

객체지향프로그래밍실습 3-1

학부: 컴퓨터정보공학부

학번: 2019202021

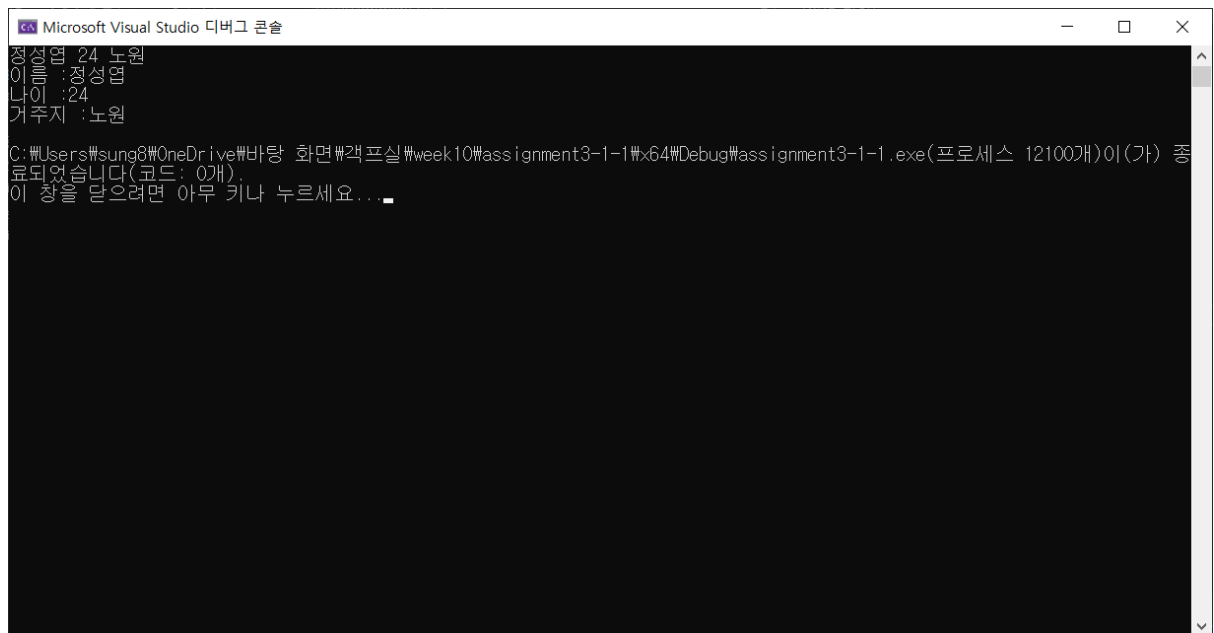
이름: 정성엽

<1번문제>

1. 문제설명

- Operation Overloading 문제로 class Person에 사람 이름, 나이, 거주지 정보를 가지고 있는 class를 만들고 main함수가 아래와 같을 때 동작하도록 한다. class 에 << 와 >>에 대한 operator를 두어 만들도록 한다.

2. 결과화면



```
Microsoft Visual Studio 디버그 콘솔
정성엽 24 노원
이름 :정성엽
나이 :24
거주지 :노원
C:\Users\sung8\OneDrive\바탕 화면\객 프 실\week10\assignment3-1\64\Debug\assignment3-1-1.exe(프로세스 12100개)이(가) 종
료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

3. 고찰

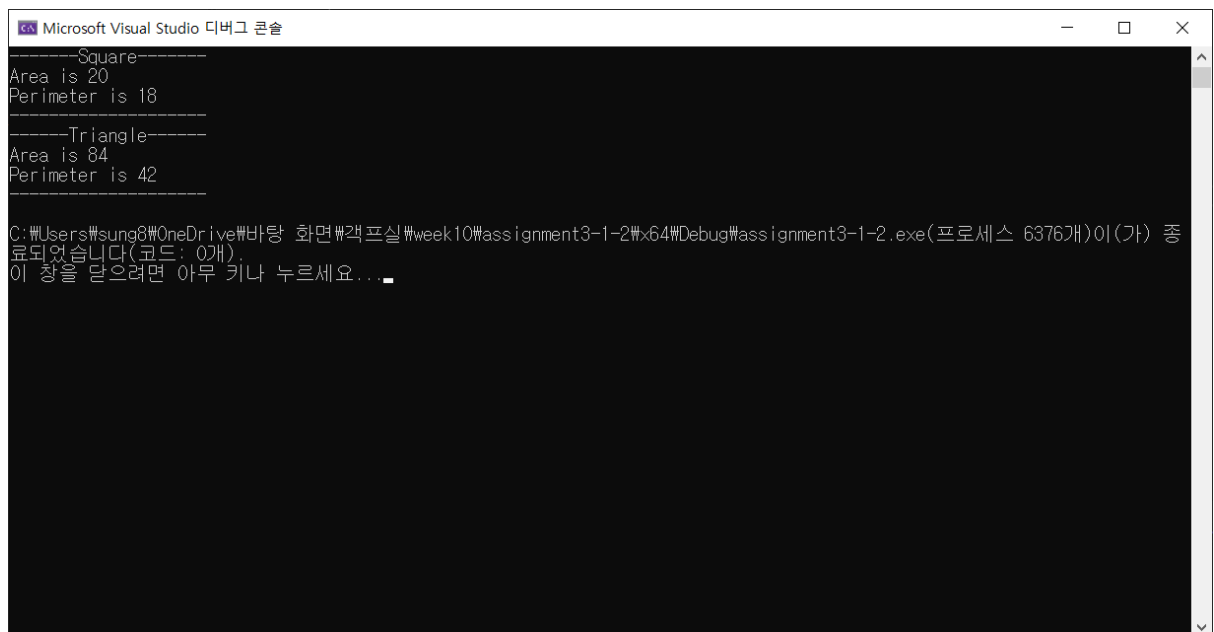
- 수업에서 들던 예시와 비슷하게 istream 그리고 ostream을 이용해 <<와 >>에 대한 operator를 만들고 이에 대해 각 매개변수에서 class Person을 P로 선언하여 input >> P.P_num과 같이 입력되도록 하고, output << P.P_num과 같이 출력되도록 했다. 여기서 중요한 것은 istream과 ostream을 friend 명령어를 통해 class Person의 private에 접근할 수 있도록 한다.

<2번문제>

1. 문제설명

- inheritance에 대한 문제로 class Polygon, Triangle, Square이 있으며 Polygon은 Triangle과 Square에 상속되어 있다. 각각의 class는 그림과 같은 멤버 변수와 멤버 함수만을 가지고 있으며, 생성자와 소멸자만 추가해서 프로그래밍한다. 또한 main 함수 또한 문제에 주어진 것처럼 만들어 Polygon이 정상작동 하는지 확인한다. 사각형 둘레는 $(a+b)*2$ 이고 넓이는 $a*b$ 이며 삼각형의 둘레는 $a+b+c$ 이고 넓이는 헤론의 공식인 $s=(a+b+c)/2$ 일 때, $\sqrt{s * (s - a) * (s - b) * (s - c)}$ 이다.

2. 결과화면



```
Microsoft Visual Studio 디버그 콘솔
-----Square-----
Area is 20
Perimeter is 18
-----Triangle-----
Area is 84
Perimeter is 42
C:\Users\sung8\OneDrive\바탕 화면\백프실\week10\assignment3-1-2\Debug\assignment3-1-2.exe(프로세스 6376개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

3. 고찰

- class의 상속관계를 이해하고 class Polygon이 class Triangle, Square에 상속되어 있으므로 class 선언할 때 `class Triangle : public Polygon`와 같이 선언하고, `Triangle(int a, int b, int c) : Polygon(a, b, c)`와 같이 생성자를 두었다. class Polygon에서는 Triangle과 Square을 구분하는 방법으로 매개변수 `c`가 NULL인지 아닌지 구분하여 작동하도록 만들었다. 추가적으로 가장 중요한 것은 Triangle이나 Square class에서 polygon의 매개함수를 쓸 수 있는 이유는 상속되어 있기 때문임을 알고 프로그래밍 하였다.

<3번문제>

1. 문제설명

- 1번 문제와 같이 Operation Overloading 문제로 operator가 +=일 때는 이어 붙이기 -=일때는 원하는 char 빼기, --일때는 마지막 문제 제거, &&일 때는 원하는 문자를 제외한 모든 문자를 0으로 바꾼다. class mystring은 private 권한의 char* 변수 그리고 생성자, 소멸자, print 함수만 가지도록 하되 추가적으로 operator를 설정한다. 또한 1. add, 2 delete_char 3. delete_lastchar 4. and_operator 5. print 6. exit 하도록 command를 입력 받아 동작하도록 한다.

2. 결과화면

```

선택 Microsoft Visual Studio 디버그 콘솔
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 1 lwang
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 1 woon
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 5
lwangwoon
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 2 w
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 5
lwangoon
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 3
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 5
lwango
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 1 operator
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 5
lwangooperator
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 4 a
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 5
0a00000000a000
Please Enter command(1: add, 2: delete_char, 3: delete_lastchar, 4: and_operator, 5: print, 6: exit): 6
C:\Users\wung0\OneDrive\바탕 화면\백프실\week10\assignment3-1-3\64\Debug\assignment3-1-3.exe (프로세스 7944개)이 (가) 종료되었습니다.(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요.

```

3. 고찰

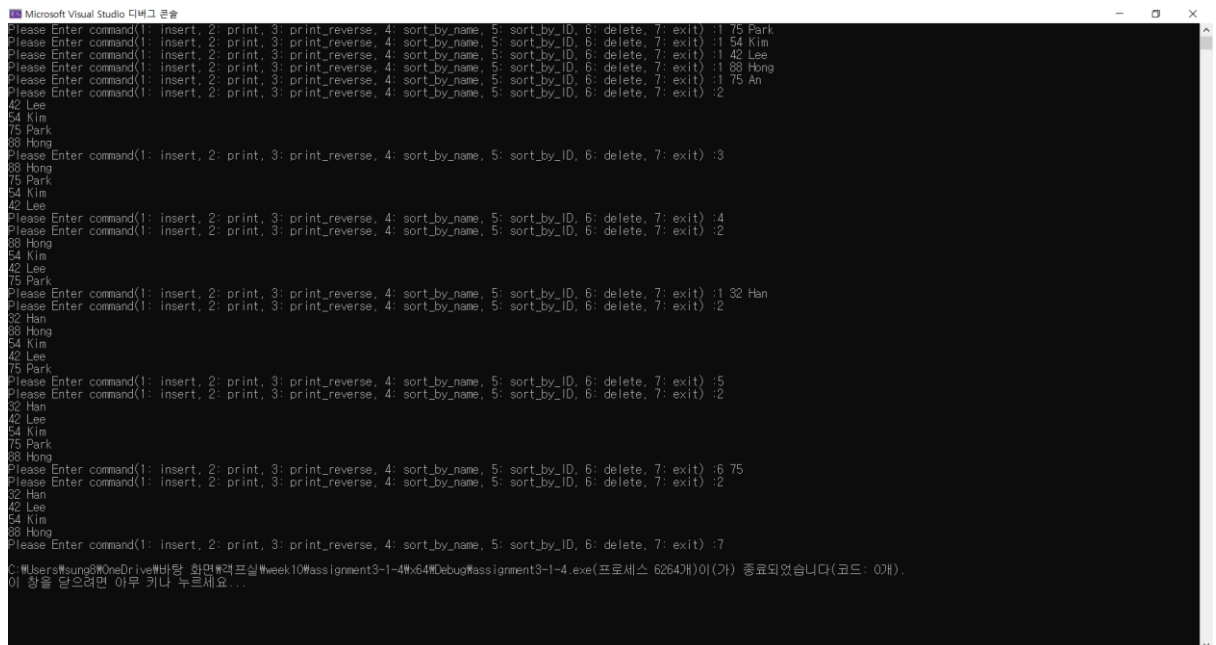
- operator 같은 경우 문제에 주어진 것과 같이 +=, -=, --, &&일 때 원하는 동작을 실행하도록 프로그래밍 했다. += 같은 경우 strcat이란 명령어를 이용하여 문장 뒤에 이어 붙였으며, -=는 처음 문자부터 끝까지 하나하나 비교하여 문자가 같은 경우 삭제하고 뒤에 있는 문장을 그대로 끌고 오도록 했다. --는 strlen을 이용하여 문장의 길이를 측정하고 마지막 문자에 해당되는 것을 'W0'로 만들었고, &&는 처음부터 끝까지 문자를 비교하여 다를 경우 모두 '0'으로 바꿨다. 추가적으로 operator도 class 내부에 포함되어야 하는지 헷갈렸으나 문제에서 operator에 대해서 table 1과 같은 동작을 하는 class이라고 했으니 해당 클래스에 operator를 두어 동작하도록 했다.
- 가장 힘들었던 것은 strcat의 오류로 생성자에서 string[0]을 'W0'로 초기화하고 소멸자에서 string을 delete 하여 해결하였다.

<4번문제>

1. 문제설명

- linked list 문제로 지난 과제에서는 next만 이용하여 다음 node를 정했다면 이번엔 next와 prev를 설정하여 다음 노드와 이전 노드 모두 정보를 저장하여 위치를 기억하도록 한다. 이를 doubly linked list라고 하며 앞선 설명과 같이 양방향 연결 구조이다. command에 따라 동작이 달라지며, insert는 새로운 정보 추가하고 같은 id인 경우 저장하지 않으며, id가 처음부터 시작해 자신보다 높은 id를 만나면 그 앞에 노드를 설정하도록 한다. print는 정방향 출력이고, print_reverse는 역순 출력이다. sort_by_name은 이름으로 오름차순하고, sort_by_ID는 ID로 오름차순한다. delete는 해당 id를 삭제한다.

2. 결과화면



```
Microsoft Visual Studio 디버그 콘솔
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :1 75 Park
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :1 54 Kim
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :1 42 Lee
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :1 88 Hong
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :1 75 An
42 Lee
54 Kim
75 Park
88 Hong
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :3
32 Han
54 Kim
75 Park
42 Lee
88 Hong
54 Kim
42 Lee
75 Park
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :1 32 Han
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :2
32 Han
54 Kim
42 Lee
75 Park
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :5
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :2
32 Han
54 Kim
42 Lee
75 Park
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :6 75
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :2
32 Han
42 Lee
54 Kim
88 Hong
Please Enter command(1: insert, 2: print, 3: print_reverse, 4: sort_by_name, 5: sort_by_ID, 6: delete, 7: exit) :7
C:\Users\Wung0\OneDrive\바탕 화면\프로젝트\week10\Assignment3-1-4\64\Debug\Assignment3-1-4.exe(프로세스 6264)이(가) 종료되었습니다(코드: 0).
0) 창을 닫으려면 아무 키나 누르세요...
```

3. 고찰

- doubly linked list인 만큼 이전 과제와 달리 노드를 생성할 때 추가적인 과정이 필요했다. 그래서 차라리 노드를 생성할 때 next 노드와 prev노드를 매개변수로 id와 이름과 같이 받도록 했다. 다만 중요한 것은 맨 앞일때, 맨 앞이 아닐때, 아무 ID도 입력된 것보다 크지 않을 때 3가지를 구분하여 앞뒤 연결을 설정하도록 프로그래밍했다. print는 head부터 출력하도록 했으며, print_reverse는 prev가 설정되어 있으므로 tail부터 출력하도록 했다. sort_by_name 같은 경우 노드 자체의 위치를 바꾸면 next와 prev의 설정도 모두 바꿔야했기에 차라리 노드의 정보를 서로 바꿔 오름차순 하도록 했다. sort_by_name도 마찬가지로 숫자를 기준으로 정보를 서로 바꿔 오름차순 하도록 했다. delete는 삭제할 id가 head일 때, tail일 때, 중간일 때를 기준으로 하여 삭제

하도록 했으며, next와 prev를 재설정하여 이어주었다. 또한 마지막으로 tail을 끝을 찾아 설정했다.

제일 중요한 것으로 문제 조건에 sort_by_name은 정렬한다만 있으므로 예제와 다르게 sort_by_name 했을 때 출력이 나오지 않도록 했다.