

ДОСЛІДЖЕННЯ МОЖЛИВОСТІ ВИКОРИСТАННЯ СКІНЧЕННИХ АВТОМАТІВ ДЛЯ РОЗРОБКИ WEB-ЗАСТОСУНКІВ

Запорізька державна інженерна академія, кафедра ПЗАС

Автомат (від грец. Αὐτόματος - самодіючий) - абстракція, що використовується для описання шляху зміни стану об'єкта в залежності від досягнутого стану та інформації отриманої ззовні.

Скінченний автомат (СА), є особливим видом автомату. Його особливістю є скінченність множини станів автомату. Автомати використовуються у так званому автоматному програмуванні.

Автоматне програмування - це парадигма програмування, при використанні якої програма або її фрагмент осмислюється як модель якогось формального автомата.

Залежно від конкретного завдання в автоматному програмуванні можуть використовуватися як скінченні автомати, так і автомати складнішої структури [1].

Визначальними для автоматного програмування є наступні особливості:

- часовий період виконання програми розбивається на кроки автомата, кожен з яких представляє собою виконання певної (однієї і тієї ж для кожного кроку) секції коду з єдиною точкою входу;
- передача інформації між кроками автомата здійснюється тільки через явно визначену множину змінних, що мають назву «стан автомата».

Повністю виконання коду в автоматному стилі являє собою цикл (можливо, неявний) кроків автомата [3].

У час зростання можливостей web-застосунків виникає необхідність у більш формалізованому підході до їх проектування та тестування. Враховувати поточний стан системи у великому web-застосунку було б дуже зручно за допомогою скінченного автомату.

Ознаки необхідності автоматного підходу до розробки програми [4]:

- Додавання поля *state* або *status* в модель.
- Логічні поля(*published, paid* тощо), тимчасові мітки, які можуть мати значення NULL.
- Записи, які дійсні лише протягом певного періоду часу (*subscriptions* з датами початку та закінчення тощо).

Проте застосовується такий підхід не досить широко. Поширені причини відмови від використання скінченного автомату:

- Думка, що доки робота незакінчена, автомат не потрібен.
- Думка, що використання автомату є надто складним.
- Думка, що навіть використання сторонньої бібліотеки з автоматом є досить складним.

У PHP є реалізація скінченного автомату у вигляді класу пакету PEAR під назвою FSM (Finite State Machine), який у свою чергу базується на FSM Python class [2].

Це проста бібліотека з відкритим кодом, яка містить лише основні функції і поля: конструктор з визначенням початкового стану й змінної для збереження корисного навантаження, функцію визначення переходів, функцію переходів і функцію визначення переходів за замовчуванням.

Подібні бібліотеки існують й на інших мовах програмування придатних для створення web-застосунків. Зокрема бібліотека *state_machine* у *Ruby*.

Для розроблення застосунку було вирішено використовувати фреймворк CodeIgniter - популярний MVC фреймворк з відкритим вихідним кодом, написаний на мові програмування PHP, для розробки повноцінних веб-систем і додатків [5].

У дипломній роботі використані наступні можливості CodeIgniter:

- Підтримка баз даних MySQL.
- Легко розширювана. Були додані додаткові helper та view для створення вкладених шаблонів.
- Підтримка як сегментованих ЧПУ і звичайних URL -ів з передачею параметрів.
- Фреймворк містить в собі безліч бібліотек. Були використані бібліотеки ActiveRecord, Cart, Email, Loader, Session, URI .
- Можливість використовувати шаблонізатор.

Висновки:

1. Детально розглянуті особливості підходу автоматного програмування; вивчено історію такого підходу, типові приклади застосування і розглянуто приклади коду; досліджено особливості використання такого підходу безпосередньо у web-програмуванні;
2. Досліджена бібліотека з відкритим кодом Finite State Machine мови PHP та розглянуто відповідні бібліотеки на інших мовах придатних для створення web-застосунків;
3. Проведене детальне дослідження деяких популярних php-фреймворків, таких як Zend Framework, Code Igniter 2 та Yii 1.1. Під час цього дослідження було виявлено, що Code Igniter є найбільш підходящим для невеликих проектів та для людей, які не мають великого досвіду у розробці на мові php. Цей фреймворк є майже мікрофреймворком. Він дуже легкий для вивчення. Має відмінну документацію. Гнучкий. У ньому легко використовувати сторонній код. На відміну від цього фреймворку Zend Framework використовується для середніх та великих проектів, не досить добре документований та має велику складність установки та налаштування. Yii фреймворк простіший для вивчення, ніж Zend Framework, але потребує більших знань ніж Code Igniter. Він підходить як для великих, так і для малих проектів. Має середню складність установки. З огляду на оброблену інформацію було вирішено використовувати фреймворк Code Igniter.

Література

1. А. Ахо, Дж. Ульман Теория синтаксического анализа, перевода и компиляции = The theory of parsing, translation and compiling. —М.: МИР, 1978. — Т. 1. — 612 с.
2. Postel, J., ed., Transmission Control Protocol, RFC 793
3. А. Оллонгрен Определение языков программирования интерпретирующими автоматами = Definition of programming languages by interpreting automata. — М.: МИР, 1977. — 288 с.
4. Туккель Н.И., Шалыто А.А. Программирование с явным выделением состояний // Мир ПК. — 2001. — № 9. — С. 132—138.
5. [Електронний ресурс]- Режим доступу: <http://cidocs.ru/213/tutorial/conclusion.html>