

# C++ 1일차 보고서

## 목차

1. 과제1 코드 설명
2. 과제1 실행 결과 및 예외처리

# 1. 과제 1 코드 설명

```
cpp_test > src > E- hw1.hpp > {} KMJ > homework > number()
1 namespace KMJ{
2   class homework{
3   private:
4     int sum = 0;
5     int many = 0;
6     float avg = 0;
7     int *n;
8   public:
9     int number(){
10      std::cout<<"몇 개의 원소를 할당하겠습니까? : ";
11      std::cin>>many;
12      if(many <= 0){
13        std::cout<<"숫자를 잘못 입력하였습니다."<<std::endl;
14        return 0;
15      }
16      n = new int[many];
17      for(int i = 0; i < many; i++){
18        std::cout<<"정수형 데이터 입력: ";
19        std::cin>>n[i];
20        if(std::cin.fail()){
21          std::cout<<"정수형 데이터를 잘못 입력하였습니다."<<std::endl;
22          return 0;
23        }
24      }
25      return 1;
26    }
27    void SUM(){
28      for(int i = 0; i < many; i++){
29        sum += n[i];
30      }
31      std::cout<<"전체합: "<<sum<<std::endl;
32    };
33    void MAX(){
34      int max = n[0];
35      for(int i = 0; i < many; i++){
36        if(max < n[i]){
37          max = n[i];
38        }
39      }
40      std::cout<<"최대값: "<<max<<std::endl;
41    };
42    void MIN(){
43      int min = n[0];
44      for(int i = 0; i < many; i++){
45        if(min > n[i]){
46          min = n[i];
47        }
48      }
49      std::cout<<"최소값: "<<min<<std::endl;
50    };
51    void AVG(){
52      avg = (float)sum / (float)many;
53      std::cout<<"평균: "<<avg<<std::endl;
54      delete [] n;
55    };
56  };
57 }
```

hpp 파일에 homework 라는 class 생성

사용자가 입력한 many 만큼 정수배열 n 을 동적할당

int number(): 몇 개의 원소를 할당할지와 정수형 데이터를 입력받는 함수

void SUM(): many 만큼 반복하여 배열 n 의 모든 요소 합계 구하는 함수

void MAX(), void MIN(): 버블 정렬을 통해 최대/최소 값 구하는 함수

void AVG(): SUM()을 통해 구한 합계를 many 만큼 나눠 평균을 구하는 함수

delete [] n 으로 메모리 해제 => 평균 값 출력이 마지막이기에 AVG()에 메모리 해제 삽입

```
hw3.hpp  Untitled-1  main.cpp  cpp_test_2/...  hw2.hpp  main.cpp  cpp_test/src  hw1
cpp_test > src > main.cpp > ...
1  #include <iostream>
2  #include "hw1.hpp"
3  using namespace KMJ;
4
5  int main(){
6      homework CAL;
7      if(CAL.number() == 0){
8          return 0;
9      }
10     CAL.MAX();
11     CAL.MIN();
12     CAL.SUM();
13     CAL.AVG();
14 }
```

#include “hw1.hpp”를 통해 hpp 에 있는 class 를 cpp 에서 사용 가능

hpp 파일 전체를 namespace KMJ 로 묶었기에 using namespace KMJ 선언

이후 CAL 을 class homework 로 선언해주고 최대, 최소, 합, 평균순으로  
진행되게 함

## 2. 과제 1 실행 결과 및 예외처리

```
kmj@kmj: ~/intern_ws/cpp_test/build
kmj@kmj: ~/intern_ws/cpp_test/build 96x55
kmj@kmj:~/intern_ws/cpp_test/build$ make
Consolidate compiler generated dependencies of target test
[100%] Built target test
kmj@kmj:~/intern_ws/cpp_test/build$ ./test
몇 개의 원소를 할당하겠습니까? : -3
숫자를 잘못 입력하였습니다.
kmj@kmj:~/intern_ws/cpp_test/build$ ./test
몇 개의 원소를 할당하겠습니까? : asd
숫자를 잘못 입력하였습니다.
kmj@kmj:~/intern_ws/cpp_test/build$ ./test
몇 개의 원소를 할당하겠습니까? : 5
정수형 데이터 입력: aaaaa
정수형 데이터를 잘못 입력하였습니다.
kmj@kmj:~/intern_ws/cpp_test/build$ ./test
몇 개의 원소를 할당하겠습니까? : 4
정수형 데이터 입력: 1
정수형 데이터 입력: 2
정수형 데이터 입력: -5
정수형 데이터 입력: 4
최대값: 4
최소값: -5
전체합: 2
평균: 0.5
kmj@kmj:~/intern_ws/cpp_test/build$
```

원소를 할당할 때 0이하면 프로그램 종료

정수형 데이터를 입력받을 때 정수형 데이터 이외의 값을 받으면 프로그램 종료

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