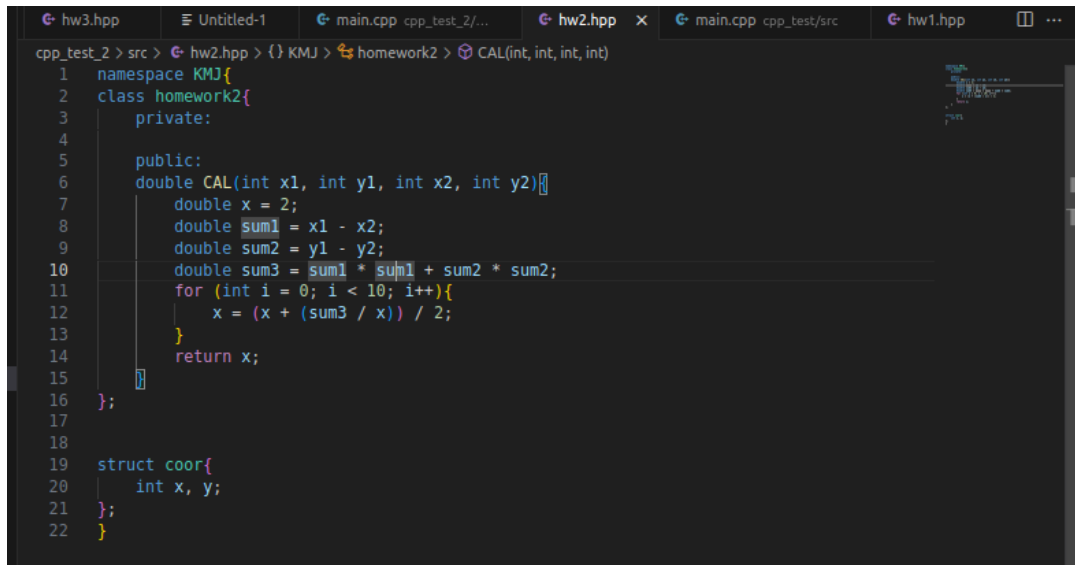


# C++ 1일차 보고서

## 목차

1. 과제2 코드 설명
2. 과제2 실행 결과 및 예외처리

## 1. 과제 2 코드 설명



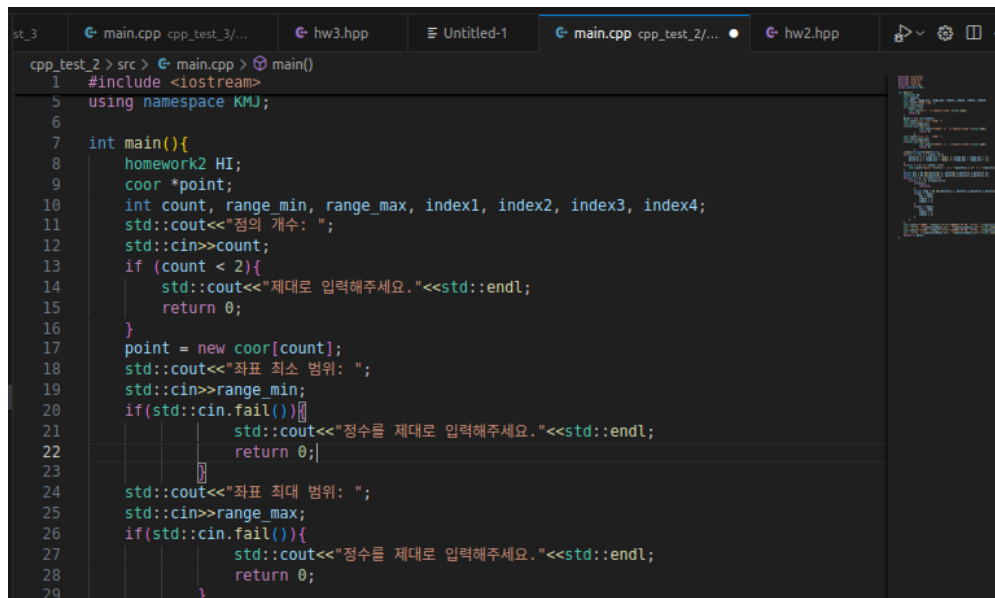
```
1 namespace KMJ{
2   class homework2{
3     private:
4
5     public:
6       double CAL(int x1, int y1, int x2, int y2){
7         double x = 2;
8         double sum1 = x1 - x2;
9         double sum2 = y1 - y2;
10        double sum3 = sum1 * sum1 + sum2 * sum2;
11        for (int i = 0; i < 10; i++){
12          x = (x + (sum3 / x)) / 2;
13        }
14        return x;
15      }
16    };
17
18
19    struct coor{
20      int x, y;
21    };
22  }
```

hpp 파일에 homework2 이름의 class 생성

두 점 사이의 거리를 구하는 함수 CAL 생성

이때 제곱근은 바빌로니아법을 통해 구함

구조체로 점의 x, y 좌표 나타낼 수 있게 coor 선언



```
1 #include <iostream>
2 using namespace KMJ;
3
4 int main(){
5   homework2 HI;
6   coor *point;
7   int count, range_min, range_max, index1, index2, index3, index4;
8   std::cout<<"점의 개수: ";
9   std::cin>>count;
10   if (count < 2){
11     std::cout<<"제대로 입력해주세요."<<std::endl;
12     return 0;
13   }
14   point = new coor[count];
15   std::cout<<"좌표 최소 범위: ";
16   std::cin>>range_min;
17   if(std::cin.fail()){
18     std::cout<<"정수를 제대로 입력해주세요."<<std::endl;
19     return 0;
20   }
21   std::cout<<"좌표 최대 범위: ";
22   std::cin>>range_max;
23   if(std::cin.fail()){
24     std::cout<<"정수를 제대로 입력해주세요."<<std::endl;
25     return 0;
26   }
27 }
```

cpp 파일에서 클래스 HI 를 선언

점의 개수를 사용자가 입력한 값인 count 만큼 동적할당

좌표 최소/최대 범위를 사용자에게 입력받음

```
30 srand((unsigned)time(NULL));
31 for(int i = 0; i < count; i++){
32     point[i].x = range_min + rand() % (range_max - range_min + 1);
33     point[i].y = range_min + rand() % (range_max - range_min + 1);
34 }
35 for(int i = 0; i < count; i++){
36     std::cout<<"point "<<i+1<<" . X = "<<point[i].x<<" Y = "<<point[i].y<<std::endl;
37 }
38 double max = HI.CAL(point[0].x, point[0].y, point[1].x, point[1].y);
39 double min = HI.CAL(point[0].x, point[0].y, point[1].x, point[1].y);
40 for(int i = 0; i < count; i++){
41     for(int j = 0; j < count; j++){
42         if(i==j){
43             continue;
44         }
45         double temp = HI.CAL(point[i].x, point[i].y, point[j].x, point[j].y);
46         if(max < temp){
47             max = temp;
48             index3 = i;
49             index4 = j;
50         }
51         if(min > temp){
52             min = temp;
53             index1 = i;
54             index2 = j;
55         }
56     }
57 }
58 std::cout<<"최솟값: "<<min<<"\n"<<"최솟값인 두 점: "<<"P1: "<<point[index1].x<< ", "<<point[index2].y<< "<<"P2: "<<point[index2].x<< ", "<<point[index2].y<< "<<std::endl;
59 std::cout<<"최댓값: "<<max<<"\n"<<"최댓값인 두 점: "<<"P1: "<<point[index3].x<< ", "<<point[index4].y<< "<<"P2: "<<point[index4].x<< ", "<<point[index4].y<< "<<std::endl;
60 delete [] point;
61 }
62 }
```

난수생성으로 사용자가 지정한 범위내에서 랜덤으로 x 좌표와 y 좌표 값이 구조체인 point 에 들어감

버블정렬을 통해 max 와 min 의 값을 저장하고 인덱스 또한 저장해 어느 점끼리의 거리인지 나타낼 수 있게함

최솟값, 최솟값인 두 점, 최댓값, 최댓값인 두 점 출력

마지막 delete [] point 로 메모리 해제

## 2. 과제 2 실행 결과 및 예외처리

```
kmj@kmj: ~/intern_ws/cpp_test_2/build
kmj@kmj: ~/intern_ws/cpp_test_2/build 96x55
kmj@kmj:~/intern_ws/cpp_test_2/build$ ./test
점의 개수: 1
제대로 입력해주세요.
kmj@kmj:~/intern_ws/cpp_test_2/build$ ./test
점의 개수: aaaawerq
제대로 입력해주세요.
kmj@kmj:~/intern_ws/cpp_test_2/build$ ./test
점의 개수: 5
좌표 최소 범위: 0
좌표 최대 범위: alw
정수를 제대로 입력해주세요.
kmj@kmj:~/intern_ws/cpp_test_2/build$ ./test
점의 개수: 10
좌표 최소 범위: 0
좌표 최대 범위: 20
point 1. X = 11 Y = 16
point 2. X = 20 Y = 15
point 3. X = 16 Y = 18
point 4. X = 9 Y = 15
point 5. X = 12 Y = 7
point 6. X = 14 Y = 0
point 7. X = 9 Y = 7
point 8. X = 9 Y = 0
point 9. X = 14 Y = 4
point 10. X = 20 Y = 19
최솟값: 2.23607
최솟값인 두 점: P1: <11, 16> P2: <9, 15>
최댓값: 21.9545
최댓값인 두 점: P1: <9, 0> P2: <20, 19>
kmj@kmj:~/intern_ws/cpp_test_2/build$
```

점 사이의 거리를 구하기 때문에 2 이상의 숫자를 받지 못했을 경우에 프로그램 종료

좌표 범위를 설정할 때 정수 외에 값이 들어가면 프로그램 종료

std::cin.fail()을 통해 정수형 데이터 입력받는 걸 실패했을 때 프로그램 종료 가능