

**МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА
ІНТЕРНЕТ-КОНФЕРЕНЦІЯ**

**«ТЕНДЕНЦІЇ ТА ПЕРСПЕКТИВИ РОЗВИТКУ
НАУКИ І ОСВІТИ
В УМОВАХ ГЛОБАЛІЗАЦІЇ»**



ВИПУСК 101

31 січня 2024 р.

м. Переяслав

ПОРІВНЯЛЬНИЙ АНАЛІЗ JAVASCRIPT-ФРЕЙМВОРКІВ ТА БІБЛІОТЕК ДЛЯ FRONT-END

У статті наведено результати дослідження і порівняльний аналіз фреймворків та бібліотек мови JavaScript для Front-End розроблення сайтів. Для порівняння взято один з найпопулярніших фреймворків Angular і одну з найпопулярніших бібліотек React.

Ключові слова: мова програмування, JavaScript, Angular, React, фреймворк, бібліотека.

The article provides research results and a comparative analysis of JavaScript frameworks and libraries for Front-End website development. One of the most popular Angular frameworks and one of the most popular React libraries are taken for comparison.

Key words: programming language, JavaScript, Angular, React, framework, library.

Вступ. JavaScript є одною з найпопулярніших мов програмування для розроблення веб-додатків. Вона дає розробникам широкий спектр можливостей для створення інтерфейсів користувача (UI) та взаємодії з ними.

Для створення Front-End-додатків існує безліч різних JavaScript-фреймворків та бібліотек. Кожен з них має свої переваги та недоліки, тому важливо вибрати правильне рішення для конкретного проекту.

У цій статті викладено порівняльний аналіз деяких найпопулярніших JavaScript-фреймворків та бібліотек, а саме фреймворку Angular і бібліотеки React. Буде розглянуто їхні основні характеристики, переваги та недоліки, а також сфери застосування.

Основна частина. Фреймворки та бібліотеки JavaScript – це інструменти, які допомагають розробникам створювати веб-додатки. Вони забезпечують базову структуру та функціональність, що дозволяє розробникам зосередитися на створенні унікального дизайну та досвіду користувача.

Фреймворки – це комплексні рішення, які пропонують широкий спектр функцій та можливостей.

Angular – написаний на TypeScript front-end фреймворк з відкритим кодом, який розробляється під керівництвом Angular Team у компанії Google, а також спільнотою приватних розробників та корпорацій [1].

Angular – це повний фреймворк, що включає в себе всі необхідні функції для створення веб-додатків, такі як система маршрутизації, система компонентів, система шаблонів, система стилів і система управління станом.

Angular використовує компоненти для створення інтерфейсу користувача. Компоненти – це самостійні модулі, які містять код, шаблони та стилі. Це дозволяє створювати чіткі та повторювані інтерфейси користувача.

Фреймворк Angular використовує шаблони для опису вигляду інтерфейсу користувача. Шаблони – це HTML-файли, які містять спеціальні синтаксичні конструкції, які дозволяють зв'язувати дані з інтерфейсом користувача.

Angular використовує систему стилів, яка заснована на CSS. Це дозволяє використовувати стандартні CSS-правила для стилізації інтерфейсу користувача. Angular використовує систему управління станом, яка дозволяє зберігати дані, які змінюються в часі. Це дозволяє створювати інтерфейси користувача, які реагують на зміни даних.

Фреймворк Angular використовує двостороннє зв'язування даних, яке дозволяє зв'язувати дані з інтерфейсом користувача в реальному часі. Також Angular побудований на мові TypeScript, яка, на відміну від JavaScript, пропонує додаткові функції безпеки та продуктивності.

Основні переваги Angular:

1. Сильний фокус на продуктивності та масштабованості.
2. Широкий спектр функцій і можливостей, що робить його придатним для широкого спектру проектів.
3. За рахунок того, що Angular є фреймворком, він дає можливість розробникам створювати складні і багаторівневі інтерфейси користувача.

Основні недоліки Angular:

1. Може бути складним у вивченні та використанні, особливо для початківців.
2. Потрібен значний досвід програмування JavaScript та TypeScript.
3. Не такий гнучкий, як React, що може обмежити можливості при розробці нестандартних інтерфейсів користувача.

Angular найчастіше використовується для розроблення односторінкових веб-додатків (SPA). Основними сферами застосування є розроблення інтернет-магазинів, соціальних мереж, бізнес-додатків, веб-ігор тощо.

Бібліотеки – це спеціалізовані рішення, які фокусуються на певній області функціональності. Вони зазвичай пропонують більш вузький спектр функцій, ніж фреймворки.

React – відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту вебсторінки, з якими стикаються в розробці односторінкових застосунків. Розробляється Meta і спільнотою індивідуальних розробників [2].

React – це бібліотека, яка фокусується на створенні компонентів інтерфейсу користувача. React використовує однонаправлене зв'язування даних, яке забезпечує більшу стабільність і продуктивності.

React також використовує компоненти для створення інтерфейсу користувача. Компоненти в React більш гнучкі, ніж компоненти в Angular, що дозволяє створювати більш нестандартні інтерфейси користувача. React використовує простір імен для організації коду. Це полегшує управління кодом і покращує його читабельність. React має велику спільноту розробників, яка створила безліч бібліотек і інструментів, які можуть бути корисними для розробки веб-додатків.

Основні переваги React:

1. Простий у вивченні та використанні.
2. Гнучкий і адаптивний, що дозволяє створювати нестандартні інтерфейси користувача.
3. Добре масштабується і продуктивний.

Основні недоліки React:

1. Не пропонує такої ж кількості функцій і можливостей, як Angular.
2. Може бути менш ефективним для створення складних і багаторівневих інтерфейсів користувача.

React також часто використовується для розроблення односторінкових веб-додатків (SPA). Основними сферами застосування є розроблення інтернет-магазинів, стартап-додатків, бізнес-додатків, веб-ігор тощо.

Висновки. Angular і React – це два потужних інструменти, які можна використовувати для створення веб-додатків. Вони мають свої власні унікальні характеристики, переваги та недоліки, які роблять їх придатними для різних проектів. Вибір між Angular і React залежить від конкретних потреб і вимог розробників та замовників ПЗ. Angular – фреймворк з широким спектром функцій і можливостей, який підходить для складних і багаторівневих інтерфейсів користувача. React – відносно проста у вивченні і використанні бібліотека, яка дозволяє швидко створювати нестандартні інтерфейси користувача і є більш гнучкою, проте менше підходить для розроблення великих бізнес-проектів.

ДЖЕРЕЛА ТА ЛІТЕРАТУРА

- 1 Angular (фреймворк). Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Angular_\(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA\)](https://uk.wikipedia.org/wiki/Angular_(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA)) (дата звернення: 04.01.2024)
- 2 React. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/React> (дата звернення: 04.01.2024).

Науковий керівник: кандидат технічних наук, доцент Антоненко С. В.

УДК 004.02

*Євгеній Мاستалярчук
(Рівне, Україна)*

РЕАЛІЗАЦІЯ КОНТЕКСТНОЇ ДОПОМОГИ ВИКОРИСТОВУЮЧИ ЛАНЦЮЖОК ОБОВ'ЯЗКІВ

При використанні даного підходу, можна сказати, що кожен об'єкт у ланцюзі обов'язків відповідає за конкретну задачу і може вирішити, чи передати обробку наступному об'єкту в ланцюзі. Цей підхід полегшує розширення та модифікацію системи, оскільки можна додавати нові об'єкти без зміни вже існуючого коду. Використання "Ланцюжка обов'язків" рекомендується там, де необхідно уникати прив'язки відправника запиту до його отримувача.

Ключові слова: ланцюжок обов'язків, програмування C#, стандартизація коду, універсальні принципи.

Using this approach, we can say that each object in the chain of responsibilities is responsible for a specific task and can decide whether to pass processing to the next object in the chain. This approach makes it easy to extend and modify the system, as new objects can be added without changing the existing code. The use of the Chain of Responsibilities is recommended where it is necessary to avoid binding the sender of the request to its recipient.

Key words: chain of responsibility, C# programming, code standardization, universal principles.

Суть патерну

Ланцюжок обов'язків – це патерн проектування з поведінкової категорії, що забезпечує передачу запитів послідовно через ланцюг обробників. Кожен наступний обробник вирішує, чи може він самостійно обробити запит, чи варто передати його далі ланцюгом.

У практиці, як і в інших поведінкових патернах, ланцюжок обов'язків ґрунтується на ідеї перетворення окремих обов'язків у відокремлені об'єкти. Кожна перевірка тепер відповідає окремому класу з єдиним методом виконання, приймаючи дані запиту як аргументи.

Ключовий момент полягає в тому, щоб з'єднати всі об'єкти обробників у єдиний ланцюг. Кожен обробник містить посилання на наступний обробник у ланцюгу, що дозволяє послідовно передавати обробку. Передаючи запити до першого обробника ланцюга, ми можемо бути впевнені, що весь ланцюг зможе їх обробити, і його довжина стає незначущою.

Найважливіше – обробник не обов'язково повинен передавати запит далі, що відкриває шлях для різних сценаріїв використання цієї особливості.

Ольга Юніна

ВИКОРИСТАННЯ ІНСТРУМЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ
У ВИВЧЕННІ ІНОЗЕМНИХ МОВ У ЗАКЛАДАХ ВИЩОЇ ОСВІТИ 112

ПСИХОЛОГІЯ / ПСИХОЛОГИЯ

Людмила Гуляєва, Людмила Бебеза, Гліб Пріб

АНАЛІЗ ЗАЦІКАВЛЕНОСТІ СТУДЕНТІВ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ
У ВИВЧЕННІ ЄВРОПЕЙСЬКОЇ ТА УКРАЇНСЬКОЇ ПРАКТИКИ
ПІДТРИМКИ ПСИХІЧНОГО ЗДОРОВ'Я НА РОБОЧОМУ МІСЦІ:
ДОСВІД АКАДЕМІЇ ПРАЦІ, СОЦІАЛЬНИХ ВІДНОСИ І ТУРИЗМУ 116

Олександр Зіненко, Світлана Зіненко

АНАЛІЗ ТА РОЗРОБКА ПРАКТИЧНОГО АЛГОРИТМУ СОЦІАЛЬНО-
ПСИХОЛОГІЧНОЇ АДАПТАЦІЇ ДІТЕЙ З ОСОБЛИВИМИ ОСВІТНИМИ
ПОТРЕБАМИ У ЗАКЛАДАХ ОСВІТИ 120

Ірина Козубенко

СТРЕСИ У ПРОФЕСІЙНІЙ ДІЯЛЬНІСТЬ ПРАЦІВНИКІВ
НАЦІОНАЛЬНОЇ ПОЛІЦІЇ У ПЕРІОД ВОЄННОГО СТАНУ 123

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ / ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Світлана Вишемирська, Микола Сигида

ВИКОРИСТАННЯ КОНСОЛІДОВАНОЇ ІНФОРМАЦІЇ ДЛЯ ПІДВИЩЕННЯ
КОНКУРЕНТОСПРОМОЖНОСТІ ПІДПРИЄМСТВА 127

Savanşir Kazımov, Mahir Mirzayev

NƏZARƏTLƏ OLAN MAŞIN TƏLİMİ ALQORITMLƏRİNDƏ KLASSIFİKASIYA
MƏSƏLƏSİNƏ K-YAXIN QONŞU ÜSULUNUN TƏTBİQİ 131

Євгенія Лашко

ПОРІВНЯЛЬНИЙ АНАЛІЗ JAVASCRIPT-ФРЕЙМВОРКІВ
ТА БІБЛІОТЕК ДЛЯ FRONT-END 133

Євгеній Масталярчук

РЕАЛІЗАЦІЯ КОНТЕКСТНОЇ ДОПОМОГИ ВИКОРИСТОВУЮЧИ
ЛАНЦЮЖОК ОБОВ'ЯЗКІВ 135

Оксана Нікітська, Євгеній Масталярчук

СИНЕРГІЯ КОМУНІКАЦІЇ ТА ІТ: ЗАБЕЗПЕЧЕННЯ
ГЛОБАЛЬНОГО УСПІХУ В СУЧАСНІЙ ОСВІТІ 140

Ірина Семігайло, Людмила Титаренко

ВИКОРИСТАННЯ ЦИФРОВИХ ІНСТРУМЕНТІВ GOOGLE НА ЗАНЯТТЯХ
З ОСНОВ ПОЧАТКОВОГО КУРСУ МАТЕМАТИКИ В ЗАКЛАДІ ФАХОВОЇ
ПЕРЕДВИЩОЇ ОСВІТИ В УМОВАХ ЗМІШАНОГО НАВЧАННЯ 143

Тетяна Хоренко, Олександр Хоренко

ВИКОРИСТАННЯ ІКТ НА ЗАНЯТТЯХ З ФІЗИКИ ТА АСТРОНОМІЇ
В ЗАКЛАДІ ФАХОВОЇ ПЕРЕДВИЩОЇ ОСВІТИ 146

ФІЗИКО-МАТЕМАТИЧНІ НАУКИ / ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

Мейрамкул Абдрахманова, Баянсұлу Амандық, Замира Нәлібек

ӘР ТҮРЛІ ТИПТЕГІ ЦИФРЛЫҚ РЕСУРСТАРДЫҢ
ФУНКЦИОНАЛДЫҚ ЕРЕКШЕЛІКТЕРІ 149

Мейрамкул Абдрахманова, Аружан Берік, Акерке Капарова

МАТЕМАТИКА САБАҚТАРЫНДА ЦИФРЛЫҚ БІЛІМ БЕРУ
РЕСУРСТАРЫН ПАЙДАЛАНУ 152

Мейрамкул Абдрахманова, Қырмызы Төлендіқызы, Аида Олжабай

САНДЫҚ БІЛІМ БЕРУ РЕСУРСТАРЫ МАТЕМАТИКА САБАҚТАРЫНДА
ОҚУ-ТАНЫМДЫҚ ҚҰЗЫРЕТТІЛІКТІ ДАМУ ТҰРАЛЫ РЕТІНДЕ 154

Гүлмара Амангали

ЭЛЕКТРОНДЫҚ ОҚУЛЫҚТАРДЫҢ АРТЫҚШЫЛЫҚТАРЫ
МЕН КЕМШІЛІКТЕРІ 157