

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Звіт

з лабораторної роботи №10

з дисципліни «Ефективність та якість архітектурних рішень  
інформаційних систем»

Виконав:

студент групи ІКМ-М225В

Суліма Нікіта Володимирович

Перевірив:

аспірант каф. КМПС Хорошун Андрій Сергійович

Харків 2025

## Mediator — інтерфейс посередника

```
class Mediator {  
public:  
    virtual void notify(string sender, string event) = 0;  
};
```

Це базовий інтерфейс, який описує метод `notify(sender, event)`.  
Будь-який “посередник” повинен вміти приймати повідомлення від компонентів.

## Component — базовий клас усіх елементів форми

```
class Component {  
protected:  
    Mediator* mediator;  
public:  
    void setMediator(Mediator* m) { mediator = m; }  
};
```

Кожен елемент знає, хто його посередник.  
Якщо щось змінюється в елементі — він повідомляє посередника.

## Конкретні елементи інтерфейсу (форми)

DeliveryDatePicker Вибір дати доставки. При зміні дати повідомляє посередника.

```
void selectDate(string newDate) {  
    date = newDate;  
    mediator->notify("DeliveryDate", "DateChanged");  
}
```

## TimeSlotSelector

Список доступних часових проміжків.  
Посередник оновлює цей список, коли змінюється дата доставки.

```
void updateAvailableSlots(vector<string> slots) {  
    availableSlots = slots;  
}
```

## OtherPersonCheckbox

Чекбокс “Отримувач інша особа”.

```
void toggle(bool state) {
```

```
checked = state;
mediator->notify("OtherPersonCheckbox", "Toggled");
}
```

Коли цей прапорець активовано, посередник показує поля для імені й телефону.

### RecipientNameField і RecipientPhoneField

Текстові поля, які можуть бути показані або сховані:

```
void setVisible(bool visible) {
    cout << "[Поле Ім'я] " << (visible ? "Показано" : "Сховано") << endl;
}
```

### PickupCheckbox

Чекбокс “Самовивіз”.

Якщо користувач його вмикає — доставка відключається, всі поля ховаються.

```
void toggle(bool state) {
    checked = state;
    mediator->notify("PickupCheckbox", "Toggled");
}
```

OrderFormMediator — головний “мозок” форми

Цей клас координує всі елементи.

```
class OrderFormMediator : public Mediator {
private:
    DeliveryDatePicker* datePicker;
    TimeSlotSelector* timeSelector;
    OtherPersonCheckbox* otherCheckbox;
    RecipientNameField* nameField;
    RecipientPhoneField* phoneField;
    PickupCheckbox* pickupCheckbox;
```

У конструкторі він реєструє всі компоненти і встановлює себе як їхній “медіатор”.

Метод notify(sender, event) обробляє всі події:

Зміна дати доставки

```
if (sender == "DeliveryDate" && event == "DateChanged") {  
    if (datePicker->getDate() == "Сьогодні")  
        timeSelector->updateAvailableSlots({"12:00", "14:00", "16:00"});  
    else  
        timeSelector->updateAvailableSlots({"10:00", "12:00", "15:00", "18:00"});  
}
```

### Увімкнення “Отримувач інша особа”

```
else if (sender == "OtherPersonCheckbox" && event == "Toggled") {  
    bool state = otherCheckbox->isChecked();  
    nameField->setVisible(state);  
    phoneField->setVisible(state);  
}
```

### Увімкнення “Самовивіз”

```
else if (sender == "PickupCheckbox" && event == "Toggled") {  
    bool state = pickupCheckbox->isChecked();  
    if (state) {  
        timeSelector->updateAvailableSlots({});  
        nameField->setVisible(false);  
        phoneField->setVisible(false);  
    }  
}
```

## Висновок

У ході виконання лабораторної роботи було реалізовано патерн проєктування «Посередник (Mediator)» на прикладі сторінки оформлення замовлення в службі доставки квітів. Під час реалізації було створено низку компонентів (вибір дати доставки, часовий проміжок, чекбокси, поля вводу тощо), які не взаємодіють безпосередньо між собою, а обмінюються подіями через об'єкт-посередник.