

객체지향프로그래밍 및 실습

OOP 2-4 과제

학과 : 컴퓨터정보공학부

학번 : 2021202043

이름 : 이은서

제출일 : 2022.05.04 (수)

문제 1.

1. 문제 설명

실행 예제와 같이 결과가 출력되도록 overloading 개념을 이용하여 클래스를 구현하는 문제이다.

overloading은 함수 명이 같더라도 변수 자료형 타입이 다르면 각각 다르게 읽혀서 문제없이 코드가 돌아간다. 즉, 함수 이름은 같지만 매개변수와 리턴 자료형만 다르게 구현하면 된다.

2. 결과 화면

```
"/Users/ieunseo/Downloads/practice C++/cmake-build-debug/2-4-1"
Int type add : -5
Float type add : 10.4
Unsigned int type add : 10

종료 코드 0(으)로 완료된 프로세스
```

3. 고찰

overloading 개념이 조금 생소했지만, 매개변수의 자료형만 다르게 해도 코드가 읽힌다는 것을 새롭게 알게되었다. 하지만, 함수의 이름을 각각 다르게 붙여서 코드를 활용하는 것이 훨씬 가독성이 좋을 것 같다.

문제 2.

1. 문제 설명

Student class를 이용하여 학생의 이름과 점수를 관리하고 출력하는 프로그램을 구현한다. 클래스를 잘 이용하여 학생의 이름을 입력받는 함수, 점수를 입력받는 함수, 출력하는 함수 등등을 각각 구현하면 된다.

2. 결과 화면

```

"/Users/ieunseo/Downloads/practice C++/cmake-build-debug/2-4-2"
Please Enter Command(1.insert, 2.sort by name, 3. sort by score, 4.quit) : 1
Name : 이은서
Score : 90
Please Enter Command(1.insert, 2.sort by name, 3. sort by score, 4.quit) : 1
Name : 홍길동
Score : 40
Please Enter Command(1.insert, 2.sort by name, 3. sort by score, 4.quit) : 1
Name : 고길동
Score : 75
Please Enter Command(1.insert, 2.sort by name, 3. sort by score, 4.quit) : 2
고길동 75
이은서 90
홍길동 40
Please Enter Command(1.insert, 2.sort by name, 3. sort by score, 4.quit) : 3
홍길동 40
고길동 75
이은서 90
Please Enter Command(1.insert, 2.sort by name, 3. sort by score, 4.quit) : 4

종료 코드 0(으)로 완료된 프로세스

```

3. 고찰

각각 기준에 따라 정렬하는 부분이 막혔었지만, 각각의 이름과 점수를 따로 불러오는 함수를 이용하는데, 서로의 값을 바꿔 정렬한 다음 출력하도록 하였다. 모든 정보를 한번에 입력했더니 어려워서 이름이면 이름, 점수면 점수대로 각각 저장하도록 하는 함수를 만든 다음 그 함수들을 사용했다.

문제 3.

1. 문제 설명

링크드리스트를 활용하여 값을 저장하고, 탐색, 삭제, 출력하는 프로그램을 구현하고자 한다. 5개의 명령어에 따라 숫자를 저장하거나 찾거나 삭제해야하기 때문에 명령어를 공백 기준으로 나눠준 다음 입력한다.

2. 결과 화면

```

"/Users/ieunseo/Downloads/practice C++/cmake-build-debug/2-4-3"
Please Enter Command(1: insert, 2: find, 3: delete, 4: print, 5: quit) : insert 5
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 2
Please Enter Command(1: insert, 2: find, 3: delete, 4: print, 5: quit) : insert 5
Please Enter Command(1: insert, 2: find, 3: delete, 4: print, 5: quit) : print
5
13
2
5
Number of nodes in the list : 4
Please Enter Command(1: insert, 2: find, 3: delete, 4: print, 5: quit) : find 5
Found Index : 1
Found Index : 4
Please Enter Command(1: insert, 2: find, 3: delete, 4: print, 5: quit) : delete 5
Please Enter Command(1: insert, 2: find, 3: delete, 4: print, 5: quit) : print
13
2
Number of nodes in the list : 2
Please Enter Command(1: insert, 2: find, 3: delete, 4: print, 5: quit) : find 5
Please Enter Command(1: insert, 2: find, 3: delete, 4: print, 5: quit) : quit

종료 코드 0(으)로 완료된 프로세스

```

3. 고찰

find함수에서 같은 숫자가 중복으로 여러 개 있는 경우 맨 위의 인덱스만 찾고 그 다음부터는 돌지 않는 문제가 생겼었다. 그래서 생각한 것이 find함수에서 찾으려고 하는 숫자를 찾은 다음에도 다음 next 값을 지정하게끔 하였더니 성공했다.

문제 4.

1. 문제 설명

링크드리스트를 이용하여 끝말잇기를 구현하는 문제이다. 끝말이 이어지는지 안이어지는지 확인해야하고, 이어지더라도 중복된 단어인지 아닌지를 확인한 후, 그 다음에 문자를 저장해야한다. 그리고, 저장 후에는 지금까지 저장된 단어들을 보여주면 된다.

2. 결과 화면

```
"/Users/ieunseo/Downloads/practice C++/cmake-build-debug/2-4-4"
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)>> exit

종료 코드 0(으)로 완료된 프로세스
```

3. 고찰

만약에 같은 단어를 입력했을 경우 판단해서 맞는지 안맞는지를 확인하는 부분과 마지막으로 저장된 단어의 끝글자와 입력할 단어의 첫 글자를 비교해야하는데, 그 과정에서 많은 시행착오를 겪었다. 그러다가 알게된 것이 링크드리스트는 처음부터 끝까지 차례대로 지나가면서 정보를 겪어야하고, 만약 특정 정보만 한번에 확인하고 싶으면 따로 주소 값을 저장하는 식으로 해야한다는 것이다.