 읽어서 정렬한 후 결과화면과 같이 저장된 단어가 존재하는 알파벳과 그 단어들을 출력한다. (프로그램 구현 시 파일명 Assignment.txt 준수)



<Figure 1>

문제 설명: linked list를 이용하여 메모장에서 읽어온 단어들을 사전식으로 정렬하여 출력하는 문제이다.

필요한 개념: linked list 사용법, 파일 스트림 사용법

```

Microsoft Visual Studio 디버그 콘솔
A->add->Apple->arch->ASIC
B->banana->begin->binary
C->Call
D->Dell->different->DS
E->Egg->Engineer
X->xbc->xdg
Z->zebra->zoo->zotopia

```

<실행 결과>

고찰: 단방향 연결 리스트와 양방향 연결리스트를 배웠었는데 이번 단어장 문제를 통해 2D로 리스트를 활용할 수 있다는 것을 배웠다.

메모장에서 불러와 매번 새로운 단어장을 구성하는 방식은 노드의 수정을 코드화해야 했던 방식보다 사용자에게 좀 더 편리했던 것 같다.