

객체지향프로그래밍

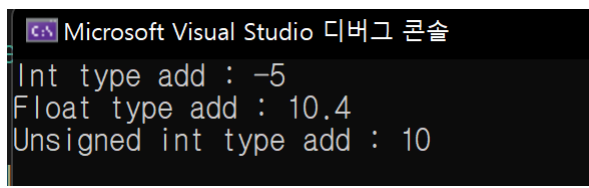
Assignment Report 2-4

2019202050 이강현

1. 아래의 main 함수가 동작해서 실행 예제와 같이 결과가 출력되도록 Overloading Class를 완성 하고 아래 main함수를 실행하는 프로그램을 구현하시오. main함수는 아래의 코드에서 수정하지 않는다.

필요한 개념: overloading이 무엇인지 알아야한다. Class의 사용법을 알아야한다.

문제 설명: main 함수부분의 코드를 살펴보면 같은 각기 다른 형의 변수들을 A클래스의 이름이 같은 함수의 인자로 각각 넣어주었다. 함수의 이름을 같게 하고 전달인자의 형을 다르게 하여 함수를 구현한다.

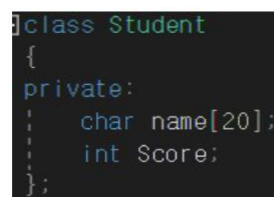


```
Microsoft Visual Studio 디버그 콘솔
Int type add : -5
Float type add : 10.4
Unsigned int type add : 10
```

<실행 예제>

고찰: add라는 함수의 전달인자가 int,float,unsigned int로 각각 다르지만 함수의 이름이 같아도 오류없이 올바른 결과 값을 얻을 수 있었다. 자료형이 같더라도 자료형의 수가 다르면 또한 정상적으로 작동한다. Overloading의 개념을 이해하기 좋은 예제였다.

2. 아래의 변수들을 가지는 Student class를 이용해 학생의 이름과 점수를 관리, 출력하는 프로그램을 구현하시오. 프로그램은 4개의 command(1 : insert, 2 : sort by name, 3 : sort by score, 4. quit)을 가지며 아래 예시와 같이 insert는 이름, 점수를 받아 저장하고 sort by name은 이름 순으로 출력, sort by score는 점수 순으로 출력, quit는 프로그램을 종료한다. Student class의 멤버변수는 추가할 수 없으며 멤버함수는 자유롭게 추가할 수 있다.



```
class Student
{
private:
    char name[20];
    int Score;
};
```

<Student class>

필요한 개념: class사용법, 정렬하는 알고리즘, 조건문의 분기등

문제 설명: 명령을 입력받을 변수를 하나 선언하고 숫자로 명령을 입력하여 조건에 맞게 해당 명령을 처리하고 클래스의 멤버변수에 값을 저장할 함수, 이름순으로 정렬할 함수, 점수순으로 정렬할 함수등을 멤버함수로 하여 프로그램을 구현한다.

```
Please Enter Command(1: insert, 2: sort by name, 3: sort by score, 4: quit) :1
Name :이강현
Score :90
Please Enter Command(1: insert, 2: sort by name, 3: sort by score, 4: quit) :1
Name :장형범
Score :95
Please Enter Command(1: insert, 2: sort by name, 3: sort by score, 4: quit) :1
Name :조현호
Score :80
Please Enter Command(1: insert, 2: sort by name, 3: sort by score, 4: quit) :1
Name :송채영
Score :70
Please Enter Command(1: insert, 2: sort by name, 3: sort by score, 4: quit) :1
Name :황세영
Score :100
Please Enter Command(1: insert, 2: sort by name, 3: sort by score, 4: quit) :2
송채영70
이강현90
장형범95
조현호80
황세영100
Please Enter Command(1: insert, 2: sort by name, 3: sort by score, 4: quit) :3
송채영70
조현호80
이강현90
장형범95
황세영100
Please Enter Command(1: insert, 2: sort by name, 3: sort by score, 4: quit) :4
C:\Users\WdlrkdW바탕 화면\과제\이강현\객프설\코드\Assignment2-4\Assignment4-2\Assi
이 창을 닫으려면 아무 키나 누르세요...■
```

<실행 예제 >

고찰: 조건에 부합할 때 해당 명령을 실행하는 흐름으로 프로그램을 구현하니 수월하게 문제를 해결할 수 있었다. 클래스배열에서 서로 문자열들을 비교하여 이름순, 점수순으로 정렬하는 게 이 문제의 핵심이었던 것 같다.

3. Linked List를 이용해 값을 저장, 탐색, 삭제, 출력하는 프로그램을 구현하시오. 프로그램은 5개의 command(insert, find, delete, print, quit)를 가지고 있으며 동작은 아래 표와 같다. 프로그램은 실행 예제와 같이 insert를 이용해 값을 저장, delete를 이용해 삭제하며 find를 이용해 값을 출력, quit를 이용해서 제거한다.

Command	Format	설명
insert	Insert 숫자	수를 Linked List에 저장
find	find 숫자	Linked List에 입력받은 수가 있을 경우 저장되어있는 Index를 모두 출력
delete	Delete 숫자	Linked List에 입력받은 수가 있을 경우 모두 제거
print	print	모든 data를 출력
quit	quit	프로그램 종료

<Command List>

필요한 개념: linked list의 개념(노드 찾기, 생성, 추가, 삭제등)

문제 설명: linked list를 이용하여 숫자들을 저장하기 위해 데이터와 다른 노드의 주소값을 가지는 노드 구조체를 형성한 후 클래스에서 노드 멤버변수를 선언하여 값을 저장하고 문제와 같은 명령들을 실행한다.

```

Microsoft Visual Studio 디버그 콘솔
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : insert 1
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : insert 3
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : insert 2
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : insert 8
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : insert 4
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : insert 6
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : insert 2
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : insert 1
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : print
1
3
2
8
4
6
2
1
Number of nodes in the list: 8
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : find 1
1
8
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : delete 2
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : print
1
3
8
4
6
1
Number of nodes in the list: 6
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : quit
C:\Users\WdIrkdW바탕 화면\과제\OI강현\객프설\코드\Assignment2-4\Assignment4-3\WAss
이 창을 닫으려면 아무 키나 누르세요...

```

고찰: 노드를 생성하고 추가하고 삭제하는 법이 많이 헷갈리고 어려웠다. 노드끼리 연결되어 있는 꼴이라 접근하는 법이 명확했지만 인덱스를 이용하여 접근하는 배열과 비교했을 때 바로 접근하지 못하고 헤드부터 조건문을 이용하여 접근해야 하는 점이 탐색속도에서 배열보다 밀린다는 생각을 했다. 하지만 데이터의 추가삭제가 용이하고 한번 생성되면 공간만이 남아 비효율적인 배열과는 달리 동적할당을 이용하여 메모리를 필요할 때마다 추가 삭제할 수 있다는 점이 linked list의 장점이라고 생각했다.

4. Linked List를 이용해 끝말잇기 프로그램을 구현하시오. 프로그램은 사용자로부터 단어를 입력 받아 끝말이 이어지는지 아닌지, 이전에 입력을 받았는지 받지 않았는지 검사해서 조건에 만족할 경우 단어를 저장한다. 단 단어의 대,소문자는 구별하지 않고 같은 알파벳으로 취급한다.

필요한 개념: 클래스 사용법, linked list의 사용법, 조건문 사용

문제 설명: linked list를 이용하여 각 노드에 데이터를 저장할 것이고 그 데이터의 각 자리에 접근하여 앞 노드의 마지막 자리가 뒷 노드의 앞 자리와 같아야 한다는 조건문 하나, 모든 노드의 데이터에 접근하여 새로운 데이터와 비교하는 구문을 이용하여 중복 단어를 받지 않는다는 조건문 하나, 대소문자 구별이 없게 하기 위해 조건문에 데이터를 넣기 전에 데이터를 대문자 혹은 소문자로 모두 바꾸어주는 함수 하나등등을 이용하여 문제를 해결한다.

```
Microsoft Visual Studio 디버그 콘솔
CMD(Word/exit)>> tower
tower ->
CMD(Word/exit)>> rabbit
tower -> rabbit ->
CMD(Word/exit)>> tower
Already Exists
tower -> rabbit ->
CMD(Word/exit)>> apple
Not chained
tower -> rabbit ->
CMD(Word/exit)>> toilet
tower -> rabbit -> toilet ->
CMD(Word/exit)>> Tower
Already Exists
tower -> rabbit -> toilet ->
CMD(Word/exit)>> TALL
tower -> rabbit -> toilet -> TALL ->
CMD(Word/exit)>> EXit

C:\Users\wdlrkd\바탕 화면\과제\이강현\객 프
개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요....
```

<실행 결과>

고찰: 노드의 데이터를 추가하는데 무조건 뒤에만 추가하면 되었기에 데이터의 추가는 구현이 비교적 수월했으나 데이터에 접근하여 끝 자리까지 탐색하는 코드가 핵심이었던 것 같다. 맨 마지막 노드는 null을 가리킨다는 것을 이용하여 데이터를 끝까지 탐색하는 방법을 배웠고 오버로딩을 이용하여 소문자로 데이터를 변환해가며 오버로딩에 대한 개념도 잡을 수 있었다.