Технічні науки

УДК 004.43

Булах Богдан Вікторович

кандидат технічних наук, доцент кафедри системного проектування

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Булах Богдан Викторович

кандидат технических наук, доцент кафедры системного проектирования

Национальный технический университет Украины

"Киевский политехнический институт имени Игоря Сикорского"

Bulakh Bogdan

Candidate of Technical Sciences,

Associate Professor of the System Design Department

National Technical University of Ukraine

"Igor Sikorsky Kyiv Polytechnic Institute"

Лашко Олена Вікторівна

викладач кафедри приладів та систем неруйнівного контролю

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Лашко Елена Викторовна

преподаватель кафедры приборов и систем неразрушающего контроля

Национальный технический университет Украины

"Киевский политехнический институт имени Игоря Сикорского"

Lashko Olena

Lecturer at the Non-Destructive Testing Instruments and Systems Department

National Technical University of Ukraine

"Igor Sikorsky Kyiv Polytechnic Institute"

ВІРТУАЛЬНИЙ КАБІНЕТ ВИКЛАДАЧА ПРОГРАМУВАННЯ
ВИРТУАЛЬНЫЙ КАБИНЕТ ПРЕПОДАВАТЕЛЯ ПРОГРАММИРОВАНИЯ
VIRTUAL WORKSPACE FOR A PROGRAMMING TEACHER

Анотація. *Досліджено проблему створення інструментарію для автоматизації роботи викладача програмування на основі набору інструментів промислової програмної інженерії.*

Ключові слова: *віруальний кабінет, програмування, інструментарій.*

Аннотация. *Исследована проблема создания инструментария для автоматизации работы преподавателя программирования на основе набора инструментов промышленной программной инженерии.*

Ключевые слова: *виртуальный кабинет, программирование, инструментарий.*

Summary. *The problem of the development of the toolkit based on software engineering industrial tools for programming course teacher' work automation was investigated.*

Key words: *virtual workspace, programming, toolkit.*

Постановка задачі. Сучасна методологія викладання курсів, мета яких - навчити студентів певній мові програмування або принципам якоїсь із парадигм програмування (процедурної, об'єктно-орієнтованої тощо), передбачає наявність практичних занять. Інколи їх відносять до лабораторних, однак коректніше їх називати саме практичними заняттями, бо на них, як правило, відпрацьовуються навички студентів з написання алгоритмів та їх реалізація у вигляді кодів програм. Часто якість роботу студента на таких заняттях страждає через необхідність встигнути як виконати завдання згідно варіанту, так і захистити його одночасно зі своїми

колегами. Так виникає ідея скоротити час, необхідний на перевірку викладачем коректності виконання завдання, за рахунок автоматизації ряду дій, що можливе в рамках "віртуального робочого середовища (кабінету) викладача". Актуальність даної теми також обумовлюється все більш популярним "дистанційним" режимом навчання, який пропонують все більше закладів вищої освіти. В такому режимі наявність віртуального кабінету - необхідність.

Віртуальний кабінет викладача має як ряд спільних для усіх предметів (базових) функцій та ряд специфічних функцій, характерних лише для обмеженої групи навчальних курсів або навіть одного курсу. Такими притаманними курсам з програмування функціями є:

- перевірка коректності написання програми (контроль синтаксису),
- перевірка коректності виконання програми (відповідність результатів роботи завданню),
- перевірка коректності побудови програми: (чи вірні алгоритмічні рішення, структури даних, архітектурні шаблони тощо (назвемо їх проектними рішеннями) було обрано студентом для вирішення завдання).

В усіх цих підзадачах контролю можлива автоматизація з використанням інструментів зі стеку засобів промислової розробки програмного забезпечення. Включаючи хмарні сервіси та ресурси [1,2]. Таким чином, основна ідея даної публікації - показати можливі варіанти побудови віртуального кабінету викладача програмування з використанням (інтеграцією) існуючих спеціалізованих інструментів. Це дасть змогу суттєво скоротити витрати часу викладача на пошук помилок у студентських роботах, а студенти зможуть оперативніше отримувати зворотний зв'язок від викладача та зможуть швидше та якісніше виправити виявлені недоліки. Це все, в результаті, покликане підвищити якість викладання та закріплення матеріалу.

Складові інструменти кабінету викладача

На нашу думку, віртуальний кабінет викладача міг би спиратися на наступні спеціалізовані інструменти.

1. *Віддалений репозитарій коду*, на який студенти будуть завантажувати чергові версії своїх робіт. Хоча використання простого хмарного диску-сховища допустиме, але більш доцільно організувати репозитарії коду саме з використанням *систем контролю версій*. Це дасть змогу переглядати та контролювати усі зміни, які студенти вноситимуть у код їх програм, маючи змогу порівнювати різні версії (по хронології) їх коду. До того ж при внесенні змін студенти будуть їх коментувати, що є додатковою інформацією для викладача. І, зрештою, випадково видалити код буде неможливо. Стандартними де-факто нині є такі децентралізовані системи контролю версій, як Git та Mercurial, які підтримують роботу з кількома гілками (версіями) коду.

2. *Хмарне середовище розробки*. Може бути корисне як самим студентам (середовище доступне скрізь де є доступ до мережі, тобто можна не інсталювати програмне забезпечення у випадку розробки відносно простих програм), так і викладачу - як інструмент для зручного аналізу коду. Середовище має підтримувати розширений механізм пошуку (з використанням регулярних виразів), взаємодіяти з системами контролю версій (може виступати графічним клієнтом для цих систем) та завантажувати код з віддалених репозиторіїв, а також бажаними є інструменти для поглибленого рефакторингу та аналізу коду (для простішого з'ясування коректності обраних студентом проектних рішень). Нині є достатньо подібних рішень, інтегрованих з репозитаріями та середовищами виконання, наприклад Cloud 9, Eclipse Che.

3. *Інструмент автоматизованого збирання програм*. Має підтримувати усі необхідні компілятори та інструменти пакетної збірки програм, що використовуються в конкретному курсі. Іншими словами, має

бути достатньо розширюваним. Окрім добревідомих утиліт типу CMake варто звернути увагу на системи неперервної інтеграції типу Jenkins, які мають досить широкі можливості.

4. *Средовище тестування програм*. Як мінімум, має підтримувати автоматизоване виконання консольних програм (в ідеалі - і графічних, однак в рамках більшості курсів програмування достатньо обмежитись консольними програмами) на наборі тестових вхідних даних з визначенням відповідності отриманих результатів очікуваним. Альтернативний варіант - пропагування стилю "розробка через тестування" та розробка студентами unit-тестів. Можливе підключення сематничних аналізаторів коду [3].

5. *Система обліку помилок* (баг-трекер). Для обліку виявлених при перевірці помилок в коді та відстеження прогресу їх усунення. З можливістю визначати різні категорії та пріоритети задач.

6. *Система контролю загального прогресу* виконання заввдань студентами. Може базуватись на програмному забезпеченні управління проектами, з можливістю керування підзадачами (лабораторні роботи, курсова робота).

7. *Сервіс спілкування зі студентами* (поштовий сервіс, мессенджери). Для вирішення питань, що виходять за рамки баг-трекера, або ж для дублювання важливих подій на поштові скриньки студентів. Звичайно, бажано тримати всю історію взаємодії зі студентами в одному місці, тож даний сервіс є скоріше резервним, за наявності баг-трекера.

8. *Система віртуалізації або контейнеризації*. Для виконання зібраних кодів з метою перевірки виконання тестів. Контейнери Docker зарекомендували себе надійним та зручним інструментом в тому числі і для тестування програм, тож стануть в нагоді і для вирішення задач кабінету викладача.

Варіант організації віртуального кабінету викладача

В результаті відбору найбільш значущих інструментів, які до того ж є відкритими та безкоштовними, було запропоновано наступну будову кабінету викладача, представлену на рис.1.

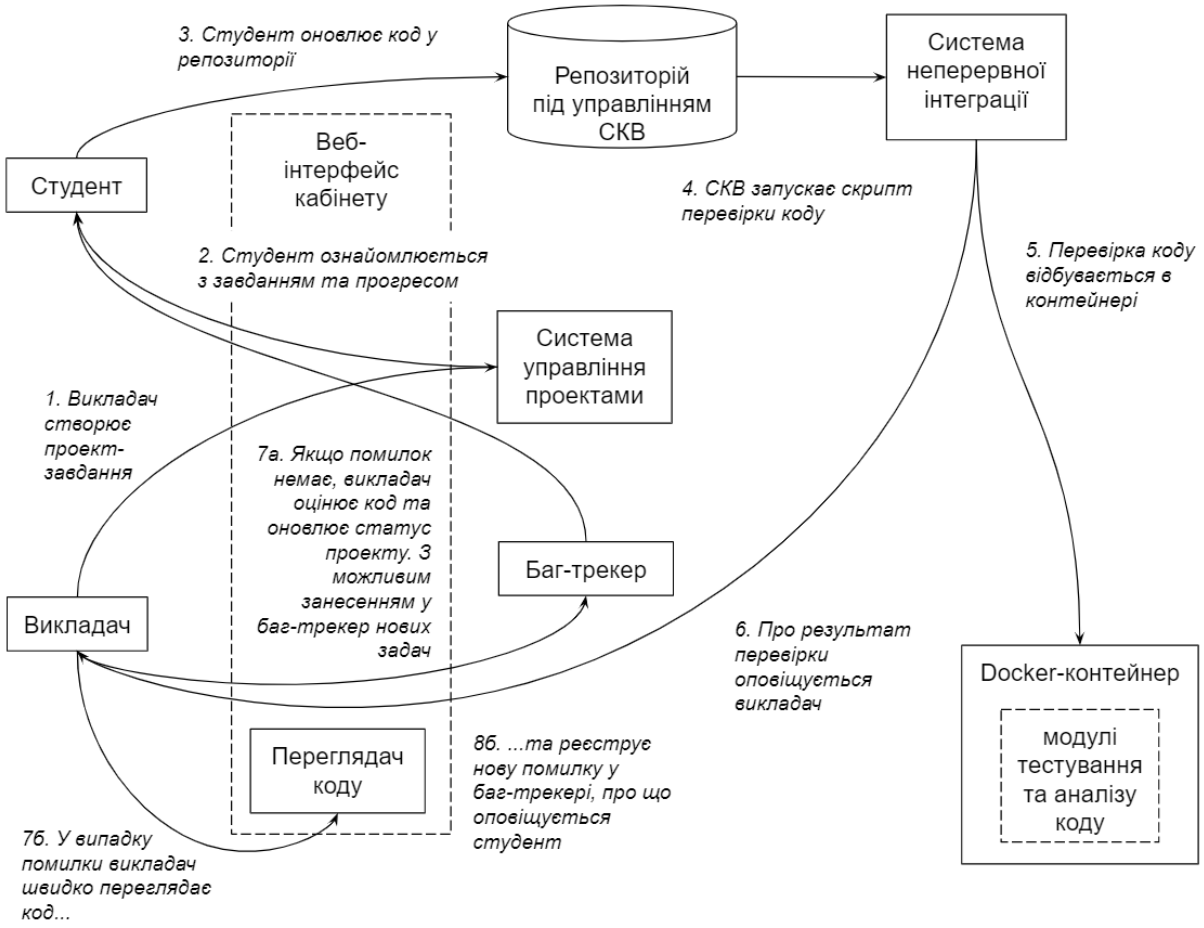

Рис. 1. Основні будівельні блоки віртуального кабінету викладача та послідовність роботи в ньому

Роль точки входу викладача та студентів відіграє веб-портал, що надає резюме по інформації інтегрованих інструментів (статус проекту, задачі на усунення помилок, перегляд коду з репозиторію). У якості системи управління проектами та баг-трекеру обрано Redmine, система контролю версій - Git, засіб неперервної інтеграції - Jenkins. Дана система знаходиться в експериментальній експлуатації, і навіть без функціонального порталу довела свою зручність при роботі з великими групами студентів. У якості

майбутнього розвитку системи будуть додані модулі перевірки кодів на плагіат та автоматизованого пошуку "антипатернів" у студентських кодах.

Висновки. В даній статті розглядається проблема автоматизації роботи викладача програмування за допомогою створення програмного комплексу "віртуальний кабінет викладача". Виділено коло функцій, які потребують автоматизації та запропоновано максимально скористатися існуючими зрілими рішеннями, поширеними у промисловості. Досвід використання прототипу подібної системи на кафедрах авторів – позитивний.

Література

1. Kurelovi K., Rako S. and Tomljanovi J. Cloud Computing in Education and Student's Needs // Information & Communication Technology Electronics & Microelectronics (36th Int. Conference MIPRO). – 2013. – PP. 856-861.
2. Wang B., and Xing H. The Application of Cloud Computing in Education Informatization // IEEE int. Conference on Computer Science and Service System (CSSS). – 2011. – PP. 2673-2676.
3. Булах Б. В. Застосування семантичних технологій для аналізу та рефакторингу програмного коду // Міжнародний науковий журнал "Інтернаука". – 2018. – №12.