Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт по лабораторній роботі № 4

“Створення проекту «Blog». Cтворення компонентів.”

з дисципліни: «Реактивне програмування»

Студент: Зусько Владислав Юрійович

Група:  ІП-02

Дата захисту роботи:

Викладач: доц. Полупан Юлія Вікторівна

Захищено з оцінкою:

Київ, 2023

## Зміст

[Опису компоненту Post 3](#_Toc148135837)

[Опис компоненту PostForm 5](#_Toc148135838)

[Література 7](#_Toc148135839)

## Опису компоненту Post

Цей фрагмент коду представляє Angular компонент з назвою PostComponent. Давайте розглянемо його функціонал, мету та методи.

PostComponent є Angular компонентом, який відповідає за відображення та обробку інформації про пости. Він приймає вхідні дані (post) через атрибут @Input('toPost') public post!: Post;. Ці дані використовуються для відображення інформації про пост в шаблоні компонента.

PostComponent надає можливість видалення поста за допомогою методу removePost(), який активується при події onRemove. Головною метою PostComponent є відображення інформації про пост (ім'я, текст тощо) та надання можливості видалення цього поста. Компонент містить наступні методи:

* ngOnInit(): Це метод, який імплементує інтерфейс OnInit і викликається під час ініціалізації компонента. У даному випадку, метод порожній і не виконує жодних конкретних дій.
* ngOnDestroy(): Це метод, який імплементує інтерфейс OnDestroy і викликається перед тим, як компонент буде знищений. У цьому методі виводиться повідомлення у консоль ("метод ngOnDestroy"), але в інших випадках цей метод може використовуватися для відписки від підписок або виконання інших завершальних дій.
* removePost(): Цей метод викликається, коли користувач намагається видалити пост. Він емітує подію onRemove, передаючи ідентифікатор поста (this.post.id) як дані події. Це дозволяє батьківському компоненту або іншим частинам програми реагувати на видалення поста.

Розглянемо фрагмент коду представляє шаблон компонента Angular, який використовується для відображення даних поста.

* <div class="card">: Це контейнер <div>, який має клас "card" і представляє структуру для відображення поста.
* <h2>{{ post.title }}</h2>: Цей елемент <h2> відображає заголовок поста. Заголовок отримується з властивості post.title і вставляється в шаблон за допомогою подвійних фігурних дужок {{}}. Таким чином, заголовок поста буде відображатися в цьому <h2> елементі.
* <p>{{ post.text }}</p>: Цей елемент <p> відображає текст поста. Текст отримується з властивості post.text і також вставляється в шаблон за допомогою подвійних фігурних дужок {{}}. Таким чином, текст поста буде відображатися в цьому <p> елементі.
* <button class="btn" (click)="removePost()">&times;</button>: Це кнопка <button>, яка має клас "btn" і містить символ "×" (амперсанд-знак) для видалення поста. Важливою частиною цього елемента є атрибут (click)="removePost()", який вказує, що при кліку на кнопку має викликатися метод removePost(). Цей метод використовується для видалення поста і, ймовірно, був реалізований у відповідному компоненті.
* <ng-content></ng-content>: Цей тег <ng-content> використовується для вставки контенту, який може бути переданий в цей компонент ззовні через <app-post> (або іншу) обгортку компонента. Всі додаткові елементи або текст, які вставляються в <app-post>, будуть відображені тут.

Загальною метою цього шаблону є відображення інформації про пост, включаючи його заголовок та текст, а також надання можливості видалення поста за допомогою кнопки "×".

## Опис компоненту PostForm

Цей фрагмент коду представляє Angular компонент з назвою PostFormComponent. Давайте розглянемо його функціонал, мету та методи. PostFormComponent є Angular компонентом, який відповідає за форму для створення нових постів. Він має два внутрішніх поля: title та text, які використовуються для введення інформації про новий пост. PostFormComponent надає можливість створення нового поста та надсилання його даних через вихідну подію onAdd.

Головною метою PostFormComponent є збір даних для створення нового поста та надсилання цих даних батьківському компоненту або іншим частинам програми за допомогою події onAdd.

Методи:

* ngOnInit(): Це метод, який імплементує інтерфейс OnInit і викликається під час ініціалізації компонента. У даному випадку, метод порожній і не виконує жодних конкретних дій.
* addPost(): Цей метод викликається, коли користувач намагається додати новий пост. Він перевіряє, чи обидва поля title і text не містять лише пропуски (порожні рядки), і якщо так, то створює об'єкт Post з введеними даними. Після цього він емітує подію onAdd, передаючи створений пост як дані події. Крім того, метод виводить інформацію про новий пост у консоль та очищає поля title та text, готуючи форму для створення наступного поста.

Розглянемо шаблон компоненту Angular, який складається з HTML-коду та використовує директиви Angular для забезпечення функціональності.

* <div>: Цей елемент <div> визначає контейнер для інших елементів у шаблоні.
* <input> (x2): Два елементи <input> використовуються для введення даних. Кожен із них має наступні атрибути:
  + type="text": Вказує, що це текстове поле для введення текстових даних.
  + class="form-control": Застосовує CSS-клас "form-control" для стилізації поля вводу.
  + placeholder: Встановлює текстовий заповнювач, який з'являється у полі вводу, щоб надати вказівки користувачу, що потрібно вводити.
  + [(ngModel)]="title" та [(ngModel)]="text": Використовуються для двостороннього зв'язування даних. Значення введених користувачем даних зберігаються в змінних title і text моделі компонента, і зміни в цих змінних впливають на відображення у відповідних полях вводу.
* <button>: Елемент <button> використовується для створення кнопки "Додати пост". Він має наступний атрибут:
  + (click)="addPost()": Ця директива Angular встановлює обробник події, який викликає метод addPost() компонента, коли користувач клікає на кнопку.

Всі ці елементи розміщені всередині <div>, утворюючи контейнер для введення даних та кнопки для додавання нового поста. Користувач може ввести заголовок поста в перше поле вводу, текст поста - в друге поле вводу, і при кліку на кнопку "Додати пост" викликається метод addPost(), який створює новий пост і відправляє його дані на подальше оброблення.

## Література

1. <https://angular.io/>
2. <https://github.com/angular/angular>
3. https://developer.mozilla.org/en-US/docs/Learn/Tools\_and\_testing/Client-side\_JavaScript\_frameworks/Angular\_getting\_started