

Homework #2

1. 1개-3개의 숫자를 입력해서 각각 선분(길이), 직사각형(가로, 세로), 직육면체(가로, 세로, 높이)에 해당하는 값을 가지고 있으며, 선분의 경우 길이, 직사각형의 경우 넓이, 직육면체의 경우 부피를 구할 수 있는 함수 spec()을 가진 Class를 가지고 아래 main() 함수를 작성하시오.

- 조건 -

- 1) 중복된 생성자를 이용
- 2) 입력 받는 숫자는 양의 정수로 제한
- 3) .cpp와 .h 파일은 분리하여 코드 작성

```
int main() {  
    int a1, a2, a3; // 입력 받는 최대 3개의 숫자, 입력 받는 부분은 생략하였음  
  
    // Class 이름은 Shape  
    // spec()은 선분은 길이, 직사각형은 넓이, 직육면체는 부피를 출력하는 함수  
  
    Shape S1(a1); // 생성자 : 선분  
    S1.spec(); // 선분의 길이 출력  
  
    Shape S2(a1, a2); // 생성자 : 직사각형  
    S2.spec(); // 직사각형의 넓이 출력  
  
    Shape S3(a1, a2, a3); // 생성자 : 직육면체  
    S3.spec(); // 직육면체의 부피 출력  
  
    return 0;  
}
```

2. 다음 정보(이름, 나이, 부서, 전화번호)를 가지고 있으며 이를 출력할 수 있는 Class와 표의 내용을 Class 배열을 통해 저장하고 배열의 순서대로 정보를 출력하는 코드를 작성하시오.

<그림 1>

| 이름 | 나이 | 부서 | 전화번호 |
|-----|----|-------|---------------|
| 김철수 | 23 | 인사팀 | 010-2342-1453 |
| 정연우 | 31 | 연구계획팀 | 010-4231-1241 |
| 박지운 | 35 | 인사팀 | 010-3421-6894 |
| 최지우 | 26 | 영업1팀 | 010-9653-3532 |

- 조건 -

- 1) 자료형으로는 나이는 int, 나머지는 문자열(c++ string class사용)
- 2) 나이, 이름, 부서, 전화번호는 class의 private에 선언(class 선언부에서 초기화 X)

```
class employee {
private:
    int age; // (O)
    int age = 23; // (X)
    ....
}
```

- 3) 내용 출력은 클래스 public 영역에 멤버 함수를 만들어 출력
- 4) class 배열은 동적 할당을 통해 생성
- 5) .cpp 파일과 .h 파일은 분리하여 코딩 (main.cpp, Employee.cpp, Employee.h)
- 6) 아래 헤더파일(h)에 정의된 생성자 및 함수들을 employee.cpp에 구현
- 7) Employee의 정보는 main.cpp에서 출력

```
// Employee.h
#ifndef EMPLOYEE_H
#define EMPLOYEE_H

#include <iostream>
#include <string>
using namespace std;

class Employee {
private:
    string e_name; // 이름
    int e_age;      // 나이
    string e_depart; // 부서
    string e_phone; // 전화번호

public:
```

```

Employee(); // 디폴트 생성자
Employee(string name, int age, string depart, string phone); // 매개변수 생성자

// 각각의 멤버 변수의 값을 반환해주는 함수
string getName();
int getAge();
string getDepart();
string getPhone();

// 각각의 멤버 변수의 값을 매개변수의 값으로 바꾸어주는 프로시저
void setName(string name);
void setAge(int age);
void setDepart(string depart);
void setPhone(string phone);

void printEmp(); // 이름, 나이, 부서, 전화번호를 출력해주는 함수
~Employee(); // 소멸자
}

```

```

#endif

```

```

// Employee.cpp

```

```

#include "Employee.h"

```

```

Employee::Employee() {
    ....
}

```

```

Employee::Employee(string name, int age, string depart, string phone_num) [
    ....
]

```

```

string Employee::getName() { .... }
int Employee::getAge() { .... }
string Employee::getDepart() { .... }
string Employee::getPhone() { .... }

```

```

void Employee::setName(string name) { .... }
void Employee::setAge(int age) { .... }
void Employee::setDepart(string depart) { .... }

```

```
void Employee::setPhone(string phone_num) { .... }
```

```
void Employee::printEmp() {  
    ....  
}
```

```
Employee::~Employee() { .... }
```

```
// main.cpp
```

```
#include "Employee.h"
```

```
int main() {  
    // Class 객체로 이뤄진 배열 생성(동적 할당) 및 초기화  
  
    // 배열의 길이는 4, 위의 직원들의 정보를 넣어 초기화  
  
    // class의 출력 프로시저를 통해 객체들의 정보를 나열해서 출력  
  
    // 종료할 때 동적 할당한 메모리의 반환을 확실히 할 것  
  
    return 0;  
}
```

3. 위 2번 문제에서 작성한 파일을 사용하여 아래의 코드를 main.cpp에서 구현하시오.

(1) 부서명을 문자열로 입력 받아서 해당 부서에 있는 직원들의 정보를 출력하시오. 만약 입력 받은 부서에 사람이 없으면 해당 부서에 직원이 존재하지 않는다고 출력하시오.

(2) 위 직원 중에 최지우 직원의 부서를 영업2팀으로 바꾼 후에 출력하시오.

```
// main.cpp
```

```
int main() {  
    // Class 객체로 이뤄진 배열 생성(동적 할당) 및 초기화  
  
    // 배열의 길이는 4, 위의 직원들의 정보를 넣어 초기화  
  
    // class의 출력 프로시저를 통해 객체들의 정보를 나열해서 출력  
  
    // 부서명을 문자열로 입력 받기  
  
    // (1) 문제  
    if() {  
        // 입력 받은 부서명이 객체 배열에 존재하면 해당 직원 정보 출력  
    }  
    else {  
        // 객체 배열에 존재하지 않으면 직원이 존재하지 않는다고 출력  
    }  
  
    // (2) 문제  
    // 이름이 최지우인 직원 객체를 클래스 배열에서 찾아서 부서를 바꾼 후 해당 객체의 정보(이름, 나이, 부서, 전화번호) 출력  
  
    // 종료할 때 동적 할당한 메모리의 반환을 확실히 할 것  
    return 0;  
}
```

4. 아래와 같은 새로운 직원 정보들을 기존에 동적 할당을 받은 배열에 추가하려고 한다.

<그림 2>

| 이름 | 나이 | 부서 | 전화번호 |
|-----|----|-------|---------------|
| 장지훈 | 40 | 영업2팀 | 010-8634-4577 |
| 오명길 | 45 | 연구계획팀 | 010-2421-2342 |

기존에 동적 할당 받은 객체 배열의 크기는 4이다. 새로운 직원의 정보를 추가하려면 새로운 크기의 객체 배열을 할당 받아 추가해야 한다. 새로운 직원 정보를 포함한 기존의 직원 정보까지 모두 출력하시오.

- 조건 -

- 1) 문제 2번의 직원 정보들<그림 1>을 크기가 4인 객체 배열을 동적 할당을 받아 저장
- 2) cin을 통해 새로운 추가된 직원들의 수를 기존의 직원 정보의 수를 합한 수를 입력 받아 새롭게 동적 배열 할당
- 3) 기존에 있던 직원들의 정보를 새롭게 할당 받은 배열에 복사
- 4) cin을 통해 새로운 직원 정보를 입력 받고 객체 배열에 저장
- 5) Employee class에 구현된 printEmp()를 통해 모든 직원 정보를 출력

// main.cpp

```
int main() {
    Employee *emp_list1 = new Employee[4];
    // Class 객체로 이뤄진 배열 생성(동적 할당) 및 초기화

    // 배열의 크기는 4, <그림 2> 직원들의 정보를 넣어 초기화

    // cin을 통해 객체 배열 크기 입력 받기

    Employee *emp_list2 = new Employee[ .... ];
    // 입력 받은 크기에 맞는 객체 배열 생성(동적 할당)

    // emp_list1에 있던 정보들 emp_list2에 복사

    // emp_list1의 동적 메모리 반환

    // 추가된 직원 정보를 cin을 통해 입력 받고 emp_list2에 넣기

    // class의 출력 프로시저를 통해 객체들의 정보를 나열해서 출력

    // 종료할 때 동적 할당한 emp_list2의 메모리의 반환을 확실히 할 것
```

```
    return 0;  
}
```