УДК 681.3:658.5

Бублик В. В., Давиденко А. М.

КОЛАБОРАТИВНІ МЕТОДИ В ЕЛЕКТРОННОМУ НАВЧАННІ ПРОГРАМУВАННЯ

Стаття містить звіт про важливий напрям діяльності авторів, спрямований на створення та впровадження новітніх засобів розроблення програмних систем в університетські курси програмування на основі використання тісної співпраці учасників програмних проектів у рамках систем електронного навчання.

Ключові слова: електронна освіта, колаборативні середовища, програмування, групові проекти.

Вступ

Співпраця викладача зі студентом є запорукою успіху в кожній освітній системі, а застосування електронних методів навчання виводить її на якісно новий рівень. У роботах одного з авторів (див., наприклад, [1]) було розглянуто переваги електронного навчання програмування засобами системи MOODLE (https://moodle.org/), розгорнутої на факультеті інформатики Національного університету «Києво-Могилянська академія», до яких у першу чергу належить цілодобова доступність навчальних матеріалів, розвинені засоби контролю і зворотного зв'язку, обговорення в тематичних форумах тощо. Об'єднаними зусиллями викладачів і студентів було створено додаткові сервіси, спрямовані на інтенсифікацію самостійної роботи студентів. Це, зокрема, засоби взаємного рецензування проектів і виконання та перевірка проектів он-лайн.

Наявні засоби, як штатні в системі MOODLE, так і її розширення та доповнення, розраховані на індивідуальну роботу студента. Разом з цим наразі назріла потреба в електронній підтримці активно впроваджених на факультеті процесів навчання групового розроблення програм [2]. Навчальні проекти, а особливо такі, що є груповими, мають низку особливостей порівняно з промисловим розробленням. Насамперед, це прозорість процесу створення програмного коду як для спостереження з боку наставника чи інших членів команди розробників, так і для безпосереднього втручання керівника команди з порадами чи зауваженнями.

Як основний засіб роботи над програмним проектом зазвичай використовують інтегровані середовища розробки. Цілям навчального процесу найповніше відповідатиме колаборативне середовище розробки у вигляді віртуального простору, призначеного для співпраці учасників проекту. На відміну від традиційних інтегрованих середовищ розробки, колаборативні середовища мають вбудовані засоби комунікації, управління конфігураціями та системи баг-трекінгу. Найголовніше, що такі середовища розробки дають змогу багатьом програмістам редагувати програмний код одночасно у режимі реального часу. Традиційні середовища розробки акцентуються на підвищенні ефективності кожного розробника, натомість колаборативні середовища розробки – на підвищенні ефективності процесу розробки в команді в цілому [3].

Особливо популярними є колаборативні середовища розробки, доступ до яких надається за моделлю «програмне забезпечення як послуга» (Software as a Service), оскільки немає необхідності налаштовувати середовище розробки на кожному комп'ютері, з якого ведеться процес розроблення програмного забезпечення. Саме тому дослідження колаборативних середовищ розробки є актуальним завданням.

Особливості хмарних середовищ розробки для організації колаборативної розробки

На відміну від десктопних середовищ розробки, хмарні середовища розробки дають змогу миттєво здійснювати обмін програмним кодом та здійснювати комунікацію з іншими розробниками. Такі хмарні середовища розробки, як Cloud9, CodeAnywhere та Codenvy, дозволяють програмістам переглядати, редагувати і ділитися програмним кодом з іншими розробниками, що робить колаборацію простішою. У хмарному середовищі Cloud9 багато розробників можуть працювати над одним і тим самим програмним кодом одночасно, роблячи процес редагування

схожим на редагування документів у Google Docs чи Microsoft Office 365. Крім того, можливість доступу до середовища розробки з браузера дає змогу програмістам працювати з будь-якого комп'ютера без наявності потужного комп'ютера з налаштованим середовищем розробки, оскільки використання хмарного середовища розробки не потребує значних ресурсів комп'ютера для його роботи. Однак деякі хмарні середовища розробки мають помітну затримку на дії користувача. Це зумовлено необхідністю обробляти команди користувача на сервері і аж потім надсилати відповідь клієнтові. Також необхідність мати надійне та швидке Інтернет-з'єднання для роботи у хмарному середовищі розробки для мінімізації затримки.

Засоби створення колаборативного середовища розробки

Для побудови колаборативного середовища розробки перед авторами стояло завдання вибрати засіб для обміну даними між клієнтським і серверним застосунками в режимі реального часу. Очевидним є те, що традиційні методи опитування сервера про нові дані для відображення (polling та long polling) не є найкращими рішеннями через накладні витрати, що пов'язані з використанням заголовків протоколу http. До того ж такий зв'язок не буде двонаправленим і повнодуплексним.

Натомість, веб-сокети поєднують простоту використання, надійність та ефективність. Вони дають змогу забезпечити постійний двонаправлений зв'язок між клієнтським і серверним застосунками. Веб-сокети підходять для додатків, які отримують і відсилають велику кількість даних за короткий проміжок часу. Саме тому веб-сокети є оптимальним вибором для обміну повідомленнями між клієнтом та сервером при створенні колаборативного середовища розробки.

Для повноцінного функціонування колаборативного середовища розробки необхідно мати засіб для компіляції і виконання коду. Платформа Sphere Engine надає компілятори для більш ніж 60 популярних мов програмування. Значною перевагою є те, що з використанням Sphere Engine розробникам колаборативного середовища розробки не треба перейматися стосовно розгортання середовища на серверах для підтримки компіляції, його налаштування та створення API.

API компілятора Sphere Engine дає змогу:

- завантажувати код;
- запускати програму з вхідними даними на сервері з підтримкою більш ніж 60 популярних мов програмування;
- отримувати результат виконання (вивід, інформацію та помилки компіляції, час виконання, використання пам'яті тощо).

API Sphere Engine підтримує архітектуру RESTful для відправки та отримання запитів. З кожним запитом як параметр необхідно передавати access_token для ідентифікації користувача сервісу. Значення access_token можна отримати в особистому кабінеті Sphere Engine після реєстрації та підтвердження особового профілю користувача.

До особливостей API Sphere Engine можна віднести те, що він працює лише з однофайловими завантаженнями, тобто немає можливості завантажити великий структурований проект тощо. Також недоліком є неможливість встановлення точок переривання виконання програми (breakpoints) для її налагодження.

Опис прототипу колаборативного середовища розробки

Компонентна архітектура прототипу колаборативного середовища розробки є досить простою та складається з трьох основних компонент (рис. 1):

- API Sphere Engine;
- серверний додаток;
- клієнтський додаток.

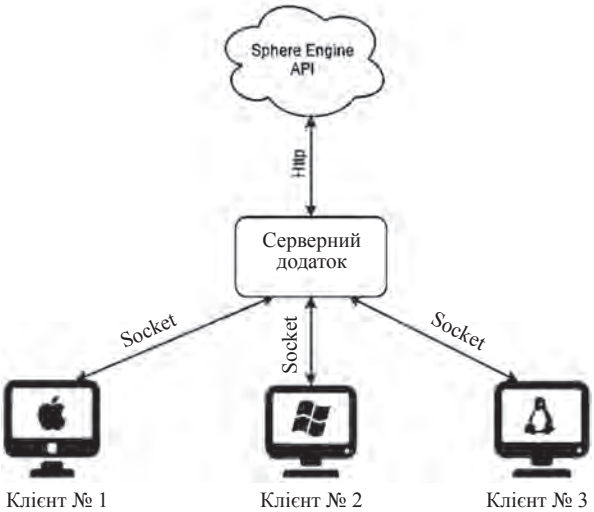

Рис. 1. Компонентна архітектура прототипу колаборативного середовища розробки

Серверний додаток реалізовано мовою JavaScript на платформі Node.js. Для підтримки можливості колаборативного редагування було використано бібліотеку з відкритим кодом OT.js,

принцип роботи якої базується на методі операційних перетворень [4; 5].

Клієнтський додаток реалізовано з використанням HTML, CSS та JavaScript. Для імплементації графічного інтерфейсу користувача використовують Bootstrap – безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-застосунків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу [6]. За допомогою спеціальних директив Bootstrap можна створити адаптивний дизайн веб-застосунку, зокрема для мобільних пристроїв. Для реалізації текстового поля редагування коду та підсвітки синтаксису використовують бібліотеку CodeMirror [7]. CodeMirror являє собою універсальний текстовий редактор, що реалізований на JavaScript для використання у браузері. Бібліотека CodeMirror призначена для редагування коду і підтримує багато мов програмування і доповнень, які реалізують розширені функціональні можливості редагування.

Обмін даними між Sphere Engine та серверним додатком здійснюється за допомогою RESTful API, обмін між клієнтами та серверним додатком – за допомогою веб-сокетів. Запити на компіляцію програмного коду від клієнтів передаються на сервер, де вони оброблюються і пересилаються до API Sphere Engine. Після отримання результату компіляції серверним додатком, вони передаються до клієнта, що ініціював процес компіляції.

Проаналізувавши методи та засоби для створення колаборативного середовища розробки, автори створили прототип колаборативного середовища розробки, який доступний у вигляді веб-застосунку за посиланням http://bit.ly/collaborative_ide. Прототип має базову функціональність для середовища розробки та містить:

- редактор коду з підсвічуванням синтаксису;
- підтримку різних мов програмування (C, C++, C#, Java, Lua, Python, Pascal);
- компіляцію коду;
- вікно виведення результату компіляції та роботи програми;
- підтримку кількох курсорів (кожен користувач має курсор випадкового кольору) для колаборативного редагування.

Після завантаження веб-сторінки необхідно ввести ім’я користувача і натиснути кнопку «Join». Після входу інтерфейс прототипу виглядає таким чином (рис. 2).

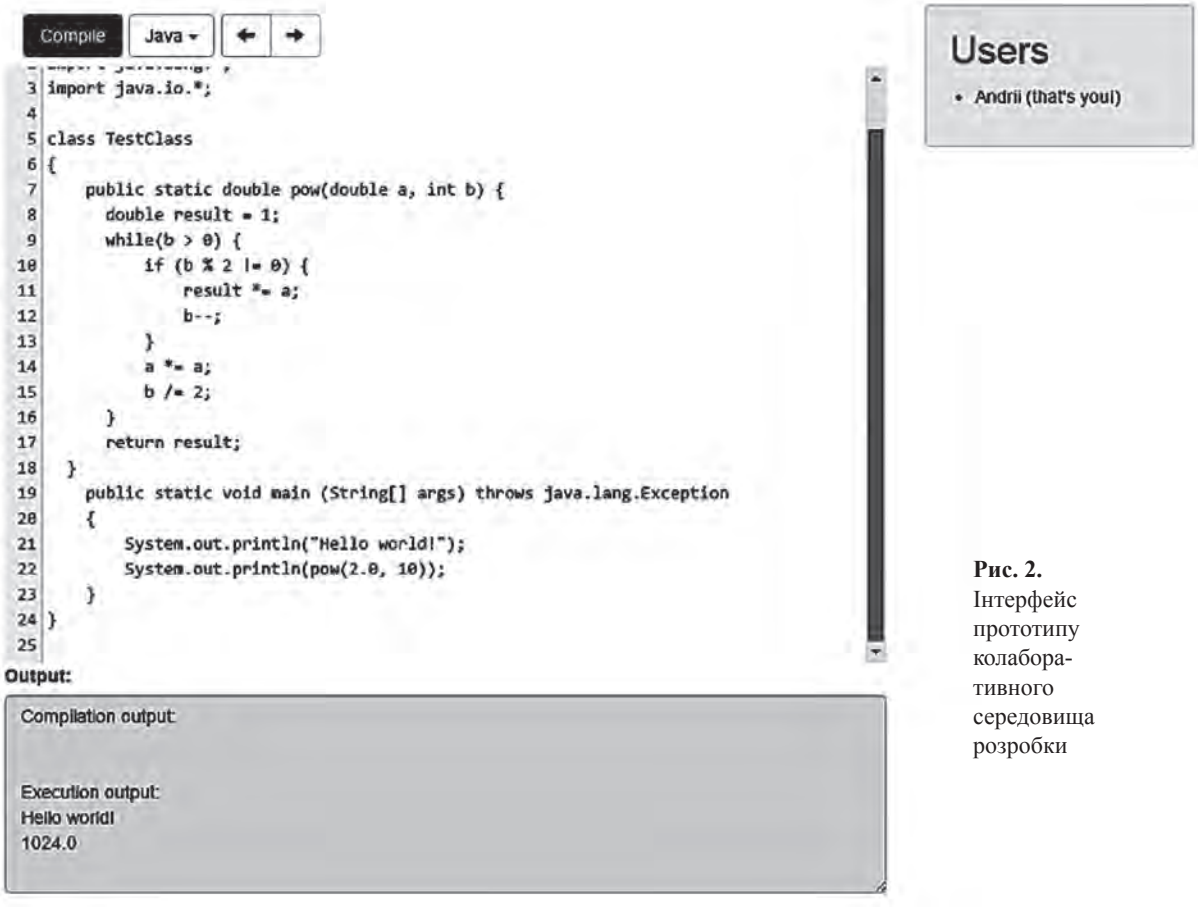

Рис. 2. Інтерфейс прототипу колаборативного середовища розробки

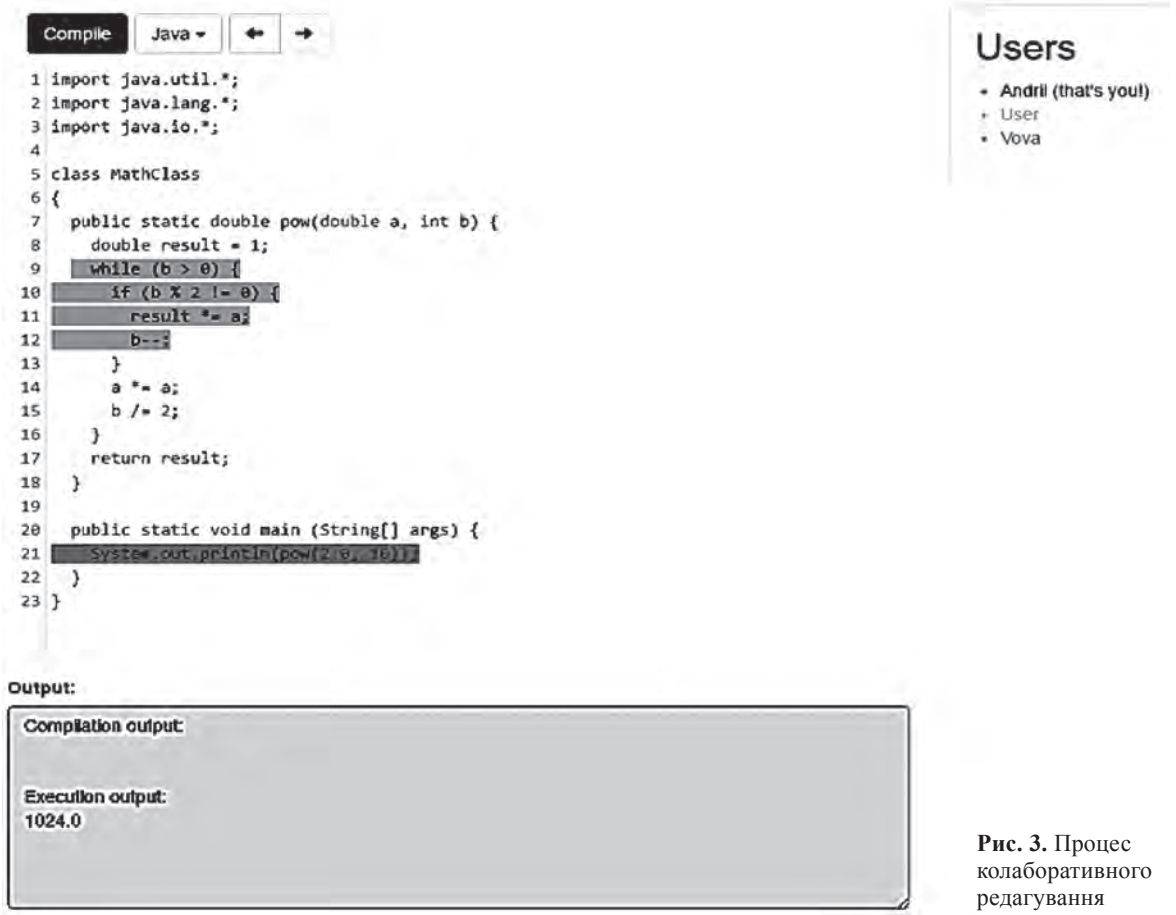

Рис. 3. Процес колаборативного редагування

Компіляція відбувається після натиснення на кнопку «Compile». Після виконання компіляції та отримання результату виконання програми від API SphereEngine у вікно «Output» виводиться результат компіляції та роботи програми.

Поточний список користувачів відображено праворуч у розділі «Users». Якщо до колаборативного середовища розробки підключено декілька користувачів, то кожен матиме випадковим чином призначений колір, який використовуватиметься для позначення курсору в редакторі коду та при виділенні тексту (рис. 3). Зміни, що були внесені кожним учасником у текстовому редакторі, буде синхронізовано для усіх учасників колаборативного редагування.

Вдалим прикладом використання подібного середовища розробки є електронне навчання програмування, адже у такому середовищі поєднується баланс необхідної функціональності та зручності використання. Зокрема, важливість створення колаборативних середовищ розробки для навчання підкреслюється повсюдним поширенням засобів електронного навчання у вищій школі, що відкриває широкі перспективи для досягнення якісно нового рівня співпраці учасників навчального процесу. Як відомо, електронна освіта розширює коло учасників, послаблює залежність від їхнього місцеперебування. Справді, з використанням колаборативного середовища розробки можна досягнути якісно нового рівня викладання і вивчення програмування. Наприклад, проведення дистанційних майстеркласів із програмування із використанням колаборативного середовища розробки дає змогу досягнути кращого сприйняття, ніж із використанням утиліт для трансляції поточного вмісту екрана. Створене колаборативне середовище розробки буде вагомим доповненням до вже наявних засобів для електронного навчання програмування.

Подальшою роботою над прототипом стане інтеграція засобів комунікації між учасниками розробки, а також створення окремих сесій редагування, доступ до яких здійснюватиметься за URL, що генеруватиметься автоматично при вході на головну сторінку веб-сервісу.

Висновки

Для повноцінної міграції з класичних інтегрованих середовищ розробки до колаборативних середовищ необхідно перенести весь функціонал, наявний у десктопних системах. Цей процес ускладюється тим, що колаборативні середовища розробки змінюють усталені соціальні аспекти організації розробки програмного забезпечення. Таким чином розвиток створеного прототипу та його впровадження стане важливим кроком у напрямі виховання професійної етики майбутніх розробників програмного забезпечення з погляду успішної колективної творчості.

Результати щорічного опитування студентів підтверджують доцільність і значущість групових навчальних проектів. Тож упровадження створених засобів до навчального процесу дасть змогу проаналізувати ефективність використання колаборативних середовищ для систем електронного навчання груповому програмуванню, зокрема на прикладі проектування веб-застосунків.

Список літератури

1. Бублик В. В. До питання електронного навчання програмуванню / В. В. Бублик // Наукові записки НаУКМА. – 2013. – Т. 151 : Комп'ютерні науки. – С. 112–115.
2. Бублик В. В. Особливості впровадження навчальної групової розробки програм / В. В. Бублик, А. О. Афонін, С. О. Борозенний // Наукові записки НаУКМА. – 2008. – Т. 86 : Комп'ютерні науки. – С. 73–77.
3. Booch G. Collaborative Development Environments [Electronic resource] / Grady Booch. – 2007. – Mode of access: http://www.drdobbs.com/architecture-and-design/collaborative-development-environments/196900222. – Title from the screen.
4. Ellis C. A. Concurrency control in groupware systems / C. A. Ellis, S. J. Gibbs // Proceedings of the 1989 ACM SIGMOD international conference on Management of data / C. A. Ellis, S. J. Gibbs. – New York, NY, USA : ACM, 1989. – P. 399–407.
5. Operational Transformation in JavaScript [Electronic resource] – Mode of access: http://operational-transformation.github.io/ot-for-javascript.html. – Title from the screen.
6. Bootstrap Examples [Electronic resource]. – Mode of access: http://getbootstrap.com/getting-started/. – Title from the screen.
7. CodeMirror [Electronic resource]. – Mode of access: http://codemirror.net/. – Title from the screen.

V. Boublik, A. Davydenko

COLLABORATIVE METHODS FOR E-LEARNING IN PROGRAMMING

The article presents a report on the authors' activities aimed to create and implement the latest software development tools in the university courses of programming through the use of close collaboration of software project participants within the framework of e-learning systems.

Cooperation between the student and the teacher is the key to success in each of the educational systems, and the use of electronic methods of training, e. g. e-learning open source system MOODLE, brings it to a new level. But traditional e-learning systems have been oriented at the individual student work. Contra verse the most sophisticated educational software projects assume teamwork which is a standard for industrial software development.

From the educational point of view, the most suitable development framework should be a collaborative environment, which provides virtual space for co-operation of all project participants. Such development environments allow many developers to edit the code simultaneously in real time. While traditional development environments emphasize improving the efficiency of each developer's collaborative development, the environment focuses on improving the efficiency of the development process in the team as a whole.

Especially popular is the collaborative development environment, the access to which is provided by the model Software as a Service, since there is no need to configure the development environment on each computer from which the process of software development is conducted.

A prototype of collaborative development environment has been created, which includes the following: the code editor with syntax highlighting; support for various programming languages (C, C ++, C #, Java, Lua, Python, Pascal); code compilation; output window for compile and work program; support for multiple cursors (each user has a random colour) for collaborative editing. The collaborative development environment will be a valuable addition to already existing e-learning tools used for education of software developers.

Keywords: e-learning, collaborative environments, programming.

Матеріал надійшов 25.09.2016