

Proszę założyć, że udostępniona jest przestrzeń nazw std

Zadanie 1 - obiektowe wzorce projektowe (2pkt)

Obiekty klas pochodnych po Staff reprezentują osoby zatrudnione w przedsiębiorstwie. Czasem życia zarządza obiekt typu HRSystem.

```
using Money = int;
using UStaff = unique_ptr<Staff>;
using VStaff = Staff*; //widok, nie zarządza obiektem, z mozliwoscia przechowywania w std::vector
// UWAGA, w C++17 lepiej 'using VStaff = reference_wrapper<Staff>', nie ma to wpływu na zadanie!;
using VEmployee = Employee*; //podobnie jak dla Staff
class Visitor {
public:
    virtual void visit(const Employee& e) = 0;
    virtual void visit(const Trainee& t) = 0;
    virtual void visit(const Manager& m) = 0;
};
class HRSystem {
public:
    HRSystem() : id_(0), baseSalary_(4242) {}
    VStaff createEmployee() { staff_.push_back( make_unique<Employee>(*this, ++id_) ); return staff_.back().get(); }
    VStaff createTrainee() { staff_.push_back( make_unique<Trainee>(*this, ++id_) ); return staff_.back().get(); }
    VStaff createManager() { staff_.push_back( make_unique<Manager>(*this, ++id_) ); return staff_.back().get(); }
    Money getBaseSalary() const { return baseSalary_; }
    void setBaseSalary(Money m) { baseSalary_ = m; }
    const vector<UStaff>& getStaff() { return staff_; }
private:
    int id_;
    Money baseSalary_;
    vector<UStaff> staff_;
};
```

Zmodyfikuj klasy pochodne po Staff aby można było reprezentować strukturę w firmie, gdzie mamy hierarchiczną strukturę, ale jej głębokość jest dowolna, tzn. kierownik może być podwładnym innego kierownika. Jaki wzorzec Jakiego wzorca użyłaś/ użyłeś?

```
class Staff {
public:
    Staff(HRSystem& hr, int id) : hr_(hr), id_(id) {}
    virtual ~Staff() {}
    virtual void accept(Visitor& v) const = 0;
    virtual void addSubordinate(const VEmployee& s) {}
    HRSystem& hr_;
    int id_;
};
class Employee : public Staff { // pracownik etatowy
public:
    Employee(HRSystem& hr, int id) : Staff(hr, id) {}
    void accept(Visitor& v) const override { v.visit(*this); }
};
class Trainee : public Staff { // stazysta
public:
    Trainee(HRSystem& hr, int id) : Staff(hr, id){}
    void accept(Visitor& v) const override { v.visit(*this); }
};
class Manager : public Staff { //kierownik
public:
    Manager(HRSystem& hr, int id) : Staff(hr, id) {}
    void accept(Visitor& v) const override { v.visit(*this); }
    void addSubordinate(const VEmployee& s) override { sub_.push_back(s); }
    vector<VEmployee> sub_;
};
```

Zadanie 2 - obiektowe wzorce projektowe (2pkt)

Pensja pracowników firmy obliczana jest następująco: dla stażysty (**Trainee**) to mnożnik pensji bazowej, obecnie równy 0.8, dla pracownika (**Employee**) to mnożnik pensji bazowej, obecnie równy 1.1, zaś dla kierownika (**Manager**) to pensja bazowa plus 0.3 sumy pensji jego podwładnych. Dostarcz dla hierarchii **Staff** funkcje obliczania pensji. Pensja bazowa jest przechowywana w obiekcie **HRSystem**, każdy obiekt **Staff** ma uchwyt do **HRSystem**. Używamy kodu z Zad 1, kodu **HRSystem** nie można modyfikować. Jakiego wzorca użyłaś/użyłeś?

```
static const double TraineeFactor = 0.8;
static const double EmployeeFactor = 1.1;
static const double ManagerFactor = 1.0;
static const double ManagerSubFactor = 0.3;

class Employee : public Staff { // pracownik etatowy
    //zakladamy, ze metody i skladowe tej klasy, z Zad 1, sa tutaj dostepne

};

class Trainee : public Staff { // stazysta
    //zakladamy, ze metody i skladowe tej klasy, z Zad 1, sa tutaj dostepne

};

class Manager : public Staff { //kierownik
    //zakladamy, ze metody i skladowe tej klasy, z Zad 1, sa tutaj dostepne

};

Money calcSalary(VStaff staff) {

}
```

Uwagi do prowadzącego (R. Nowaka):

Zadanie 5 - Wymagania (2pkt)

Proszę zaproponować kilka wymagań dla procesu tworzenia, przeprowadzenia i sprawdzenia kolokwium. Zbiór nie musi być kompletny, ale musi być spójny.

Uwagi do prowadzącego (K. Grochowskiego):