

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

Содолиський Вадим

Перевірив:

Мягкий М. Ю.

Київ 2024

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

Хід роботи

Патерн Flyweight

Flyweight (або "Приспособленець") — це структурний шаблон проектування, який дозволяє зменшити витрати пам'яті шляхом спільного використання однакових об'єктів замість створення їхніх дублікатів.

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.

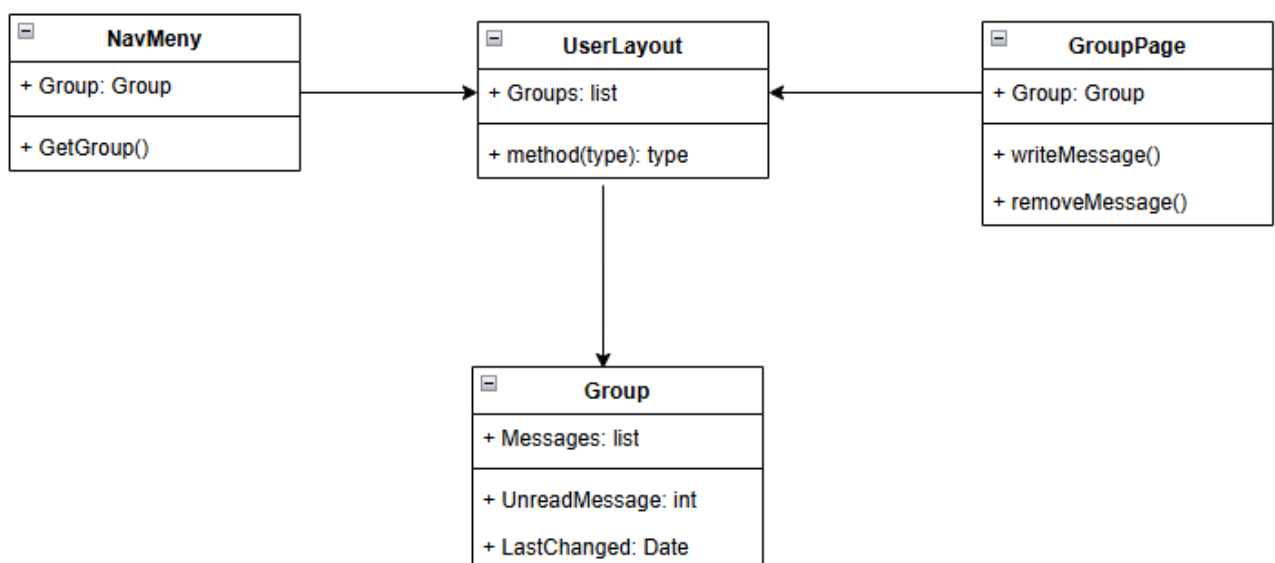


Рис. 1 – Реалізація патерну Flyweight

```

@foreach (var group in FilteredGroups)
{
    <div class="list-item" @onclick="() => OnGroupSelected.InvokeAsync(group)">
        
        <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

```

private async Task SendMessage()
{
    if (string.IsNullOrEmpty(content) && SelectedFiles.Count() == 0) return;

    List<Document> documents = new();
    foreach (var file in SelectedFiles)
    {
        documents.Add(new Document()
        {
            Name = file.Name,
            UniqueIdentifier = string.Empty
        });
    }

    string uniqueIdentifier = Guid.NewGuid().ToString();
    Group.Chat.Messages.Add(new Message()
    {
        Content = content,
        UserId = user.Id,
        User = user,
        ChatId = Group.ChatId,
        Date = DateTime.Now,
        UniqueIdentifier = uniqueIdentifier,
        Documents = documents
    });

    string copyContent = content;
    content = string.Empty;
    await OnSendMessage.InvokeAsync(new MessageStorageModel(user.Id, uniqueIdentifier, Group.ChatId, copyContent, nameof(Group), SelectedFiles));
    SelectedFiles.Clear();
}

```

Рис. 3 – Метод надсилання повідомлення групі

```

private async Task DeleteMessage(int messageId)
{
    Message? message = Group.Chat.Messages.FirstOrDefault(m => m.Id == messageId);
    if (message == null) return;
    Group.Chat.Messages.Remove(message);
    await OnDeleteMessage.InvokeAsync(new DocumentMessageChatId(0, messageId, Group.ChatId, message.UniqueIdentifier));
}

```

Рис. 4 – Метод видалення повідомлення у групі

Висновок: я використав патерн Flyweight для економії ресурсів застосунка шляхом створення посилань на об'єкт замість створення його копій.