Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №8

з дисципліни <<Технології розробки програмного забезпечення>>

Тема <<ШАБЛОНИ «COMPOSITE», «FLYWEIGHT», «INTERPRETER», «VISITOR»

Виконав:
студент ІА-23
Содолиснький Вадим

Перевірив:
Мягкий М. Ю.

Київ 2024

Тема: ШАБЛОНИ «COMPOSITE», «FLYWEIGHT», «INTERPRETER», «VISITOR»

Хід роботи

Варіант №13

Office communicator

Мережевий комунікатор для офісу повинен нагадувати функціонал програми Skype з можливостями голосового / відео / конференц-зв'язку, відправки текстових повідомлень і файлів (можливо, оффлайн), веденням організованого списку груп / контактів.

Короткий опис патернів

Патерн Composite

Шаблон Composite дозволяє організувати об'єкти у вигляді ієрархічної деревоподібної структури, де як окремі об'єкти, так і їхні групи можуть оброблятися однаково. Composite забезпечує уніфікований інтерфейс для роботи з об'єктами, які можуть містити підоб'єкти. Цей шаблон часто використовується для роботи з графічними компонентами, файловими системами чи організацією складних структур даних (наприклад, документ із заголовками, параграфами та таблицями). Composite спрощує роботу з ієрархіями, дозволяючи виконувати операції, не знаючи, чи це одиничний об'єкт, чи ціла група.

Патерн Flyweight

Flyweight зменшує використання пам'яті, оптимізуючи зберігання об'єктів із великою кількістю повторюваних даних. Замість створення окремих об'єктів для кожного екземпляра, загальні дані винесені в спільний об'єкт, а унікальні — зберігаються зовні. Flyweight оптимізує ресурси, зменшуючи навантаження на пам'ять.

Патерн Interpreter

Шаблон Interpreter визначає спосіб обробки граматики чи мови, дозволяючи інтерпретувати та обчислювати вирази в конкретній предметній області. Кожен тип виразу представлений у вигляді окремого класу, що інкапсулює його логіку. Interpreter використовується в реалізації математичних калькуляторів, SQL-запитів або парсерів мов програмування. Він добре працює в системах, де необхідно динамічно обробляти складні вирази чи команди.

Патерн Visitor

Шаблон Visitor дозволяє додавати нову поведінку для об'єктів без зміни їхніх класів. Відвідувач відокремлює операції від структури даних, дозволяючи легко додавати нові операції, не порушуючи наявну систему. Visitor часто використовується в системах, де є складні об'єкти чи структури (наприклад, дерева), і для кожного типу елемента потрібна різна поведінка. Це робить його корисним для реалізації генераторів звітів, обхідників дерев або компіляторів.

Вибір патерну Flyweight замість Composite

Вибір патерну Flyweight замість Composite обґрунтований вимогами до ефективності використання пам'яті та обробки великої кількості повторюваних елементів, таких шаблони повідомлень.

Office Communicator — це система, яка обробляє велику кількість контактів, кожен із яких має свої атрибути (ім'я, стан тощо). При цьому багато з цих атрибутів можуть бути однаковими для багатьох користувачів. У такій ситуації ключовим завданням є мінімізувати дублювання даних і скоротити споживання пам'яті.

Патерн Flyweight дозволяє розділяти загальний стан між об'єктами, щоб зменшити кількість дублікатів. Це значно знижує використання пам'яті, особливо у випадках, коли список контактів дуже великий.

На противагу цьому, Composite більше підходить для організації об’єктів у деревоподібні структури, наприклад, для моделювання

ієрархії груп контактів. Composite дозволяє працювати з групами контактів і окремими контактами однаково, що корисно для управління ієрархічними структурами.

Однак у Office Communicator потреба в оптимізації пам'яті та продуктивності часто переважає над потребою в організації складних ієрархій. Контакти, хоча й можуть бути організовані в групи, зазвичай відображаються у вигляді плоского списку, де важливіше зберегти ресурси, ніж забезпечити складні операції над групами.

Таким чином, Flyweight обирається, оскільки він дозволяє ефективно управляти великими обсягами повторюваних елементів, що знижує навантаження на пам'ять і підвищує загальну продуктивність системи. Composite у цьому випадку не є настільки доцільним, оскільки головний акцент у системі робиться на оптимізацію, а не на ієрархічну організацію даних.

Використання патерну Flyweight

Переваги

Цей шаблон корисний у випадках, коли програма працює з великою кількістю схожих об'єктів, які мають загальні властивості, що можуть бути перенесені в спільний об'єкт.

Застосування

В застосунку, який працює на стороні клієнта зберігається такі елементи як група або контакт, які можуть змінювати свій стан динамічно. Для економії ресурсів програми можна використовувати один і той самий об'єкт просто посилаючись на нього з різних частин додатку, що значно покращує продуктивність.

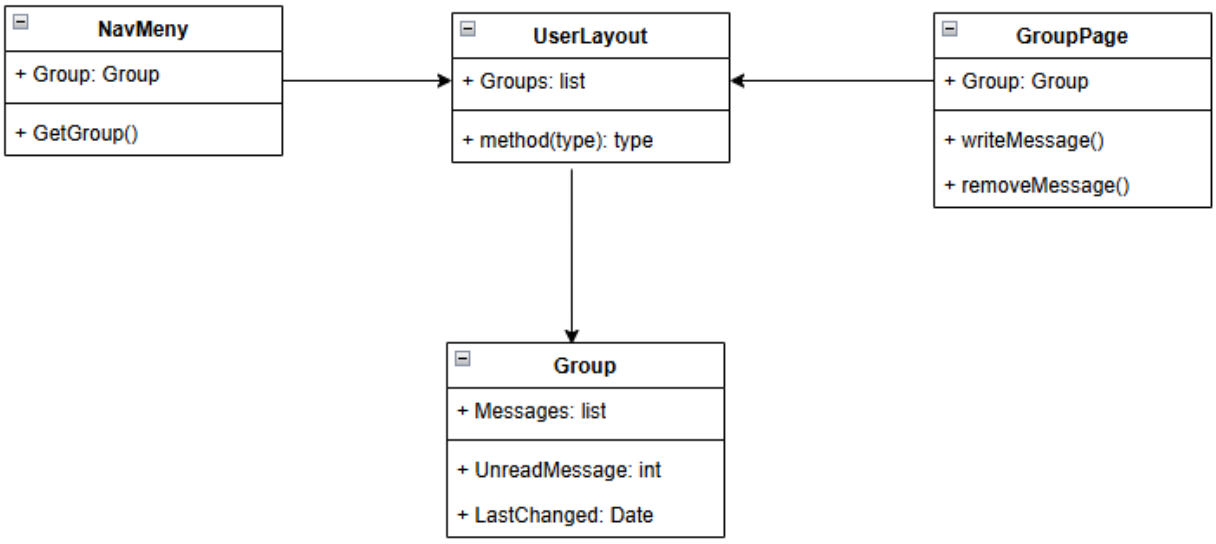

Рис. 1 – Реалізація патерну Flyweight

Діаграма показує реалізацію патерну Flyweight, у якій клас UserLayout є точкою доступу до об’єктів Group. Це означає, що об’єкти Group не є глобальними або розділеними самі по собі, а натомість отримуються через UserLayout, який керує їхнім збереженням і використанням.

Клас UserLayout зберігає список груп (Groups: list), що належать користувачу, і надає методи для роботи з цими групами. Коли інші компоненти, такі як NavMeny або GroupPage, потребують доступу до конкретної групи, вони звертаються до UserLayout.

Наприклад, NavMeny отримує список Groups через UserLayout, щоб забезпечити навігацію до відповідної групи. Аналогічно, GroupPage використовує групу, отриману з UserLayout, для взаємодії з повідомленнями в межах цієї групи.

```
@foreach (var group in FilteredGroups)
{
    <div class="list-item" @onclick="() => OnGroupSelected.InvokeAsync(group)">
        <img src="images/group.png" />
        <div>@group.Name</div>
        <div>@group.UnviewedMessages</div>
        @if (group.DateTime != null)
        {
            <div>@group.DateTime.Value.ToString("HH:mm")</div>
        }
    </div>
}
```

Рис. 2 – Відображення груп в NavMenu

Код використовує перебір списку груп, що містяться у колекції FilteredGroups, і відображає кожну групу у вигляді HTML-елементів. Колекція FilteredGroups містить відфільтровані дані.

При натисканні на елемент списку викликається метод OnGroupSelected.InvokeAsync(), який відповідає за сповіщення інших компонентів про вибір користувачем конкретної групи. Цей підхід дозволяє забезпечити взаємодію між компонентами відкриття відповідного чату.

```
private async Task HandleGroupSelected(Group group)
{
    var result = await ChatApiService.GetGroupAsync(group.Id, token);
    if (!result.IsSuccess)
    {
        errorMessage = result.ErrorMessage ?? "Error";
        isError = true;
        return;
    }
    selectedGroupId = group.Id;
    selectedContactId = 0;

    selectedGroup = result.Data;
    group.UnviewedMessages = 0;

    if (selectedGroup == null)
    {
        await ExitUserPage();
        return;
    }
    isContactOpened = false;
    isGroupOpened = true;
    isChatOpened = true;
    selectedContact = null;
    await InvokeAsync(StateHasChanged);
}
```

Рис. 3 – Метод HandleGroupSelected який відповідає за зміну стану об’єкта selectedGroup

Метод HandleGroupSelected реалізує логіку обробки вибору групи користувачем у чаті. Коли користувач вибирає певну групу, метод завантажує інформацію про цю групу, оновлює стан програми та відповідно змінює інтерфейс.

Спочатку метод викликає ChatApiService.GetGroupAsync, щоб отримати дані про вибрану групу, використовуючи її ідентифікатор і токен авторизації. Якщо запит виконується невдало, метод записує повідомлення про помилку або встановлює значення за замовчуванням "Error", вказує, що сталася помилка, і завершує роботу.

Якщо дані групи успішно завантажені, метод оновлює стан програми. Отримані дані зберігаються в selectedGroup.

```
@code {

    [Parameter]
    public Group Group { get; set; }

    [Parameter]
    public User user { get; set; }

    [Parameter]
    public List<Contact> Contacts { get; set; }
```

Рис. 3 – Параметри які отримує GroupPage від UserLayout

Обрана група відкривається на сторінці GroupPage, яка відображає вміст групи та взаємодії з нею. Наприклад, додати нове повідомлення або додати нового користувача.

```
<GroupChat Group="selectedGroup" Contacts="user.Contacts" user="user"/>
```

Рис. 4 – Компонент GroupChat, в який передається selectedGroup, user.Contacts та user

Висновок: Патерни «Composite», «Flyweight», «Interpreter» і «Visitor» вирішують різні задачі у проектуванні програм, забезпечуючи ефективність, гнучкість і масштабованість. Вони оптимізують роботу зі структурами, ресурсами та поведінкою об'єктів, роблячи систему більш організованою та зручною для розширення. З цією метою я використав Flyweight для економії обчислювальних ресурсів шляхом

створення посиланнь на об’єкт, який часто використовується різними компонентами.