

## МЕТОДИ РОЗШИРЕННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА

Смирнов О.Є., Зеленьова І.Я.

Донецький національний технічний університет, м. Донецьк

Широке розповсюдження Internet-технологій спричинило інтенсивний розвиток різноманітних сайтів, які використовуються підприємствами, організаціями, приватними особами тощо.

У даній роботі розглянуто проектування форми для реєстрації за допомогою шаблону MVVM, тому що він дозволяє виконати прив'язку даних. За рахунок прив'язки властивостей представлення до моделі представлення виконується певне зв'язування цих компонентів, що повністю звільняє розробника від необхідності писати в моделі представлення код, безпосередньо оновлюючи саме представлення. Система прив'язки даних підтримує також перевірку допустимості введення, що забезпечує стандартний шлях передачі помилок перевірки допустимості представлення [1,3].

Існує три типи шаблонів проектування: MVC, MVP та MVVM.

MVC (модель-вид-контроллер) – архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон поділяє систему на три частини: модель даних, вигляд даних та керування. Застосовується для відокремлення даних (модель) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача [4].

Мета шаблону – гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності. Принцип взаємодії компонентів структури MVC зображено на рис. 1.

MVP (модель-вид-презентатор) – шаблон проектування, похідний від MVC, який використовується для побудови інтерфейсу користувача. У MVP презентатор виконує роль посередника (виконує ті ж самі функції, що і контролер у в MVC) [1]. Окрім цього, презентатор відповідає за керування подіями, які відбуваються на стороні інтерфейсу користувача (наприклад mouseDown, keyDown тощо), які зазвичай обробляв контролер.

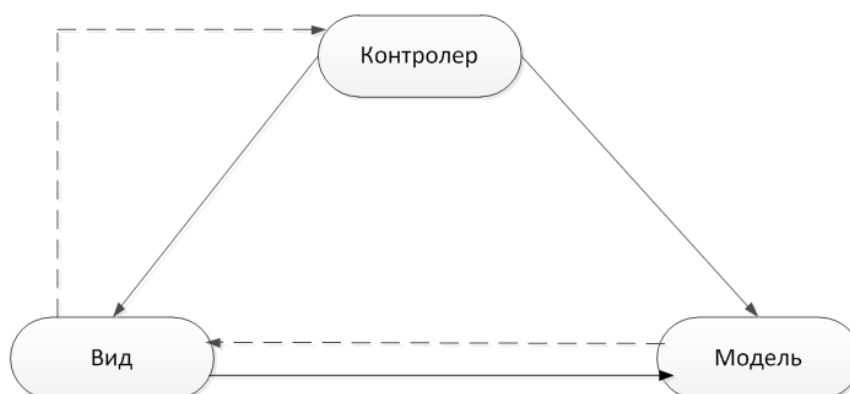


Рисунок 1. Діаграма взаємодії між елементами шаблону MVC

MVVM (модель-вид-модель вида) – це шаблон, що застосовується під час проектування архітектури застосунків (додатків). Публічно вперше був представлений Джоном Госсманом у 2005 році як модифікація шаблону Presentation Model. MVVM орієнтований на такі сучасні платформи розробки, як Windows Presentation Foundation та Silverlight від компанії Microsoft [3].

MVVM використовується для відокремлення моделі та її представлення. Необхідністю цього є надання можливості змінювати їх незалежно одну від одної. Наприклад, розробник працює над логікою роботи з даними, а дизайнер відповідно із користувацьким інтерфейсом.

MVVM зручно використовувати замість класичного MVC та йому подібних у тих випадках, коли на платформі, де ведеться розробка, присутнє «зв'язування даних». Принцип взаємодії компонентів структури MVVM зображено на рис. 2.

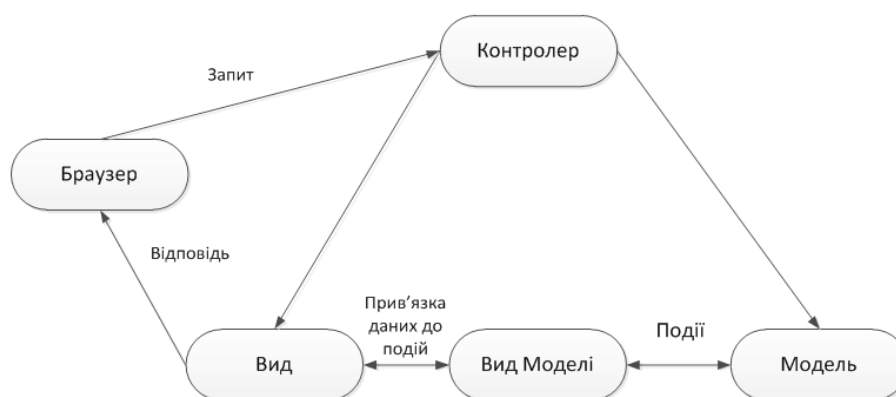


Рисунок 2. Діаграма взаємодії між елементами шаблону MVVM

Саме за таким шаблоном було реалізовано інтерфейс користувача для реєстрації на сервері. Використовувалася бібліотека knockout.js – з її допомогою встановлювалися зв'язки між даними та відображенням на веб-сторінці. Усі дані зберігаються в моделі, відображення лінкується до даних, а зв'язка вид-модель слідує за синхронізацією цих двох параметрів. Також була задіяна графічна оболонка Twitter Bootstrap – це набір інструментів від Twitter (відноситься до класу інструментів: CSS-фреймворк), створений для полегшення розробки веб застосунків та сайтів. Він включає CSS та HTML для типографії, форм, кнопок, таблиць, сіток, навігації, тощо.

Bootstrap використовує найсучасніші напрацювання в області CSS та HTML, тому необхідно бути уважним при підтримці старих браузерів.

На основі проведеного аналізу шаблонів пропонується форма, розроблена за шаблоном MVVM, яка буде виконувати наступні функції:

- реєстрація користувача на сервері;
- перевірка коректності даних, що вводяться. Якщо була допущена помилка при введенні, то дані не можливо відправити.

Для того, щоб створити нову форму, необхідно натиснути кнопку “Create”. Праворуч від полів, які необхідно заповнити наведені підказки, наприклад, що ім'я повинно бути не менш ніж 2 символи. Якщо все було введено коректно, праворуч з'явиться напис “Correct” і, за натисненням кнопки “Submit”, можна буде відправити ці дані у форматі JSON-строки на сервер. Також можна створити ще одну таку ж саму

форму, якщо натиснути “Add a contact” та відправити її.

Отримана форма в подальшому може бути вдосконалена шляхом оптимізації початкового коду, що дозволить збільшити швидкість завантаження. Також можна змінити розташування кнопок, для більш зручного користування.

### **Перелік посилань**

- [1] Троелсен, Ендрю. Язык программирования C# 2010 и платформа .NET 4.0, 5-е вид. : переклад з англ.: – М. : ООО «И.Д. Вильямс», 2011. – 1392 с. : ил. – Парал.тит.англ. ISBN 978-5-8459-162-2
- [2] Knockoutjs 2.0. . [Електронний ресурс]. – Режим доступу: <http://knockoutjs.com/>
- [3] MSDN Magazine, Microsoft. [Електронний ресурс]. – Режим доступу: <http://msdn.microsoft.com>
- [4] ASP.NET. [Електронний ресурс]. – Режим доступу: <http://www.asp.net>