

АВТОМАТИЧНА ГЕНЕРАЦІЯ ПРОГРАМНИХ КОДІВ ЗАСОБАМИ YII-ФРЕЙМВОРКУ

Зарицький Р.Б.¹⁾, Яковів В.І.²⁾

Тернопільський національний економічний університет

¹⁾ магістрант; ²⁾ ст. викладач

I. Вступ

У програмній інженерії давно доведено факт повторюваності дій програміста, який призводить до переписування однотипних процедур і методів та зменшує таким чином швидкість написання програмного коду. Для вирішення цієї задачі було запропоновано концепцію автоматичної генерації коду, в основі якої лежить можливість генерування окремих компонент програми по заданих параметрах [1,2].

Прикладом автоматичного генератора коду є Yii-фреймворк, який дає можливість згенерувати найпростіші форми введення, виведення і зміни інформації в базі даних, створюючи при цьому необхідні контролери, моделі і відображення [3]. При цьому програмісту потрібно через веб-інтерфейс просто вказати назву таблиці, для якої потрібно згенерувати форму. Одним з найпотужніших інструментів Yii-фреймворку є модуль кодогенерації, основною задачею якого є економія часу програміста по створенню основи зв'язку веб-додатку з базою даних, основного каркасу модулю, моделі, розширення чи файлів операції CRUD.

II. Мета роботи

Метою роботи є розробка шаблону типового модуля веб-додатку та системи взаємодії між модулями на основі кодогенератора Yii-фреймворку.

III. Особливості розробки шаблону

Для розробки шаблону веб-модуля був обраний фреймворк Yii [3], написаний на мові програмування PHP відповідно до патерну проектування MVC [4].

Створений у роботі кодогенератором веб-модуль має наступну структуру:

Моделі:

- <модель> - головна модель сутності, що створюється, вміщує основну бізнес-логіку сутності модуля;
- <модель_перекладів> - модель перекладів, зв'язана з головною моделлю та містить в собі усі мультимовні поля сутності;
- <модель_пошуку> - модель, що наслідує головну модель, та містить бізнес-логіку для пошуку даних по сутності, здійснює пошук по головній моделі та перекладам;
- <модель_запиту> - модель запиту головної моделі до бази даних;

Контролер:

- <контролер> - містить логіку зчитування даних, відданих користувачем системі, їх переправлення на модель та відображення кінцевого результату користувачеві;

Види: створення, редагування, форма, частина форми, таблиця.

Поведінки (є спільними для всіх модулів та сутностей):

- поведінка деревовидної структури Nested Set – містить логіку для вибірки елементів по дереву, додавання елементів та переміщення їх по дереву;
- мультимовність – містить логіку для вибірки та створення зв'язку між головною моделлю та її перекладами;
- поведінка зв'язаних моделей – містить логіку створення зв'язків між головною моделлю та моделями, вказаними як зв'язки;
- поведінка прикріплених файлів – містить логіку збереження файлів та створення їх зв'язку до головної моделі.

Трейти:

- <трейт_зв'язку> - трейт, що відноситься до моделі та містить методи зв'язку з іншими моделями.

На рисунках 1-2 наведено UML діаграми, що описують структуру та роботу модуля.

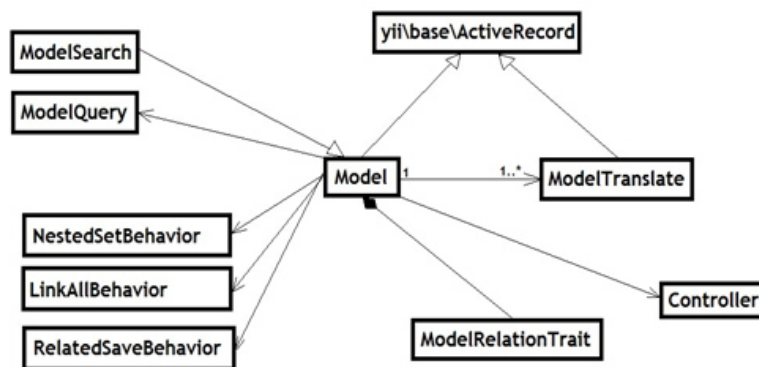


Рисунок 1 - Діаграма класів модуля

На діаграмі класів видно, що головним класом- «диригентом» є клас моделі сутності – він містить основну частину бізнес-логіки модуля – взаємодіє з компонентами системи, які займаються збереженням даних, кожен у своєму ключі – збереження дерева у моделі, збереження зв'язаних моделей, збереження мультимовних полів; містить в собі зв'язки з іншими модулями системи (за це відповідає ModelRelationTrait) та займається пошуком по даним сутності (що включає пошук по головній таблиці та перекладам).

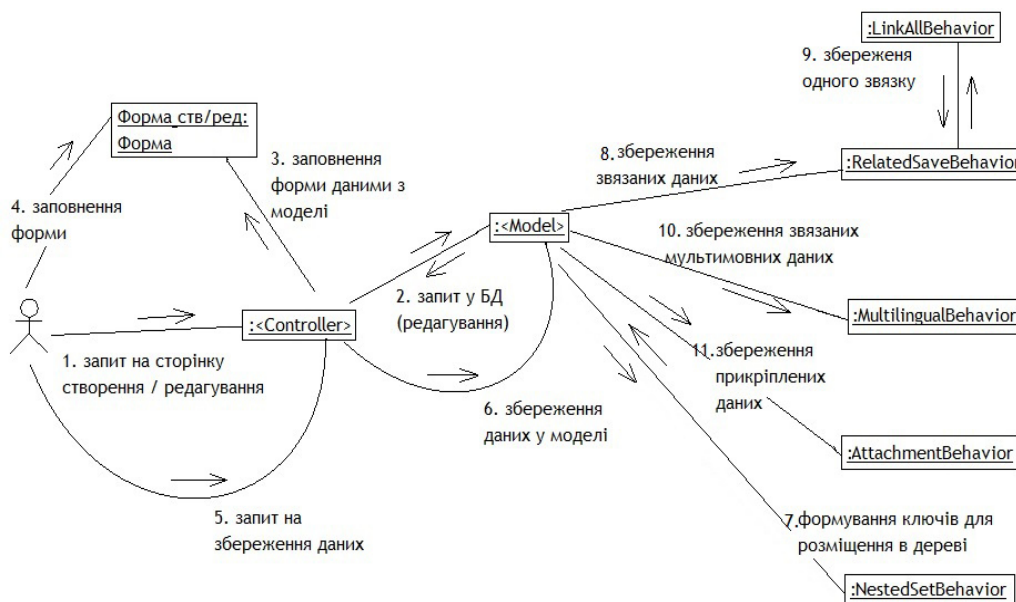


Рисунок 2 - Діаграма кооперації

На наведеній вище діаграмі кооперації зображений прецедент – створення/редагування сутності. У цьому прецеденті взаємодіють такі об'єкти, як контролер, що приймає запити від користувача, та в свою чергу робить запит на модель на отримання даних; модель та форма створення/редагування. Коли користувач посилає запит на збереження даних, цим займається об'єкт моделі, яка окрім збереженням даних власної моделі, керує збереженням перекладів, модифікації дерева, прикріплених файлів та зв'язків з іншими моделями системи.

Реалізоване рішення дозволяє суттєво зменшити час та витрати на прототипування і реалізацію веб-сайтів.

Висновок

У роботі створено шаблон веб-модуля засобами фреймворку Yii та мови програмування PHP, який надає змогу програмісту автоматично генерувати веб-компоненти і зв'язки між ними. Реалізоване рішення дозволяє суттєво зменшити час та витрати на прототипування і реалізацію веб-сайтів.

Список використаних джерел

1. Шальго А.А. Технология автоматного программирования // Мир ПК. 2003. № 10, с.74–78
2. Селлз К. Современные способы автоматизации повторяющихся задач программирования //MSDN Magazine. 2002, № 6
3. <http://www.yiiframework.com>
4. <http://rstdn.ru/article/patterns/generic-mvc.xml>