

객체지향프로그래밍실습 2-4

학부: 컴퓨터정보공학부

학번: 2019202021

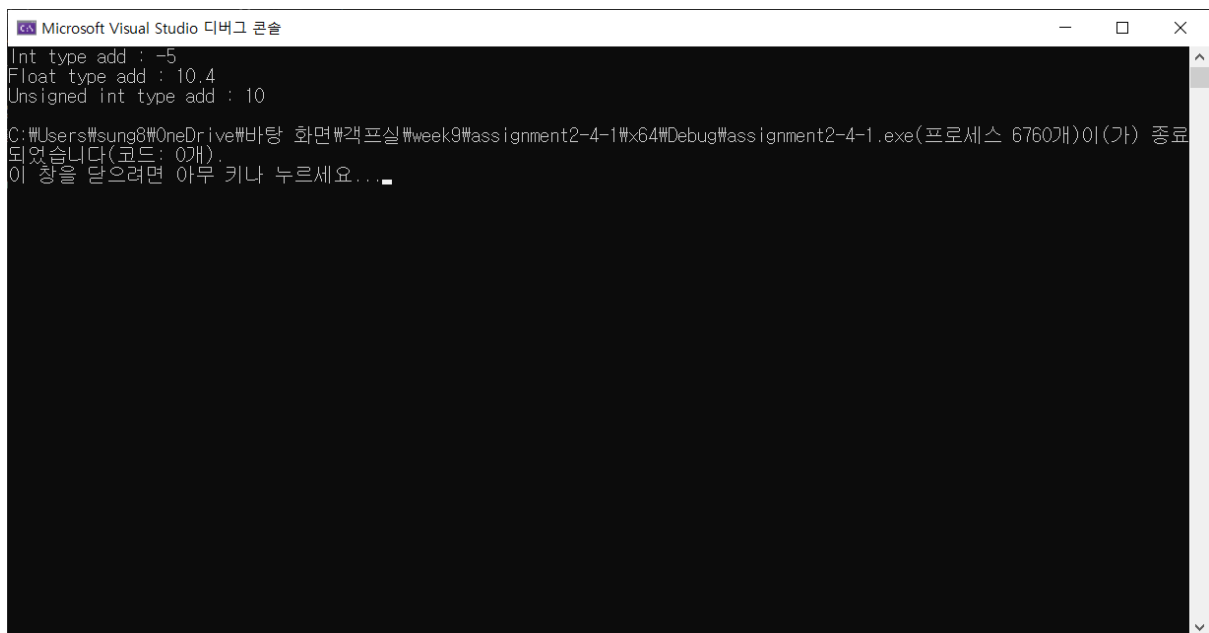
이름: 정성엽

<1번 문제>

1. 문제설명

- 주어진 아래의 main 함수가 실행 예제와 같이 결과가 출력되도록 Overloading Class 를 만들어야 한다. 이 때 class Overloading에 있는 add 멤버 함수가 입력되는 변수에 따라 다르게 입력 받고 계산하게 해야 한다.

2. 결과화면



```
Microsoft Visual Studio 디버그 콘솔
Int type add : -5
Float type add : 10.4
Unsigned int type add : 10
C:\Users\#sung8#OneDrive#바탕 화면#객프실#week9#assignment2-4-1#x64#Debug#assignment2-4-1.exe(프로세스 6760개)이(가) 종료
되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요....
```

- 코드를 실행했을 때 결과 화면.

3. 고찰

- 멤버 함수 add의 매개 변수를 각각 int, float, unsigned int로 각각 설정한 다음 main 함수 그대로 값이 입력되도록 했다. 멤버 함수는 void로 선언하여 따로 return 값이 없이 함수에서 각각 출력되도록 프로그래밍했다.

<2번 문제>

1. 문제설명

- 문제에서 주어진 변수들을 가지는 Student class를 이용해 학생의 이름과 점수를 관리, 출력하는 프로그램을 구현해야 한다. 4개의 command(1. insert, 2. sort by name 3. sort by score, 4. quit)를 받았을 때 insert는 이름 점수를 받아서 저장, sort by name은 이름의 오름차순, sort by score은 점수의 오름차순, quit는 프로그램 종료이다. name 같은 경우 한글로 받아도 영문자 오름차순 했을 때처럼 문자를 각각 비교하여 오름차순으로 정렬하도록 하고, score은 낮은 것부터 오름차순 정렬을 하면 된다.

2. 결과화면

- 각각 이름과 점수를 저장한 후 정렬했을 때 결과화면

3. 고찰

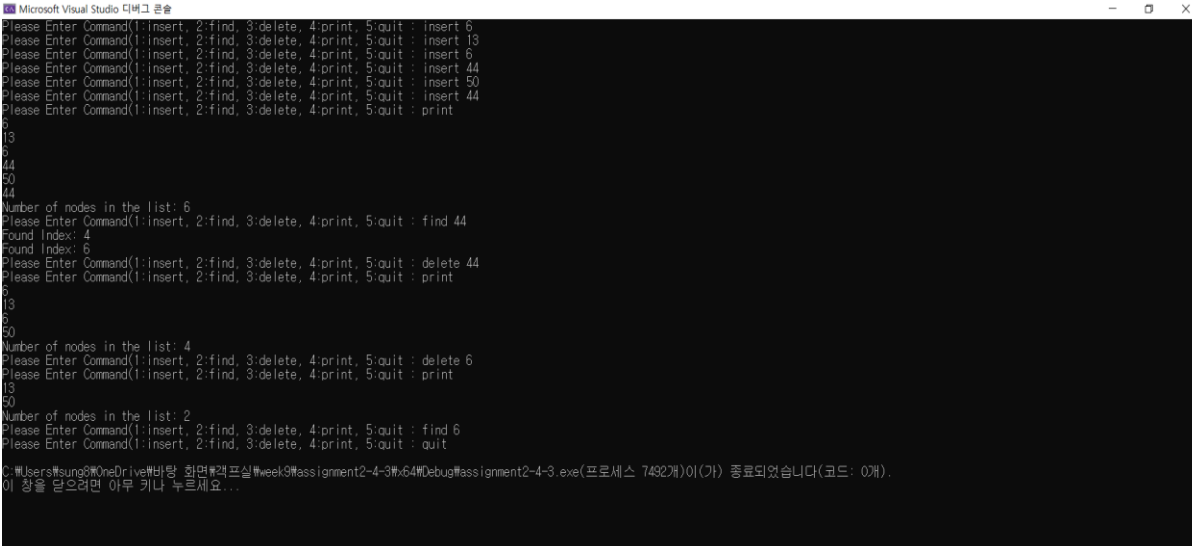
- 이전의 2-3과제의 4번 문제처럼 Student class를 두고 따로 관리하는 manage class를 두어 student class에 접근하도록 했다. Student class에는 단순 값 저장을 하는 함수나 저장된 값을 불러오는 함수를 만들었으며, manage class에서 이 저장하는 과정, 이름 순으로 오름차순으로 정렬하는 과정, 점수 순으로 오름차순으로 정렬하는 과정을 실행할 수 있도록 프로그래밍 했다. 처음에는 각 student list에 저장된 것들을 따로 가져와서 그 안에서 정렬한 것을 출력하는 프로그램을 만들려고 했으나 그것보단 student에서 바로바로 순서를 바꾸는 것이 수월하여 직접 바꾸도록 프로그램을 바꾸어 진행하였다. 마지막으로 각 class 새로 할당하면서 생긴 list를 delete를 통해 해제하였다. 이번 문제 같은 경우 command를 숫자로 부르게 하였다.

<3번 문제>

1. 문제설명

- Linked list를 이용해 값을 저장, 탐색, 삭제, 출력하는 프로그램을 구현하는 것으로 각 노드를 새로 만들어가며 저장을 하고, 원하는 수를 노드를 탐색하며 찾고, 원하는 수를 가진 노드를 삭제, 그리고 전체 출력할 수 있도록 한다. 객체지향프로그래밍설계에 서의 수업과 같이 class Node를 만들고 이를 접근하는 class Link를 만들어 값의 저장, 탐색, 삭제, 출력하도록 한다. quit를 할 경우 새로 만들어낸 노드들을 모두 delete 하도록 한다.

2. 결과화면



```
Microsoft Visual Studio 디버그 콘솔
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 13
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 44
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : insert 50
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : insert 44
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : print
6
13
6
50
44
Number of nodes in the list: 6
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : find 44
Found index: 4
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : delete 44
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : print
6
13
6
50
Number of nodes in the list: 4
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : delete 6
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : print
13
50
Number of nodes in the list: 2
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : find 6
Please Enter Command(1:insert, 2:find, 3:delete, 4:print, 5:quit) : quit
C:\Users\Sung\OneDrive\바탕 화면\백프실\week9\assignment2-4-3\64\Debug\assignment2-4-3.exe (프로세스 7432개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

- 여러 수들을 insert 하고 find, delete, print 했을 때 결과

3. 고찰

- 처음 써보는 linked list에서 새로운 노드를 생성하여 새로운 값을 저장하는 것, 만들어진 노드에서 원하는 값을 찾아내는 것, 모든 노드를 출력하는 것들은 비교적 이전에 프로그래밍하던 방법과 비슷하여 linked list에서 값을 불러오는 것만 제대로 알아낸다면 어렵지 않은 문제였지만 삭제하는 것은 해당 노드를 삭제할 뿐만 아니라 이전의 노드와 그 다음 노드를 새로 이어줘야 했다. 특히 1번 index의 노드를 삭제할 경우 예외 처리하여 head와 다음 노드를 이어주도록 해야 했다. 또한 삭제했을 경우 1을 반환하여 한 번 더 반복할 수 있도록 하고, 0일 반환된 경우 더 이상 삭제할 것이 없다는 것으로 설정하여 반복문을 탈출하도록 했다. 추가적으로 노드가 추가될 때는 개

수를 카운트하였으며 삭제했을 때는 개수를 차감하였다.

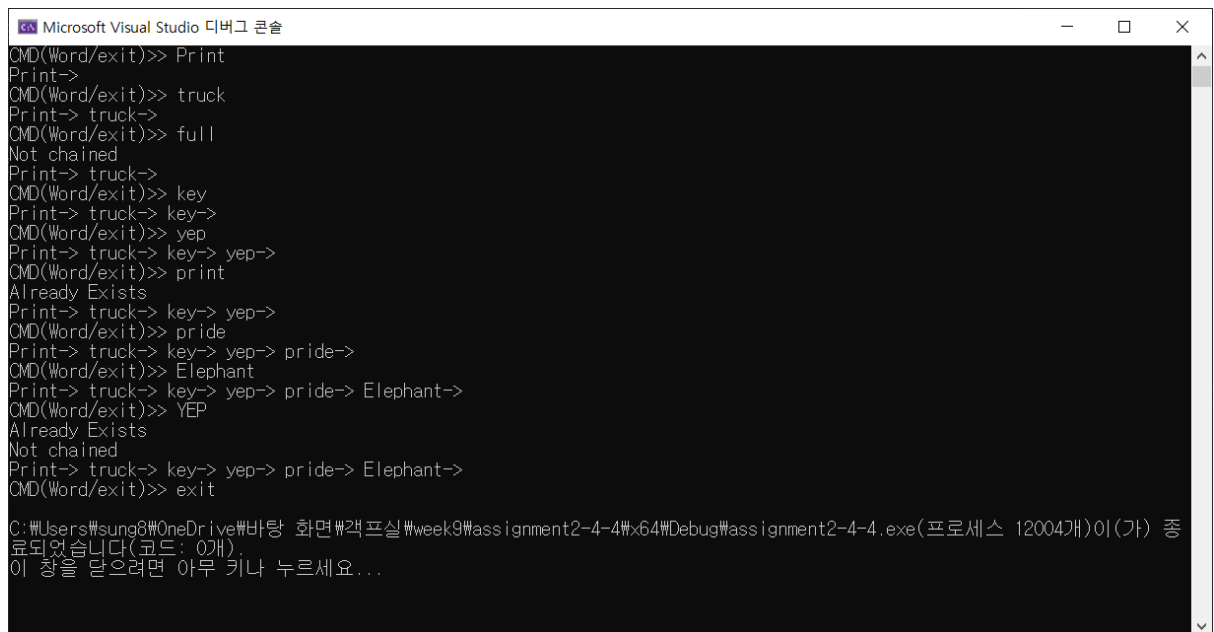
특히 이 linked list에서 중요한 것은 class node에 Node* next;를 지정하여 Node에 접근하는 class Link에서 다음 노드를 지정할 때 ~~~->next를 하여 다음에 접근하도록 했다. 마지막으로 quit를 할 경우 새로 할당한 노드들을 모두 delete를 통해 삭제하도록 했다.

<4번 문제>

1. 문제설명

- 앞서 사용했던 linked list를 이용해 끝말잇기 프로그램을 만드는 것으로 프로그램은 사용자로부터 단어를 받아 끝말이 이어지는지 확인하고, 대소문자 구별하지 않고 이전에 같은 입력을 했는지 확인한다. 입력과 전체 출력은 3번문제와 동일하게 하며, 이전과 이어지는 지 확인하는 것은 이전 입력을 미리 따로 저장하여 비교하도록 한다. 단어가 입력이 되면 잘못된 입력이든 잘 된 입력이든 모두 전체 출력하도록 한다.

2. 결과화면



```
Microsoft Visual Studio 디버그 콘솔
CMD(Word/exit)>> Print
Print->
CMD(Word/exit)>> truck
Print-> truck->
CMD(Word/exit)>> full
Not chained
Print-> truck->
CMD(Word/exit)>> key
Print-> truck-> key->
CMD(Word/exit)>> yep
Print-> truck-> key-> yep->
CMD(Word/exit)>> print
Already Exists
Print-> truck-> key-> yep->
CMD(Word/exit)>> pride
Print-> truck-> key-> yep-> pride->
CMD(Word/exit)>> Elephant
Print-> truck-> key-> yep-> pride-> Elephant->
CMD(Word/exit)>> YEP
Already Exists
Not chained
Print-> truck-> key-> yep-> pride-> Elephant->
CMD(Word/exit)>> exit

C:\Users\sung8\OneDrive\바탕 화면\객체프실\week9\assignment2-4-4\Debug\assignment2-4-4.exe (프로세스 12004개)이 (가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

3. 고찰

- 이전에 직접 구현할 때는 -32 또는 +32하여 대소문자 구별을 하였으나 이번 프로그램에서 새롭게 사용한 것은 _stricmp로 이는 문자열을 비교할 때 대소문자를 구별하지 않고 구별하도록 한다. 그래도 입력되었던 마지막 단어의 마지막 문자와 입력되는 첫번째 문자를 구별할때는 모든 경우의 수를 제대로 비교할 수 있도록 각각 +32 또는 -32 또는 아무것도 안 했을 때, 그리고 처음 입력되었을 때를 예외로 두어 프로그램이 진행되도록 했다. 이미 존재하는 것을 체크할 때는 bool 변수를 활용하여 이미 있는 것이면 true 없으면 false로 구별하도록 했다.