

Міністерство освіти і науки України
Хмельницький національний університет
Кафедра комп'ютерної інженерії та інформаційних систем

ЛАБОРАТОРНА РОБОТА № 7
з дисципліни: Об'єктно-орієнтовані технології програмування

Виконав студент: Роюк Р.В.

Група: КІ2м-24-2

Перевірив: Лисенко С.М.

Завдання: написати програмне забезпечення, що описує застосування патерну «Зберігач».

Отже у логістичній компанії є система, яка керує замовленнями на доставку. Іноколи потрібно зберігати поточний стан доставки або повернутися до попереднього стану, якщо, наприклад, доставку скасували або змінили.

```
#include <iostream>
#include <string>
#include <stack>

//
// ==== Клас, що представляє стан доставки ====
//
class Delivery {
private:
    std::string cargo;
    std::string destination;
    std::string transport;
public:
    Delivery(std::string c, std::string d, std::string t)
        : cargo(c), destination(d), transport(t) {}

    void show() const {
        std::cout << "Вантаж: " << cargo
                    << ", Місце призначення: " << destination
                    << ", Транспорт: " << transport << std::endl;
    }

    // Зберігач стану
    class Memento {
        friend class Delivery;
    private:
        std::string cargo, destination, transport;
        Memento(std::string c, std::string d, std::string t)
            : cargo(c), destination(d), transport(t) {}
    };

    Memento save() const {
        return Memento(cargo, destination, transport);
    }

    void restore(const Memento& memento) {
        cargo = memento.cargo;
        destination = memento.destination;
        transport = memento.transport;
    }
}
```

```

    void update(std::string newCargo, std::string newDest, std::string newTrans) {
        cargo = newCargo;
        destination = newDest;
        transport = newTrans;
    }
};

//
// ==== Менеджер збережень (CareTaker) ====
//
class DeliveryHistory {
private:
    std::stack<Delivery::Memento> history;
public:
    void save(const Delivery& d) {
        history.push(d.save());
    }

    void undo(Delivery& d) {
        if (!history.empty()) {
            d.restore(history.top());
            history.pop();
        }
    }
};

//
// ==== Демонстрація ====
//
int main() {
    Delivery delivery("Ноутбук", "Київ", "Авто");
    DeliveryHistory history;

    std::cout << "[Початковий стан]" << std::endl;
    delivery.show();

    // Зберігаємо стан
    history.save(delivery);

    // Змінюємо дані
    delivery.update("Телевізор", "Львів", "Поїзд");
    std::cout << "[Оновлений стан]" << std::endl;
    delivery.show();

    // Відкат до попереднього
    history.undo(delivery);
    std::cout << "[Відновлений попередній стан]" << std::endl;
    delivery.show();

    return 0;
}

```

}

Патерн «Зберігач» дозволяє зберігати й відновлювати попередній стан об'єкта без порушення його інкапсуляції.

Це дуже зручно в ситуаціях, коли потрібно мати можливість "відкотити" стан системи — наприклад, до попереднього стану доставки. У коді це реалізовано як:

- Delivery – основний об'єкт, чий стан зберігається;
- Delivery: :Memento – об'єкт, що зберігає стан об'єкта без розкриття деталей реалізації;
- DeliveryHistory – (Caretaker) керує збереженням/відновленням станів об'єкта.

Перевагами патрену «Зберігач» є те, що він дає змогу створювати історію змін об'єкта, не порушує інкапсуляцію, зручний для «відкату» операції.