

# Linux day2 hw1 결과 보고서

2025407012/로봇학부/송연우

## 목차

1. hpp파일
2. cpp파일
3. 실행 결과

## 1. hpp 파일

다음은 작성한 헤더파일 코드입니다.

```
1     namespace SYU{
2     class Jungsu{
3     private:
4         double n;
5         double *p;
6         int count;
7         int sum;
8         int max;
9         int min;
10        int check;
11
12
13    public:
14        Jungsu(); //생성자
15        ~Jungsu();
16        void getElement(); //다른 함수
17        int getMemo();
18        void Calculate();
19        bool CinSecure(double p);
20    };
21    }
```

namespace SYU 안에 클래스를 선언 코드를 작성하였습니다. Private 부분에 클래스 멤버 변수를, public 부분에 클래스 멤버함수를 작성해 cpp파일에서 클래스 객체 1개를 생성해 작동시킬 수 있

게 했습니다.

여기서 double로 선언된 \*p는 동적 할당을 사용하기 위해, n은 정수형 데이터 입력을 받을 때 실수형 데이터 입력을 방지하기 위해 만들어졌습니다. 나머지 int형으로 선언된 값들은 각각 동적 할당할 메모리 수, 합계, 최소/최댓값을 저장하고, check는 n과 같은 용도로 사용됩니다.

다음으로 함수의 간단한 역할에 대해서 설명하겠습니다.

Jungsu()함수는 생성자, ~Jungsu()함수는 소멸자로 생성자에서는 위에서 설정한 최댓값, 최솟값 등의 초깃값을 정해주고, 소멸자는 main문의 끝에서 메모리에 동적 할당 한 것을 소멸시킵니다. 나머지 함수 중 getElement()함수는 몇 개의 원소를 할당할 지 사용자에게서 입력받습니다. 이어서 getMemo()함수에서 메모리를 할당하고, 정수형 데이터를 입력받아 저장합니다.

CinSecure()함수는 bool형으로, if문에서 간편하게 사용하기 위해 만들어졌습니다. 정수형 데이터를 입력받을 때 double형으로 입력받은 것을 매개변수로 받아 정수형 변환 후 원래 값과 비교하여 다를 때 false를, 같을 때 true를 반환합니다.

Calculate함수에서는 sum, min, max값을 찾고 출력합니다.

## 2. cpp 파일

다음은 작성한 소스코드입니다. 먼저 namespace에서의 함수 정의를 설명한 다음, main문을 살펴 보겠습니다.

```
1  #include <iostream>
2  #include "hw1.hpp"
3
4  using namespace std;
5
6  namespace SYU{
7
8  Jungsu::Jungsu():n(0), p(nullptr), sum(0), max(0), min(0) {} //기본값설정..
9  Jungsu::~Jungsu() {delete [] p;}
10
11  void Jungsu::getElement(){
12      cout<<"몇 개의 원소를 할당하겠습니까? :";
13      cin>>n;
14      if(!CinSecure(n)){
15          cout<<"정수만 입력해주세요."<<endl;
16          return;
17      }
18      if(n<=0){
19          p=nullptr;
20          cout<<"자연수만 입력해주세요."<<endl;
21          return;
22      }
23      count=n;
24      p=new double[count];
25  }
```

using namespace std;로 cout, cin을 사용할 때 앞에 std::를 붙이는 번거로움을 줄였습니다.

Jungsu()함수에서는 기본값을, ~Jungsu()함수에서는 메모리를 해제합니다.

getElement()함수에서는 먼저 몇 개의 원소를 할당할지 입력 받고 사용자가 입력한 값이 정수가 아니면 오류 메시지를 출력합니다. 입력한 값이 음의 정수일 때에도 오류 메시지에 출력합니다. 값이 자연수일 때에만 int형 변수에 n값을 옮겨담고 메모리를 할당합니다.

```

27  ✓ int Jungsu::getMemo(){
28      if(!p){
29          cout<<"메모리를 할당할 수 없습니다."<<endl;;
30          return 0;
31      }
32      for(int i=0;i<count;i++){
33          cout<<"정수형 데이터 입력:";
34          cin>>p[i];
35          if(!CinSecure(p[i])){
36              cout<<"정수만을 입력할 수 있습니다. 다시 입력하세요"<<endl;
37              i--;
38              continue;
39          }
40      }
41      return 1;
42  }
43
44  ✓ bool Jungsu::CinSecure(double p){
45      check = p;
46      if(check!=p){
47          return false;
48      }
49      return true;
50  }

```

getMemo()함수와 CinSecure()함수입니다. getMemo()에서는 메모리 할당이 실패하였을 때 오류 메시지를 띄우고, 아니라면 정수형 데이터를 입력받습니다. 위에서와 마찬가지로 정수형 데이터를 입력받을 때에는 실수를 입력하지 않았는지 검사합니다. 메모리 할당이 실패하면 0을 반환하고, 아니라면 1을 반환합니다.

CinSecure()함수에서는 실수형 변수와 정수형 변수 값을 비교해 두 개가 다르다고 판정 되었을 때만 false를 반환합니다.

```

52  ✓ void Jungsu::Calculate(){
53      sum=0;
54      max=p[0];
55      min=p[0];
56      for(int i=0;i<n;i++){
57          sum+=p[i];
58          if(max<p[i]){
59              max = p[i];
60          }
61          if(min>p[i]){
62              min = p[i];
63          }
64      }
65      cout<<"최댓값="<<max<<endl;
66      cout<<"최소값="<<min<<endl;
67      cout<<"전체합="<<sum<<endl;
68      cout<<"평균="<<sum/n<<endl;
69  }
70  }
71  ✓ int main(){
72
73      while(1){
74          SYU::Jungsu num1;
75          num1.getElement();
76          if(num1.getMemo()==0)continue;
77          num1.Calculate();
78          break;
79      }
80
81      return 0;
82
83  }

```

---

Calculate()함수와 main함수입니다. Calculate()함수에서는 for문으로 sum값을 구하고, min, max값을 갱신해 출력합니다.

main함수에서는 num1로 객체 선언을 하고 getElement(), getMemo(), Calculate()함수 순으로 작동합니다. getMemo()에서 반환한 값에 따라 메모리 할당이 실패하면 다시 처음으로 돌아가고, Calculate()까지 오류없이 지나가면 종료합니다.

### 3. 실행 결과

```
yu@yu:~$ cd intern_ws/hw1/build
yu@yu:~/intern_ws/hw1/build$ ./test
몇 개의 원소를 할당하겠습니까? :4
정수형 데이터 입력:2
정수형 데이터 입력:4
정수형 데이터 입력:5
정수형 데이터 입력:6
최댓값=6
최소값=2
전체합=17
평균=4.25
yu@yu:~/intern_ws/hw1/build$ ./test
```

위는 정상 작동할 때의 실행 결과입니다.

```
yu@yu:~/intern_ws/hw1/build$ ./test
몇 개의 원소를 할당하겠습니까? :-1
자연수만 입력해주세요.
메모리를 할당할 수 없습니다.
몇 개의 원소를 할당하겠습니까? :0.004
정수만 입력해주세요.
메모리를 할당할 수 없습니다.
몇 개의 원소를 할당하겠습니까? :5
정수형 데이터 입력:4.5
정수만을 입력할 수 있습니다. 다시 입력하세요
정수형 데이터 입력:-1
정수형 데이터 입력:2
정수형 데이터 입력:3
정수형 데이터 입력:3
정수형 데이터 입력:3
최댓값=3
최소값=-1
전체합=10
평균=2
yu@yu:~/intern_ws/hw1/build$ ./test
```

예외처리 부분입니다. 음수를 입력하거나 실수를 입력하면 오류 메시지를 출력합니다.