

6. Фельдштейн Д.І. Психологія особистості, що розвивається. / Д.І. Фельдштейн. — М.: Видавництво «Інститут практичної психології», 1996. – 512 с.

АВТОМАТИЗАЦІЯ ПРОЦЕСУ ПРОГРАМУВАННЯ НА ПРИКЛАДІ ГЕНЕРАТОРА АДМІНІСТРАТИВНОЇ ПАНЕЛІ НА ОСНОВІ SPA

Жильников Артем Сергійович,

*Студент 5 курсу напряму підготовки
«Програмна інженерія» Херсонського
державного університету*

Круглик Владислав Сергійович

*доктор педагогічних наук, доцент кафедри інформатики,
програмної інженерії та економічної кібернетики
Херсонського державного університету*

Анотація. Стаття присвячена розгляду змісту та переваг генерації коду. Автор виділяє проблеми, які вирішує генератор коду, особливості створення генератора коду, а також обмеження, що виникають при створенні такого генератора.

Ключові слова: програмне забезпечення, spa, генератор, stud, адміністративна панель.

Автоматичне програмування, у галузі комп'ютерних наук, - це термін, що визначає тип комп'ютерного програмування, в якому певний механізм генерує комп'ютерну програму, яка дозволяє програмістам писати код на більш високому рівні абстракції [1].

Міддред Косс, програміст UNIVAC, пояснив: "Написання машинного коду передбачало кілька трудомістких кроків - розбиття процесу на дискретні інструкції, присвоєння певних розташувань пам'яті всім командам і керування буферами вводу-виводу. Математичні процедури, бібліотека підпрограм і сортування програм, наше завдання полягало в тому, щоб розглянути більш широкий процес програмування. Нам потрібно було зрозуміти, як ми можемо повторно використовувати тестовий код і отримувати допомогу в програмуванні. Ми розглянули можливість абстрагувати ці кроки, щоб включити їх у мову більш високого рівня, і це призвело до розробки інтерпретаторів, компіляторів і генераторів - програм, призначених для роботи або створення інших програм, тобто автоматичного програмування." [2]

У нашому конкретному випадку мова піде саме про генератори. Більшість веб-розробників знайомі з генераторами html сторінок, так як майже всі фреймворки, призначені для розробки веб-додатків, підтримують генерацію динамічних веб-сторінок. Серед найбільш відомих: Jinja для Python; Razor для C#; Twig, Smarty, Blade для PHP; Handlebars, Underscore, Jade для JavaScript.

Навіщо взагалі потрібні генератори коду? По-перше, вони значно полегшують розробку деякої частини веб-додатка. По-друге, вони вирішують

однотипні задачі, які постають перед програмістом. І по-третє, генератори коду дозволяють сфокусуватися на плануванні та розробці унікальної бізнес-логіки продукту, абстрагуючись від компонентів для багаторазового використання, які саме і піддаються генерації.

Як програмісти, ми зіштовхнулись з низкою однотипних задач, що виникали як при розробці учбових проєктів в університеті, так і в реальних комерційних проєктах. Одну з них ми вирішили розв'язати шляхом створення генератора адміністративної панелі, що включає авторизацію та CRUD-операції для складових заданої бази даних.

Як було вже зазначено, існує багато технологій для створення динамічних веб-сторінок. Крім того, існують технології, які дозволяють генерувати більш цікаві речі, такі як контролери, моделі, валідатори та інше. Прикладами таких програм є `Wake`, яка входить до фреймворку `CakePHP`, а також `Rails`, яка входить до фреймворку `Ruby on Rails`. По суті це генератори, які дозволяють створити не тільки шаблони для відображення, а й повноцінний веб-додаток.

Односторінковий веб-додаток (SPA) - це саме той підхід, який часто використовується для розробки таких веб-додатків, як адміністративні панелі. Причиною тому є безліч переваг SPA над традиційним багатосторінковим веб-додатком. Головним недоліком SPA є слабка оптимізація для пошукових систем, а для адміністративної панелі це не важливо.

На сьогоднішній день є 3 найбільш популярні технології для створення SPA - це `Angular`, `React` і `Vue`. Остання підходить для невеликих веб-додатків і дуже легка у вивченні, тому прийнятна для більшості початківців. Ми обирали між `Angular` і `React`, оскільки вони підтримують розробку нативних додатків для мобільних систем.

`React` дозволяє розробникам створювати великі веб-програми, які можуть змінювати дані без перезавантаження сторінки. Основна мета `React` - бути швидким, масштабованим та простим. `React` також дозволяє створювати повторно використовувані компоненти інтерфейсу, що стало одним з переконуючих аргументів на його користь. Більше того, `React` використовує спеціальний синтаксис під назвою `JSX`, який дозволяє змішувати `HTML` з `JavaScript`. Цей синтаксис полегшує розробку компонентів для багаторазового використання. Веб-додатки, написані на `React`, дуже легко тестувати. Компоненти можуть розглядатися як функції, які залежать від поточного стану програми. Маніпулюючи цим станом, ми можемо спостерігати за зміною вигляду та/або поведінки окремого компонента, спрацюванням деяких функцій, подій тощо.

Компонентний підхід, чітко визначений життєвий цикл і використання простого `JavaScript` роблять `React` дуже простим в освоєнні та створенні професійного веб-сайту (і мобільного додатку) і підтримуванні його. `React` створює кеш структури даних у пам'яті, які обчислюють внесені зміни, а потім оновлюють браузер. Це одна із особливостей, яка дозволяє програмісту кодувати так, ніби вся сторінка перетворюється на окрему змінну, а бібліотека займається рендерингом компонентів, які фактично змінюються.[3]

Порівняння Angular та React

	Angular	React
Тип	Фреймворк	Бібліотека для побудови UI
Причина вибору	TypeScript та об'єктно орієнтований підхід	Підхід “все в javascript”
Засновник	Google	Facebook
Дата релізу	Вересень 2016	Березень 2013
Структура	Чітка структура, нав'язана фреймворком	Вільна, гнучка структура проекту

Для перевірки прав користувача на доступ до адміністративної панелі було обрано JWT авторизацію. JWT (веб-токен JSON) - це відкритий стандарт (RFC 7519), який визначає компактний та автономний спосіб безпечного передавання інформації між сторонами як об'єкт JSON. Цю інформацію можна перевірити та довіряти їй, оскільки вона підписана цифровим способом. JWT можна підписати за допомогою секретного ключа (з алгоритмом **HMAC**) або пари публічних/приватних ключів, використовуючи **RSA** . [4]

В якості серверної частини було обрано NodeJS, а саме фреймворк Express. Він дозволяє дуже швидко розробити API для клієнтської частини. Якщо не вдаватися у дрібниці, то ExpressJS зробив для NodeJS те, що Bootstrap зробив для HTML / CSS і чуйного дизайну. Цей фреймворк надав кодуванню в NodeJS більше елегантності, а програмістам додаткові можливості для розширення кодування на базі сервера. [5]

В якості ядра для генерації повноцінного додатка із заданих шаблонів виступає Handlebars. Це двигун для генерації вихідного коду на основі запропонованих шаблонів. Зазвичай будь-яка логіка в таких шаблонах відсутня.

Отже, дослідження існуючих технологій для створення адміністративної панелі допомогли нам зрозуміти, як правильно створити надійну та масштабовану систему. Кожну програмну частину адміністративної панелі було уніфіковано. Таким чином ми отримуємо шаблони, на основі яких генератор здатний створити повноцінний веб-додаток, що покриває CRUD операції для заданої бази даних. Ключовою особливістю такого підходу є те, що програміст отримує клієнтську та серверну частини, які він з легкістю здатен розширити та модифікувати за своїми потребами.

Література:

1. Ricardo A. M. Automatic Inductive Programming. ICML Tutorial: 2006.
2. Wendy H. K. On Software, or the Persistence of Visual Knowledge. Бостон: 2004. 30 с.

3. What Is ReactJS and Why Should We Use It? . – [Електронний ресурс] – Режим доступу до ресурсу: <https://www.c-sharpcorner.com/article/what-and-why-reactjs> (дата звернення 12.05.2019).
4. JSON Web Token Introduction. . – [Електронний ресурс] – Режим доступу до ресурсу: <https://jwt.io/introduction/> (дата звернення 12.05.2019)
5. Advantages of Using ExpressJS | Impressico Business Solutions. – [Електронний ресурс] – Режим доступу до ресурсу: <https://www.impressico.com/2015/10/06/advantages-of-using-express-js/> (дата звернення 12.05.2019)

ВИВЧЕННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ЯКОСТІ ОБ'ЄКТА ВИВЧЕННЯ І ЗАСОБИ НАВЧАННЯ

*Зелінська Сніжана Олександрівна,
докторант, кандидат педагогічних наук
Криворізький державний педагогічний університет*

Анотація. Розгляд можливостей розвитку методичної системи навчання інформатики за рахунок навчання технології доповненої реальності і застосування відповідних засобів навчання повинно передувати вивчення історично сформованих і сучасних підходів до навчання студентів інформаційних технологій. Подібний аналіз може сприяти обґрунтуванню доцільності і можливості запропонованого вдосконалення.

Зараз відбувається подальше становлення інформаційного суспільства, в якому домінує вид діяльності пов'язаний з виробництвом інформаційного продукту. Інформація зараз є таким самим стратегічним ресурсом суспільства, як продукти харчування в «аграрному», а матеріальні та енергетичні ресурси в «індустріальному» суспільстві.

Ключові слова. Інформатизація, засоби інформаційних технологій, технологія доповненої реальності.

Однією з ключових областей інформатизації суспільного життя є інформатизація освіти. Від оволодіння студентами узагальненими способами діяльності з використанням інформаційних технологій залежить, як ефективність їх навчання, так і підготовленість студентів до життєдіяльності в сучасному суспільстві.

Теоретичні дослідження питань необхідності використання і застосування засобів інформаційних технологій в навчальному процесі описані в роботах М. Абдуразакова, С. Атанасян, Т. Бороненко, В. Бубнова, В. Гріншкуна, І. Захарової, С. Каракозова, А. Кондакова, О. Смолянинової, М. Сурхаєва, Є. Хеннера і ін. [1-5; 9].

У нашій країні навчання інформаційним технологіям, в первинних умовах лише технології програмування, почалося практично одночасно з зародження інформатики (по-іншому кібернетики, computer science [6]) як науки, коли були опубліковані праці Н. Вінера, Д. Неймана, К. Шеннона (середина ХХ століття) [7]. Експериментальна робота з навчання програмування під керівництвом О. Єршова, С. Шварцбурда і ін. [8] (кінець 50-х років ХХ століття) показала можливість і необхідність такої діяльності.