개요 | 제출 | 편집 | 코딩 결과

과제 092 - IOrganization

제출 마감일: 2023-06-02 23:59
업로드 가능한 파일 수: 6
제출 방식: 개인

Purpose

This problem is designed to test your understanding of OOP principles, including classes, inheritance, encapsulation, and polymorphism, as well as your ability to effectively use the C++ Standard Library, particularly smart pointers and the STL container classes.

Problem

You are working on an organization management software where you have a hierarchy of organization, departments, managers, and employees. The provided code is a simple simulation of this system. This is an Object-Oriented Programming (OOP) problem where you will be implementing various classes and their interactions.

The structure of the system is as follows:

- **Organization**: This is the highest-level object in the system. An organization contains multiple departments. The organization is responsible for managing departments and overall operations like adding or removing a department, setting a department manager, moving an employee from one department to another, and printing the structure of the organization.
- **Department**: A department is a part of an organization. Each department has one manager and may have multiple employees. The department is responsible for managing its employees, including adding and removing employees.
- **Manager**: A manager is an employee who is in charge of a department. A manager can perform all the functions of an employee and also has some additional responsibilities related to the management of the department. Note that in our system, a manager is also considered an employee of the department.
- **Employee**: An employee is the basic unit of the organization. An employee belongs to a specific department and has properties like name and salary.

IOrganization interface should declare the following methods:

- addDept(sDept dept): Adds a department to the organization.
- addEmpToDept(sDept dept, sEmp emp): Adds an employee to a specific department in the organization.
- setDeptManager(sDept dept, sManager manager): Sets a manager for a specific department in the organization.
- moveEmp(sDept &from_dept, sDept &to_dept, sEmp emp): Moves an employee from one department to another in the organization.
- print(): Prints the details of the organization, including its departments and their employees.

print(): Prints the details of the organization, including its departments and their employees.

Your tasks are:

- Complete the classes **Organization**, **Department**, **Manager**, and **Employee**. Ensure that all data members are private and can only be accessed or modified through appropriate member functions. You can also add additional data members or functions if required.
- Implement a function to create an organization.
- Implement a function to create departments and add them to an organization.
- Implement a function to create employees (including managers) and add them to departments.
- Implement a function to print the entire organization structure in a hierarchical manner. The organization prints its name, then each department prints its name and its manager, and then each employee (including the manager) in the department prints their name.
- Implement a function to move an employee from one department to another. This function takes as parameters the source department, the destination department, and the employee to be moved.
- Test all the implemented functionalities in the main() function using the provided skeleton code.
- **Provide a description of your solution**, detailing the methods you used to implement these features, the responsibilities and collaborations of each object, and any challenges you encountered along the way.

<참고자료>

//OrgTest.cpp

```
int main() {
    // Create the organization
    std::shared_ptr<IOrganization> org = std::make_shared<Organization>("PNU-CSE");

    // Create departments
    auto design = std::make_shared<Department>("Design");
    auto dev = std::make_shared<Department>("Development");
    auto qa = std::make_shared<Department>("QA");

    // Create managers
    auto lee = std::make_shared<Manager>("Lee", 8'000'000);
    auto park = std::make_shared<Manager>("Park", 9'000'000);
    auto bae = std::make_shared<Manager>("Bae", 5'000'000);

    // Add departments to the organization
    org->addDept(design);
    org->addDept(dev);
    org->addDept(qa);

    // Set managers for the departments
    org->setDeptManager(design, lee);
    org->setDeptManager(dev, park);
    org->setDeptManager(qa, bae);

    // Create employees
    auto kim = std::make_shared<Employee>("Kim", 4'000'000);
    auto go = std::make_shared<Employee>("Go", 8'000'000);
    auto jeon = std::make_shared<Employee>("Jeon", 5'000'000);
    auto yoon = std::make_shared<Employee>("Yoon", 3'000'000);

    // Add employees to the departments
    org->addEmpToDept(design, kim);
    org->addEmpToDept(dev, go);
    org->addEmpToDept(dev, jeon);
    org->addEmpToDept(qa, yoon);
```

```
    // Print the organization
    org->print();

    // Move an employee to a different department
    org->moveEmp(dev, qa, jeon);

    // Print the updated organization
    org->print();
    return 0;
}
```

//IOrganization.h

```
using sEmp = std::shared_ptr<Employee>;
using sDept = std::shared_ptr<Department>;
using sManager = std::shared_ptr<Manager>;

class IOrganization {
public:
    virtual ~IOrganization() {}
    virtual void addDept(sDept dept) = 0;
    virtual void addEmpToDept(sDept dept, sEmp emp) = 0;
    virtual void setDeptManager(sDept dept, sManager manager) = 0;
    virtual bool moveEmp(sDept &from_dept, sDept &to_dept, sEmp emp) = 0;
    virtual void print() const = 0;
};
```

입력

없음

출력

Organization: PNU-CSE

Design

|---Manager: Lee 1 8000000

|---Kim 4 4000000

Development

|---Manager: Park 2 9000000

|---Go 5 8000000

|---Jeon 6 5000000

QA

|---Manager: Bae 3 5000000

|---Yoon 7 3000000

Organization: PNU-CSE

Organization: PNU CSE

Design

|---Manager: Lee 1 8000000

|---Kim 4 4000000

Development

|---Manager: Park 2 9000000

|---Go 5 8000000

QA

|---Manager: Bae 3 5000000

|---Jeon 6 5000000

|---Yoon 7 3000000

제출파일

Employee.h
Manager.h
Department.h
Organization.h
p92.csv
description.txt

(OrgTest.cpp 와 IOrganization.h 파일은 PLATO 서버에 등록되어 있습니다.)