

ТЕХНІЧНІ НАУКИ

DOI: <https://doi.org/10.33216/1998-7927-2023-280-4-101-106>

УДК 004.41

ОДИН З ПІДХОДІВ ВИКЛАДАННЯ КУРСУ «РЕАКТИВНЕ ПРОГРАМУВАННЯ»

Полупан Ю.В., Родіонов П.Ю.

ONE OF THE APPROACHES TO TEACHING THE COURSE «REACTIVE PROGRAMMING»

Polupan Yu.V., Rodionov P.Yu.

Сучасному бізнесу необхідні програмні системи нового типу, здатні залишатися чутливими при будь-яких навантаженнях. Цю потребу можна задовольнити, використовуючі прийоми реактивного програмування; однак розробка таких систем – складне завдання, яке потребує глибокого розуміння предметної галузі. Тому для бажаних статистичних показників класичними програмістами-кодерами, вивчення основ використання реактивного підходу при створенні високонавантажених додатків є наразі актуальною задачею.

Дисципліна «Реактивне програмування» входить у комплекс наскрізної безперервної підготовки спеціалістів за освітньою програмою «Інженерія програмного забезпечення інформаційних систем» спеціальності 121 «Інженерія програмного забезпечення». У структурно-логічній схемі навчання дисципліна вивчається на етапі підготовки фахівців освітньо-кваліфікаційного рівня «бакалавр». Вивчення дисципліни базується на знаннях, навичках та вміннях, отриманих під час вивчення студентами дисциплін «Основи програмування», «Основи комп'ютерних систем і мереж», «Технології паралельних обчислень», «Функціональне програмування».

Однак існує багато підходів до вивчення реактивного програмування. Можна вивчати більшу теоретичну базу, включаючи поведінкові патерни, такі як *Observer* та *Iterator*, які лежать в основі реактивності, а можна розглядати реактивність в практичній площині, вивчаючи готові

інструментарії (реактивні розширення та фреймворки), завдяки яким створюються незалежні та високопродуктивні реактивні додатки.

В статті обґрунтовується один із підходів викладання дисципліни «Реактивне програмування» у вищих навчальних закладах, та наводиться інструментарій для більш ефективного розуміння слухачами основ реактивності. При вивченні дисципліни пропонується більш практичний підхід з використанням веб-фреймворка *Angular*, в основі якого лежить бібліотека *RxJS* з її можливістю створювати реактивний код. При цьому студенти навчаються обробляти асинхронні виклики за допомогою безлічі подій та створювати веб-додатки, використовуючі популярний фреймворк *Angular* від компанії *Google*.

Популярність фреймворку обумовлена декількома важливими аспектами, головним серед яких можна назвати зручність для розробника. Програми на *Angular* пишуться порівняно швидко, код має зрозумілу структуру. *Angular* підходить для написання додатків під різні платформи, має величезну навчальну базу та чималу спільноту, до нього можна підключати сторонні інструменти та модулі, розширюючи і без того великий функціонал.

Ключові слова: реактивне програмування, *reactive extensions*, *ReactiveX*, *Rx*, *RxJS*, *RxJava*, *Angular*, *React*, фреймворк, *Observable*, *Observer*, *Subscriber*, *Spring*, *TypeScript*, *JavaScript*, високонавантажені додатки.

Вступ. З розвитком сучасних програмних додатків потреба у чутливих, масштабованих і високопродуктивних рішеннях стає дедалі гострішою. Реактивне програмування стало потужною парадигмою, що допомагає розробникам створювати ефективні реактивні системи, здатні безперешкодно обробляти події у реальному часі та адаптуватися до динамічних змін. Зростаюча популярність реактивного програмування у спільноті розробників та його інтеграція до різних бібліотек та фреймворків свідчать про значну роль цієї парадигми у визначенні майбутнього розробки програмного забезпечення.

Реактивне програмування знаходить все ширше застосування у різних мовах програмування та фреймворках, оскільки розробники усвідомлюють його потенціал для створення ефективних та масштабованих додатків. Хоча існує багато популярних бібліотек для створення реактивного коду на різних мовах, впровадження реактивного програмування неухильно зростає на рівні мов та фреймворків.

Тому включення дисципліни «Реактивне програмування» у цикл підготовки фахівців за спеціальністю «Інженерія програмного забезпечення», так би мовити класичних програмістів-кодерів, наразі є бажаним і актуальним, а також сприяє тому, щоб допомогти слухачам почати думати по-справжньому "реактивно".

Мета роботи. Метою дисципліни «Реактивне програмування» є набуття ключових фахових компетентностей, теоретичних знань і практичних навичок з технологій веб-розробки реактивних додатків. Предметом навчальної дисципліни є ключові структури даних, характерні для реактивного програмування; найпоширеніші бібліотеки, методи і засоби створення реактивних, дієво-орієнтованих додатків; теоретичні засади та використання реактивного підходу для вирішення прикладних задач. За результатами вивчення дисципліни студент повинен набути досвід: зі створення та опрацювання асинхронних потоків даних; з підготовки середовища розробки для створення реактивних веб-додатків на базі відомих фреймворків; з використання реактивних бібліотек для розробки дієво-орієнтованих програм.

Метою цієї роботи є вибір інструментарію для забезпечення ефективного вивчення дисципліни «Реактивне програмування» з

набуттям студентами відповідних умінь, знань і досвіду.

Виклад основного матеріалу Навчання реактивному підходу у програмуванні — досить не проста річ, і нестача навчальних матеріалів лише посилює цей процес. Більшість існуючих посібників [1, 2, 3] не дають глибокого огляду і не розповідають про те, як спроекувати архітектуру проекту в цілому.

Реактивне програмування (RP) — це парадигма програмування, в якій основна увага приділяється потокам даних та їх автоматичній обробці у реальному часі [4, 5, 6]. RP має на увазі швидку реакцію програми на зміни, а також використання асинхронних операцій.

Основні принципи RP — це створення реактивних функцій, керування подіями та їх обробка в реальному часі, а також використання патернів проектування, таких як Observer, для зручного керування потоками даних.

Приклади використання RP — це веб-додатки з динамічними інтерфейсами, де дані оновлюються в реальному часі, ігри з інтерактивною графікою, а також програми для обробки великих обсягів даних, наприклад, у фінансовій сфері. RP широко застосовується у мовах програмування, таких як Java, JavaScript, C#, Kotlin та інш.

Таким чином, сучасні програми повинні бути чутливі і мати можливість обробляти дані з різних джерел одночасно. Існуючі на даний момент техніки у сучасному JavaScript ні як не приведуть нас до бажаного. Вони не дозволяють масштабувати код тому, що цей код експоненційно ускладнюється, адже часто до нього додається паралелізм.

У реактивному програмуванні основною парадигмою є - більш простим та природним способом працювати з асинхронним кодом. При цьому потоки подій (Observables) чудовим способом вирішують такі задачі [4]. За допомогою Observable можна створювати об'єкти, які ініціюють асинхронні потоки, а "реактивне мислення" разом з реактивними розширеннями (ReactiveX або просто Rx) значною мірою покращують існуючі техніки [2].

Для ефективної обробки даних та подій реактивне програмування використовує видавців та підписників. Модель "видавець-підписник" застосовується у реактивному програмуванні так:

- ініціалізується об'єкт-видавець, який з часом видає потік даних чи подій;
- створюється один або кілька підписників, які реєструються у видавця і отримують

повідомлення про появу нових даних. Підписники можуть обробляти дані асинхронно та відповідним чином оновлювати стан програми. Можна налаштувати так, щоб подія отримувалась одним або кількома підписниками залежно від конкретного випадку використання;

- підтримка операторів-функцій, що використовуються для перетворення та об'єднання потоків даних. Оператори дозволяють розробникам створювати складні потоки даних шляхом компонування та перетворення потоків даних із застосуванням набору прийомів функціонального програмування;

- для забезпечення масштабованості використовуються планувальники, які допомагають керувати багатозадачністю та паралелізмом у додатку. Планувальники дозволяють розробникам контролювати обробку потоків даних і відповідають за чутливість та масштабованість програми.

Реактивне програмування може бути здійснено кількома підходами [5, 6]:

- імперативним програмуванням;
- об'єктно-орієнтованим;
- функціональним.

Однак найбільш природним базисом для реактивного програмування та роботою з реактивними структурами даних є функціональний підхід. Популярна бібліотека, яка використовує цей підхід – це RxJS, яка була взята за основу такого фреймворку як Angular.

Таким чином, можна сказати, що «реактивність» з точки зору користувача – це можливість програмного засобу миттєво реагувати на його дії. Наприклад, почали вводити текст, він одразу ж з'являється у полі. Натиснули на кнопку, форма відразу змінилася, з'явилися ладери і т.д. Фреймворки, що дозволяють створювати додатки, у яких стан інтерфейсу автоматично реагує на зміну даних, називаються реактивними. Це такі впливові фреймворки, як Spring та Angular. Вони використовують принципи реактивного програмування для підтримки розробників у створенні неблокуючих додатків з винятковою чутливістю та масштабованістю. Менш відомі фреймворки такі як React, Vue, Svelte також знаходять своїх пошанувачів та нішу у веб-розробці.

Для розробника термін «реактивність» має трохи інше значення. В даному випадку фокус зміщується на дані і реактивність розглядається як робота з асинхронними потоками даних,

наприклад, при допомозі ReactiveX, до яких належить бібліотека RxJS.

RxJS – це Javascript бібліотека для трансформації, складання та вилучення асинхронних потоків даних [7]. Вона може бути використана як у браузері, так і на стороні сервера, спрощує управління складними потоками даних за рахунок впровадження реактивних спостережуваних, pipeable операторів та планувальників для обробки паралелізму та управління ресурсами. Список популярних Rx-бібліотек доповнюють RxJava, RxSwift, RxScala, RxKotlin, RxPy і т.д. ReactiveX підтримує безліч мов програмування, включаючи Java, JavaScript, C#, Python та інші. [4, 5, 6]. Ці бібліотеки надають різні можливості для роботи з реактивними потоками даних та обробки помилок.

Angular - це потужний фреймворк веб-додатків з відкритим вихідним кодом, створений командою Angular у Google та спільнотою розробників-ентузіастів [8, 9, 10, 11]. Цей популярний JavaScript фреймворк є комплексним рішенням для створення ефективних і складних односторінкових веб-додатків (SPA).

Angular виділяється своїм багатим набором функцій, таких як двостороння прив'язка даних, модульна структура розробки, обробка форм, маршрутизація та впровадження залежностей, які оптимізують процес розробки та підвищують продуктивність [10, 11, 12, 13]. На відміну від бібліотек, орієнтованих виключно на рівень представлення, Angular надає повноцінний фреймворк, включаючи інструменти для тестування, розробки та розгортання програм. Він також використовує TypeScript для забезпечення узгодженості коду, покращення читаності та надійної перевірки типів. Компонентна архітектура Angular забезпечує багаторазове використання коду та ефективні методи розробки. Цей фреймворк особливо популярний у додатках корпоративного масштабу через його масштабованість, ремонтпридатність та сильний акцент на тестування і кращі практики кодування.

Якщо порівняти Angular з React, то його структура є значно складнішою [11, 14, 15, 16]. Разом з компонентами, тут також з'являються наступні елементи, кожен зі своїм призначенням:

- Пайпи — свого роду допоміжні функції для трансформації візуального вигляду даних у HTML темплейтах. Наприклад: форматування чисел, дат, валют тощо.

- Директиви — класи, котрі використовуються для надання додаткового функціонала іншим елементам застосунку. І також поділяються на кілька видів, в залежності від форми взаємодії з елементами. В основному застосовуються у вигляді користувацьких атрибутів елементів розмітки.

- Компоненти — є елементом для візуального зображення даних. Поєднують в собі стилі, HTML-файл з розміткою, а також JS або TS з логікою компонента.

- Сервіси — для можливості централізованого зберігання даних і утилітарних методів. Можуть використовуватись Пайпами, Директивами, Компонентами й навіть іншими Сервісами, додаючи їх за допомогою Dependency Injection.

- Модулі — спосіб згрупувати всі вищеперелічені елементи, в більшості випадків, по функціоналу або ж сторінках.

Як було сказано вище, в основі фреймворку Angular лежить бібліотека RxJS та принципи, що дають можливість створювати функціональний код, заснований на реактивному підході, що, в свою чергу, призводить до декларативної парадигми - код легко покривається unit-тестами, а засоби для тестування йдуть із коробки.

Висновки. Таким чином, можна сказати, що реактивне програмування - це потужний інструмент, який допомагає розробникам створювати більш ефективні та чутливі системи. З його допомогою створюються додатки, які реагують на зміни у навколишньому середовищі та швидко адаптуються до нових вимог. Однак, як і будь-яка інша технологія, реактивне програмування не є універсальним рішенням для всіх завдань і потребує ретельної оцінки та розуміння під час використання.

Одним із засобів створення реактивних додатків є фреймворк Angular в основі якого лежить бібліотека RxJS [17], тому при вивченні дисципліни «Реактивне програмування» пропонується використовувати саме цей фреймворк, враховуючи його популярність і те, що він підтримується компанією Google, яка забезпечує довгострокову підтримку Angular та можливість подальшого розширення екосистеми, а також має відмінну підтримку спільноти, яка допоможе вирішити будь-яке технічне питання.

При вивченні дисципліни з таким інструментарієм студенти навчаються створювати односторінкові веб-додатки з дотриманням основних принципів та механізмів використання

методик реактивного програмування взагалі та RxJS зокрема.

Л і т е р а т у р а

1. Lamis Chebbi "Reactive Patterns with RxJS for Angular", Published by Packt Publishing Ltd, 2022.
2. Sergi Mansilla "Reactive Programming with RxJS". The Pragmatic Programmers, 2015.
3. Дворецький М.Л., Дворецька С.В. Розробка односторінкових вебзастосунків та адаптивних інтерфейсів. Навчальний посібник. Миколаїв, Чорноморський університет імені Петра Могили, 2020. 69 с.
4. Praseed Pai, Peter Abraham «C++ Reactive Programming: Design concurrent and asynchronous applications using the RxCpp library and Modern C++17» // Packt Publishing, 2018. 348 pages.
5. Ćukić, I. (2019). Functional Reactive Programming in C++. In: Zsóck, V., Porkoláb, Z., Horváth, Z. (eds) Central European Functional Programming School. CEFP 2015. Lecture Notes in Computer Science(), vol 10094. Springer, Cham. https://doi.org/10.1007/978-3-030-28346-9_2.
6. Tomasz Nurkiewicz, Ben Christensen «Reactive Programming with RxJava: Creating Asynchronous, Event-Based Applications» 1st Edition // O'Reilly Publishing, 2016. 372 pages.
7. Guide of RxJS [Online] Available from: <https://rxjs.dev/guide/overview> [Accessed Dec 10, 2022].
8. BampakosA., DeelemanP. Learning Angular. Packt Publishing, 2020.
9. ChivukulaS.R., Iskandar A. «Web Development with Angular and Bootstrap» // Packt Publishing, 2019.
10. Doguhan Uluca «Angular for Enterprise-Ready Web Applications» // Packt Publishing, 2020.
11. Adam Freeman, «Pro Angular. Build Powerful and Dynamic Web Apps». Fifth Edition. Publisher: Apress Berkeley, CA. 2022, 880 pages. DOI: <https://doi.org/10.1007/978-1-4842-8176-5>.
12. Adam Freeman «Pro Angular 9. Build Powerful and Dynamic Web Apps» // Publisher: Apress Berkeley, CA. 2020, 784 pages. DOI: <https://doi.org/10.1007/978-1-4842-5998-6>.
13. Adam Freeman, «Pro Angular. Learn to harness the power of modern web browsers from within your application's code» // Second Edition. Publisher: Apress Berkeley, CA. 2017, 788 pages. DOI: <https://doi.org/DOI.10.1007/978-1-4842-2307-9>.
14. M. Holmes He «Creating Apps with React Native. Deliver Cross-Platform 0 Crash, 5 Star Apps» // Publisher: Apress Berkeley, CA. 2022, 436 pages. DOI: <https://doi.org/10.1007/978-1-4842-8042-3>.
15. Nabendu Biswas «Beginning React and Firebase. Create Four Beginner-Friendly Projects Using React and Firebase» // Publisher: Apress Berkeley,

- CA. 2022, 184 pages. DOI: <https://doi.org/10.1007/978-1-4842-7812-3>.
16. Hari Narayn «Just React! Learn React the React Way» // Publisher: Apress Berkeley, CA. 2022, 369 pages. DOI: <https://doi.org/10.1007/978-1-4842-8294-6>.
17. The web development framework for building the future [Online] Available from: <https://angular.io> [Accessed Oct 25, 2022].

References

1. Lamis Chebbi "Reactive Patterns with RxJS for Angular", Published by Packt Publishing Ltd, 2022.
2. Sergi Mansilla "Reactive Programming with RxJS". The Pragmatic Programmers, 2015.
3. Dvoretzkyi M.L., Dvoretzka S.V. Rozrobka odnistorinkovykh vebzastosunkiv ta adaptivnykh interfeisiv. Navchalnyi posibnyk. Mykolaiv, Chornomorskyi universytet imeni Petra Mohyly, 2020. 69 s.
4. Praseed Pai, Peter Abraham «C++ Reactive Programming: Design concurrent and asynchronous applications using the RxCpp library and Modern C++17» // Packt Publishing, 2018. 348 pages.
5. Ćukić, I. (2019). Functional Reactive Programming in C++. In: Zsóka, V., Porkoláb, Z., Horváth, Z. (eds) Central European Functional Programming School. CEFP 2015. Lecture Notes in Computer Science(), vol 10094. Springer, Cham. https://doi.org/10.1007/978-3-030-28346-9_2.
6. Tomasz Nurkiewicz, Ben Christensen «Reactive Programming with RxJava: Creating Asynchronous, Event-Based Applications» 1st Edition // O'Reilly Publishing, 2016. 372 pages.
7. Guide of RxJS [Online] Available from: <https://rxjs.dev/guide/overview> [Accessed Dec 10, 2022].
8. Bampakos A., Deeleman P. Learning Angular. Packt Publishing, 2020.
9. Chivukula S.R., Iskandar A. «Web Development with Angular and Bootstrap» // Packt Publishing, 2019.
10. Doguhan Uluca «Angular for Enterprise-Ready Web Applications» // Packt Publishing, 2020.
11. Adam Freeman, «Pro Angular. Build Powerful and Dynamic Web Apps». Fifth Edition. Publisher: Apress Berkeley, CA. 2022, 880 pages. DOI: <https://doi.org/10.1007/978-1-4842-8176-5>.
12. Adam Freeman «Pro Angular 9. Build Powerful and Dynamic Web Apps» // Publisher: Apress Berkeley, CA. 2020, 784 pages. DOI: <https://doi.org/10.1007/978-1-4842-5998-6>.
13. Adam Freeman, «Pro Angular. Learn to harness the power of modern web browsers from within your application's code» // Second Edition. Publisher: Apress Berkeley, CA. 2017, 788 pages. DOI: <https://doi.org/10.1007/978-1-4842-2307-9>.
14. M. Holmes He «Creating Apps with React Native. Deliver Cross-Platform 0 Crash, 5 Star Apps» // Publisher: Apress Berkeley, CA. 2022, 436 pages. DOI: <https://doi.org/10.1007/978-1-4842-8042-3>.
15. Nabendu Biswas «Beginning React and Firebase. Create Four Beginner-Friendly Projects Using React and Firebase» // Publisher: Apress Berkeley, CA. 2022, 184 pages. DOI: <https://doi.org/10.1007/978-1-4842-7812-3>.
16. Hari Narayn «Just React! Learn React the React Way» // Publisher: Apress Berkeley, CA. 2022, 369 pages. DOI: <https://doi.org/10.1007/978-1-4842-8294-6>.
17. The web development framework for building the future [Online] Available from: <https://angular.io> [Accessed Oct 25, 2022].

Polupan Yu.V., Rodionov P.Yu. One of the approaches to teaching the course "Reactive programming"

Today's business needs a new type of software system that will no longer be sensitive to any demands. This needs can be satisfied by accepting reactive programming; However, the development of such systems is a complex task and requires a deep understanding of the subject matter. Therefore, for most classic coders, learning the fundamentals of a reactive approach when creating high-value add-ons is an urgent task.

The discipline "Reactive Programming" is included in the complex of thorough, continuous training of specialists in the major program "Software Information Systems Engineering", specialty 121 "Software Engineering". In the structural-logical scheme, the initial discipline is taught at the stage of training of fakivts of the educational qualification level "bachelor". The discipline is based on the knowledge, skills and intelligence acquired by students in the disciplines "Fundamentals of Programming", "Fundamentals of Computer Systems and Measurements", "Parallel Computing Technologies", "Functional Programming".

However, there are many approaches to the development of reactive programming. You can look at a more theoretical basis, including behavioral patterns such as Observer and Iterator, which underlie reactivity, and you can look at reactivity in a practical sense, including ready-made tools (reactive extensions and frameworks), this is how independent and highly productive reactive additives are created.

In this article, one of the approaches of the discipline "Reactive Programming" is grounded in the most basic principles, and tools are provided for a more effective understanding of the basic principles of reactivity. With this discipline, a more practical approach is adopted using the Angular web-framework, which is based on the RxJS library with its ability to create reactive code. With this, students will begin to develop asynchronous clicks for an additional

impersonal approach and create web add-ons that use the popular Angular framework from Google.

The popularity of the framework is due to several important aspects, the main of which can be called ease of use for the developer. Programs in Angular are written smoothly, the code has a clear structure. Angular is suitable for writing add-ons for a variety of platforms, it has a large initial base and great flexibility, so you can connect third-party tools and modules that expand the already great functionality.

Keywords: *reactive programming, reactive extensions, ReactiveX, Rx, RxJS, RxJava, Angular, React, framework, Observable, Observer, Subscriber, Spring, TypeScript, JavaScript, high-load applications.*

Полупан Юлія Вікторівна – канд. техн. наук, доц., доц., КПІ ім. Ігоря Сікорського, факультет інформатики та обчислювальної техніки, кафедра інформатики та програмної інженерії, juliy_polupan@i.ua

Родіонов Павло Юрійович – канд. екон. наук, доц., КПІ ім. Ігоря Сікорського, факультет інформатики та обчислювальної техніки, кафедра інформатики та програмної інженерії

Стаття подана: 28.11.2023.