

О. М. КОВТУН, О. С. ЛІННИК, К. В. ІВАНЧЕНКО

РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ДЛЯ ДИСТАНЦІЙНОГО КОНТРОЛЮ ЗНАТЬ

Метою даної дослідження було розроблення та впровадження інформаційної системи для оцінки та тестування знань студентів з метою спрощення процесу створення тестів та оцінки навчальних досягнень викладачів. У ході роботи був проведений огляд існуючих комп'ютерних навчальних програм, аналіз ринку сучасних аналогів, та на основі отриманих даних сформульовано функціональні та нефункціональні вимоги до розроблюваного програмного продукту. Виходячи з поставлених вимог, було обрано технічні та програмні засоби для розробки програми, а саме мова програмування JavaScript та фреймворк ReactJS для розробки клієнтської частини веб-застосунку, що дозволяє швидко та зручно розробляти інтерактивні користувацькі інтерфейси та гарантує стабільну роботу системи. Серверна частина застосунку реалізована за допомогою потужного та гнучкого фреймворку Spring, який дозволяє створювати масштабовані та високопродуктивні веб-додатки.

Для досягнення поставленої мети в роботі вирішуються наступні задачі: реєстрація нового користувача, редагування даних користувача, вхід та вихід користувача в/із системи, перегляд/створення/редагування/видалення навчальної групи, перегляд студентів обраної групи, перегляд/створення/редагування/видалення/публікація тесту, перегляд/створення/редагування вправи у окремому тесті, отримання всіх вправ у вибраному тесті, призначення тесту на проходження обраній групі, можливість переглянути/здати призначені користувачу (його групі) тести, можливість перевірити вправи та виставити оцінку (деякі вправи можуть бути оцінені системою автоматично, якщо викладач задав вірну відповідь для порівняння), можливість переглянути результат за зданих тест. Була спроектована та побудована база даних MySQL відповідно до описаної моделі даних яка відповідає третьому ступеню нормалізації. Розроблено структурну та функціональну схеми системи. Виділено три ролі користувачів: адміністратор, викладач, студент. Велике значення приділялося забезпеченню простоти та зручності користування продуктом користувача, оскільки це грає ключову роль у використанні програмного забезпечення. У процесі розробки програми велика увага була приділена створенню інтуїтивно зрозумілого та комфортного інтерфейсу.

Як результат, була створена інформаційна система для автоматизованого тестування, яка успішно виконує поставлені завдання і відповідає всім визначеним вимогам, а також є цілком готовою до практичного впровадження. Отримані результати можуть бути використані як для інтеграції програми у навчальний процес, так і для надання студентам можливості отримати практичні навички.

Ключові слова: контроль знань, мікросервісна архітектура, spring framework, reactjs.

The purpose of this study was to develop and implement an information system for evaluating and testing students' knowledge in order to simplify the process of creating tests and evaluating the educational achievements of teachers. In the course of the work, a review of existing computer training programs, an analysis of the market of modern analogues was carried out, and on the basis of the obtained data, functional and non-functional requirements for the developed software product were formulated. Based on the requirements, technical and software tools were chosen for the development of the program, namely the JavaScript programming language and the ReactJS framework for the development of the client part of the web application, which allows for quick and convenient development of interactive user interfaces and guarantees the stable operation of the system. The server part of the application is implemented using the powerful and flexible Spring framework, which allows you to create scalable and high-performance web applications.

To achieve the goal, the following tasks are solved in the work: registering a new user, editing user data, logging in and out of the system, viewing/creating/editing/deleting a study group, viewing students of a selected group, viewing/creating/editing/deleting/ publishing a test, viewing/creating/editing an exercise in a separate test, receiving all exercises in a selected test, assigning a test to a selected group, the ability to view/take tests assigned to the user (his group), the ability to check exercises and assign a grade (some exercises may be evaluated by the system automatically if the teacher gave the correct answer for comparison), the ability to view the result for the passed test. A MySQL database was designed and built according to the described data model that corresponds to the third degree of normalization. The structural and functional scheme of the system has been developed. There are three user roles: administrator, teacher, student. Great importance was attached to ensuring the simplicity and ease of use of the user's product, as this plays a key role in the use of the software. In the process of developing the program, great attention was paid to creating an intuitive and comfortable interface.

As a result, an information system for automated testing was created, which successfully fulfills the assigned tasks and meets all the specified requirements, and is also fully ready for practical implementation. The obtained results can be used both to integrate the program into the educational process and to provide students with the opportunity to acquire practical skills.

Keywords: knowledge control, microservice architecture, spring framework, reactjs.

Вступ.

Так як необхідність переходу на дистанційну форму навчання виникла не так давно - на ринку присутня невелика кількість програмного забезпечення для дистанційного контролю знань. І на практиці це програмне забезпечення майже не використовується через складність інтеграції, відсутність локалізації, застарілість або вартість.

Викладачі надають перевагу комбінуванню різних засобів для організації дистанційного навчання. Цей підхід має багато недоліків. Навчальні матеріали легко загубити через велику кількість різних систем, де викладачі їх публікують (Google drive, One drive, Email, Teams and Zoom чати). Також за такого підходу часто відсутній зворотній зв'язок. Студенти нерідко не знають оцінок за виконані роботи, важко відслідкувати загальний прогрес по конкретній дисципліні.

Тому задля вирішення наведених проблем

виникає потреба в запровадженні уніфікованого програмного забезпечення для контролю знань студентів. Для цього можна розглянути доступні програми на ринку. Або ж розробити власний цифровий продукт який би задовольняв усі потреби, що є гарною ідеєю з огляду на те, що Україна займає 8 місце у світі з ІТ навичок (за рейтингом Global Skill Report 2022 від Coursera).

Аналіз останніх досліджень і публікацій. При виборі існуючої або розробці нової цифрової системи для дистанційного контролю знань треба проаналізувати та виділити основну функціональність, яку має підтримувати сервіс [1]. Наступний перелік функціональності мав би покрити усі потреби дистанційного навчання:

- здача завдань;
- дискусійні форуми;
- завантаження файлів;

- оцінювання;
- обмін повідомленнями;
- календар подій;
- новини та анонси подій (для різних рівнів: сайт, курс, навчальна група);
- онлайн тестування;
- вікі.

Отже розглянемо існуючі цифрові продукти, які б могли забезпечити усі або більшість з наведених функціональних потреб.

Програма ATutor (читається "Ейтьютор") розробляється та підтримується з 2001 року Greg Gay, Joel Kronenberg і Heidi Hazelton із Adaptive Technology Resource Centre, University of Toronto. Система ATutor поширюється на основі GNU General Public License (GPL), яка, зокрема, дозволяє вільно використовувати та змінювати програму [2].

Dokeos – це відкрита система дистанційного навчання, що відповідає міжнародним стандартам систем дистанційного навчання SCORM та AICC. Розповсюджується за принципом підписки. Вартість обговорюється з компанією Dokeos та може змінюватися в залежності від функціоналу, кількості користувачів та інше. Компаніє надає можливість безкоштовно спробувати продукт за допомогою trial підписки [3].

Moodle — навчальна платформа, призначена для об'єднання педагогів, адміністраторів і учнів (студентів) в одну надійну, безпечну та інтегровану систему для створення персоналізованого навчального середовища. Це безкоштовна, відкрита (Open Source) система управління навчанням. Вона реалізує філософію «педагогіки соціального конструктивізму» та орієнтована насамперед на організацію взаємодії між викладачем та учнями, хоча підходить і для організації традиційних дистанційних курсів, а також підтримки очного навчання. Перекладена на десятки мов, в тому числі і на українську [4].

Усі ці три екземпляри є якісним програмним забезпеченням з широкою функціональністю та багатою історією. Але попри всі позитивні якості, наведене програмне забезпечення рідко використовується у ВНЗ України.

Якщо ж розглядати створення програмного забезпечення з нуля, то перш за все необхідно виділити недоліки та переваги наведених вище екземплярів. Серед переваг можна вважати вже перераховану широку функціональність програм. Серед недоліків - усі екземпляри мають монолітну архітектуру та переважно не підтримуються на мобільних пристроях, часто потребують інсталяції додаткових пакетів на комп'ютер користувача.

Тому новий застосунок має перейти вже перевірені часом можливості та функціонал для користувачів і в той же час позбутися обмежень які є результатом монолітного архітектурного підходу.

Сучасним трендом у розробці програмного забезпечення є мікросервісна архітектура [5, 6]. Хоч цей підхід і має деякі недоліки у відношенні до

монолітної (такі як збільшення затримок через міжсервісну взаємодію), все ж він має велику перевагу - можливість горизонтально масштабування застосунку.

З впровадженням цього підходу більше не буде складної зв'язаності між логікою програми та користувацьким інтерфейсом. Ці аспекти стануть окремими програмними сутностями (backend та frontend сервісами). Завдяки цьому можна буде вирішити другу наведено проблему - підтримка на мобільних пристроях. Коли логіка та розрахунки програми реалізовані окремими сервісом - ми маємо змогу розробити будь яку кількість користувацьких інтерфейсів на будь яких платформах (web app, desktop app, android app, iOS app і тд.).

Реалізація застосунку - загальна архітектура.

Для розробки мікросервісного застосунку побудовано інфраструктуру з трьох шарів (рис.1):

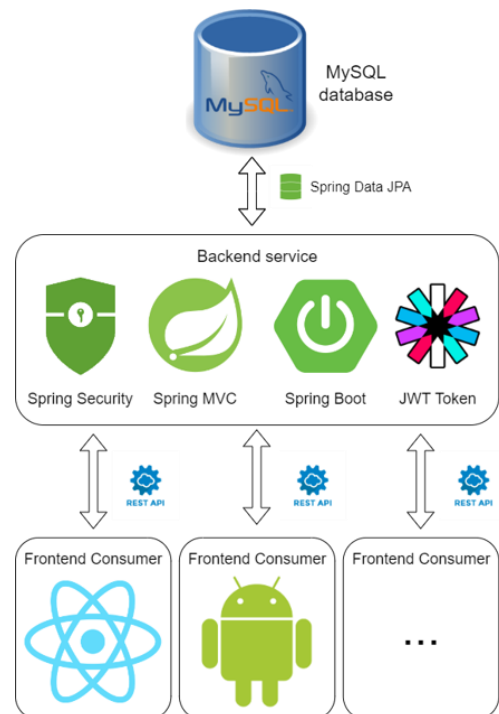


Рис. 1 - Діаграма архітектури застосунку та використаних технологій

Рівень збереження даних - одна або кластер баз даних з визначеною предметною моделлю. Для роботи використано MySQL database.

Бекенд сервіс - stateless, headless застосунок. Проводить проміжкові операції з даними перед передачею їх на фронтенд клієнти або збереженням у базі даних. У роботі використано мову Java та фреймворк Spring (Boot, MVC, Security). Для аутентифікації користувачів використовується JWT Token.

Фронтенд клієнти - сукупність застосунків, яка використовується кінцевими користувачами для взаємодії з системою. В рамках роботи розроблено веб-застосунок з використанням JavaScript та ReactJS фреймворку.

Міжрівнева взаємодія.

Бекенд взаємодіє за базою даних за допомогою бібліотеки Spring Data JPA.

Фронтенд взаємодіє з бекендом через REST API ендпоінти. Формат обміну даними - JSON.

Реалізація застосунку - серверна частина.

Структуру backend застосунку умовно можна поділити на п'ять логічних шарів (рис.2).

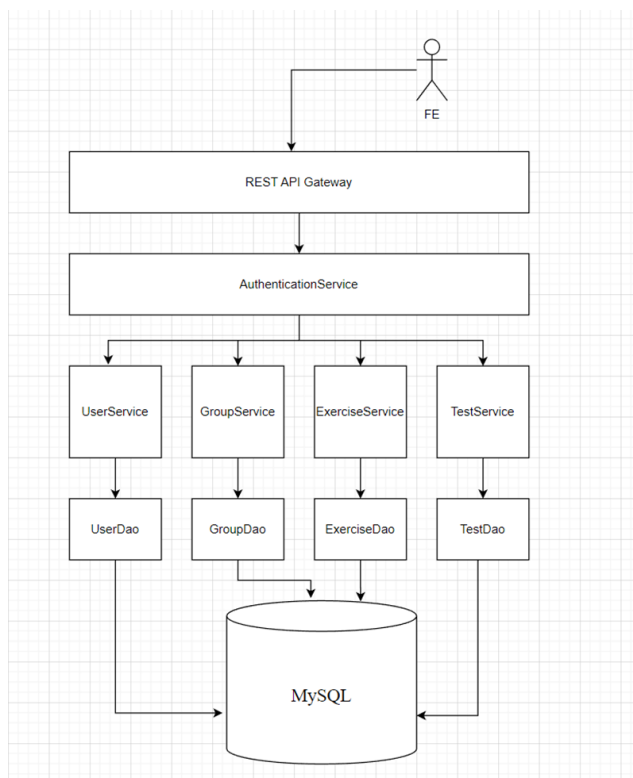


Рис. 2 - Високорівнева діаграма структури backend сервісу

Найглибшим шаром є база даних [7, 8]. До бази даних звертається шар DAO, який реалізовано за допомогою JPA (Java Persistence API). Далі розроблено шар сервісів - у яких реалізована основна програмна логіка з використанням методів DAO шару.

Далі реалізовані шари аутентифікації і REST API Gateway. Аутентифікація розроблена за допомогою модулю Spring Security та JWT токенів. Цей шар відповідальний за перевірку ролі користувача та дозволу або заборони виклику окремих ендпоінтів. В свою чергу REST API Gateway шар реалізовано з використанням модулю Spring MVC. Цей шар відповідає за створення REST ендпоінтів та їх мапінг на відповідні HTTP URLs.

В рамках проектної роботи було реалізовано контролери та сервіси для підтримки наступних дій:

- реєстрація нового користувача;
- редагування даних користувача;
- вхід та вихід користувача в/із системи;
- перегляд/створення/редагування/видалення навчальної групи;

- перегляд студентів обраної групи;
- перегляд/створення/редагування/видалення/публікація тесту;
- перегляд/створення/редагування вправи у окремому тесті;
- отримати всі вправи у вибраному тесті;
- призначити тест на проходження обраній групі;
- переглянути/здати призначені користувачу (його групі) тести;
- перевірити вправи та виставити оцінку (деякі вправи можуть бути оцінені системою автоматично, якщо викладач задав вірну відповідь для порівняння);
- переглянути результат за зданий тест.

Реалізація застосунку - користувацький інтерфейс (веб).

Задля реалізації веб інтерфейсу було використано ReactJS фреймворк. React – це ефективна і гнучка декларативна бібліотека JavaScript для збірки UI від команди Facebook [9]. Вона дозволяє без зусиль створювати інтерактивний інтерфейс користувача. У React використовується компонентний підхід. Бібліотека немає контролерів, в'ю, моделей, шаблонів – все є компонент. Компоненти можна перевикористовувати, успадковувати один від одного, об'єднувати. Компонент – це свого роду будівельна одиниця, з якої збирається інтерфейс. Перевагою є можливість порівняння React'ом віртуального DOM'а з реальним, і виконання мінімальних змін для їх синхронізації. Віртуальний DOM вирішує проблему з обробкою подій в різних браузерах, за рахунок цього React надає сумісну модель подій в будь-якому браузері. React дозволяє використовувати будь-який інструмент при розробці, він добре поєднується з іншими фреймворками. Також дану бібліотеку дуже просто освоїти. Дизайн React поєднується з асинхронними серверними архітектурами для адаптації до майбутніх технологій. Таким чином, React вартовикористовувати, якщо необхідно створити якісний і в найкоротші терміни швидкий, легкий, зручний односторінковий додаток.

Під час розробки веб-застосунку було поставлено наступний список цільової функціональності, яка має бути реалізована:

- авторизація користувачів;
- розділення функціоналу за ролями користувачів;
- підтримка локалізації (українська та англійська мови);
- менеджмент користувачів (для адміністраторів);
- менеджмент учбових груп (для адміністраторів);
- менеджмент тестів і завдань (для викладачів);

- перегляд і перевірка результатів тестів (для викладачів);
- перегляд і проходження призначених тестів (для учнів);
- перегляд результатів пройдених тестів (для учнів).

Програмну структуру проекту можна відобразити наступним способом (рис.3):

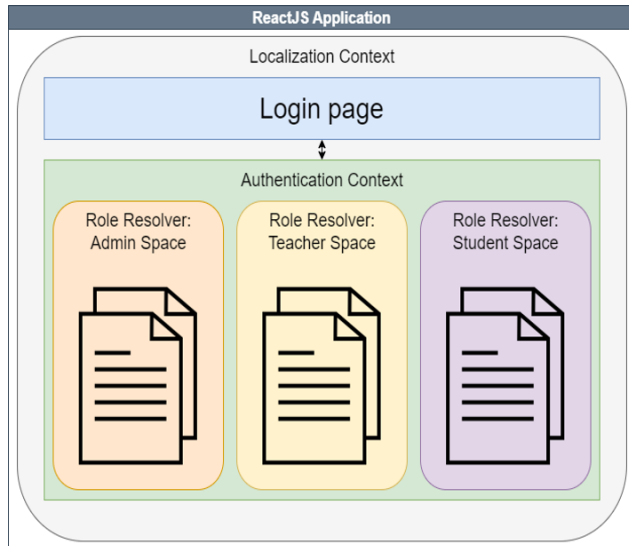


Рис. 3 - Умовна структура розробленого ReactJS сервісу

З наведеного рисунку можна описати наступні логічні шари.

1 Localization Context - глобальний шар, який надає змогу з будь якої точки програмного коду отримати переклади. Реалізовано з використанням react-localization бібліотеки.

2 Authentication Context та Login - шар відповідає за безпеку сервісу. До функціональної частини сервісу допускає тільки авторизованих користувачів, в іншому випадку перенаправляє на сторінку авторизації. Реалізовано за допомогою react-router-dom бібліотеки.

3 Role Resolver - самостійно реалізований шар для обмеження та надання доступу до функціоналу в залежності від ролі користувача (запроваджено 3 ролі: адміністратор, викладач та учень). Користувачі з різними ролями мають різний інтерфейс та можливості.

На рис.4-6 наведений вигляд інтерфейсу для відповідно адміністратора, викладача та учня.



Рис. 4 – Сторінка менеджменту користувачів

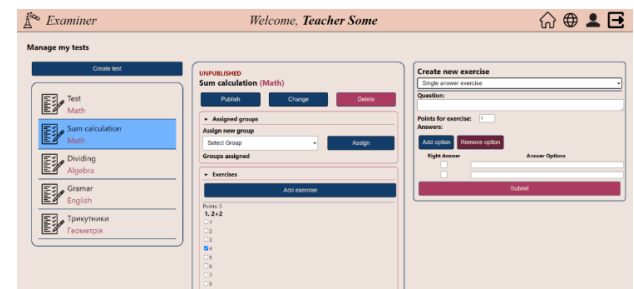


Рис. 5 – Сторінка менеджменту тестів з розгорнутим редактором тесту та формою створення нової вправи

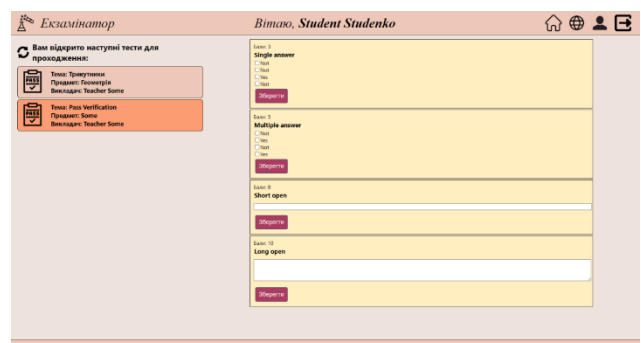


Рис. 6 – Сторінка проходження тестування учнем

Висновки. На даному етапі у сфері освіти немає уніфікованого підходу та програмного забезпечення для дистанційного контролю знань. Це викликає деякі незручності у студентів через різноманітність підходів оцінювання у викладачів.

Впровадження уніфікованого підходу з використанням спеціалізованого програмного забезпечення значно покращило би прозорість навчальних процесів. У якості такого програмного забезпечення можна обрати або існуючі екземпляри, тим паче у переліку є гарні варіанти з відкритим кодом і можливістю додавання власного функціоналу. Або, зважаючи на великий досвід розробки в Україні, створити власний сервіс перейнявши найкращі практики та використавши сучасні підходи і технології проєктування.

Список літератури

1. Пазушко М.А. Математичне та програмне забезпечення програмного додатку автоматизації процесу тестування з використанням фреймворку Spring /М.А.Пазушко, К.В.Іванченко, Д.А.Іванченко// Вісник Національного технічного університету «ХПІ». Серія: Динаміка і міцність машин. - №2. – 2022. – с. 10-16.
2. Щкодзінський О.К. Розробка навчальних курсів у системі ATutor / О.К.Щкодзінський, С.О.Войт, М.М.Луцків // Тернопіль, 2017. – 77с.
3. The Dokeos e-learning project management guide / dokeos e-learning. Dokeos, 2007. – 11с.
4. Бодненко Д.М. Тестовий контроль знань студентів у системі Moodle. / Д.М. Бодненко, Л.О.Варченко, О.Б. Жильцов// Навчально-методичний посібник для користувачів системи дистанційного навчання Moodle. К.: Київ. ун-т ім. Б. Грінченка. - 2012 р. - 112 с.
5. Приходченко С.Д. Обґрунтування вибору мікросервісної архітектури в порівнянні з монолітною.: стаття / С.Д. Приходченко, К.С. Родна, Р.В. Поштак // НТУ «Дніпровська політехніка». Дніпро. – 4с.
6. Кріс Річардсон. Мікросервіси. Патерни розробки і рефакторінга / Кріс Річардсон – 2019. – 544с.
7. REST API Developer Guide / Salesforcedocs. 2022. – 346с.
8. Hatma Suryotrisongko, Dedy Puji Jayanto, Aris Tjahyanto. Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot. / 4th Information Systems International Conference 2017, ISICO 2017, Indonesia.
9. React and Libraries. Your Complete Guide to the React Ecosystem. Elad Elrom / Montvale, NJ, USA. 2020. – 418с.

References (transliterated)

1. Pazushko M.A. Matematychnе ta prohramne zabezpechennia prohramnoho dodatku avtomatyzatsii protsesu testuvannia z vykorystanniam freimvorku Spring /M.A.Pazushko, K.V.Ivanchenko, D.A.Ivanchenko// Visnyk Natsionalnoho tekhnichnoho universytetu «KhPI». Seria: Dynamika i mitsnist mashyn. - №2. – 2022. – s. 10-16.
2. Shchkodzynskiy O.K. Rozrobka navchalnykh kursiv u systemi ATutor / O.K.Shchkodzynskiy, S.O.Voit, M.M.Lutskiv // Ternopil, 2017. – 77s.
3. The Dokeos e-learning project management guide / dokeos e-learning. Dokeos, 2007. – 11с.
4. Bodnenko D.M. Testovyi kontrol znan studentiv u systemi Moodle. / D.M. Bodnenko, L.O.Varchenko, O.B. Zhyltsov// Navchalno-metodychni posibnyk dlia korystuvachiv systemy dystantsiinoho navchannia Moodle. K.: Kyiv. un-t im. B. Hrinchenka. - 2012 r. - 112 s.
5. Prykhodchenko S.D. Obgruntuvannia vyboru mikroservisnoi arkhitektury v porivnanni z monolitnoiu.: stattia / S.D. Prykhodchenko, K.S. Rodna, R.V. Poshtak // NTU «Dniprovskа politekhnika». Dnipro. – 4s.
6. Kris Richardson. Mikroservisy. Paterny rozrobky i refaktorinha / Kris Richardson – 2019. – 544s.
7. REST API Developer Guide / Salesforcedocs. 2022. – 346с.
8. Hatma Suryotrisongko, Dedy Puji Jayanto, Aris Tjahyanto. Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot. / 4th Information Systems International Conference 2017, ISICO 2017, Indonesia.
9. React and Libraries. Your Complete Guide to the React Ecosystem. Elad Elrom / Montvale, NJ, USA. 2020. – 418с.

Надійшла (received) 13.12.2023

Відомості про авторів / About the Authors

Ковтун Олександр Миколайович (Kovtun Oleksandr) - Національний технічний університет «Харківський політехнічний інститут», магістр комп'ютерних наук; тел: (067)127-32-50; e-mail: kovtunsanter@gmail.com.

Лінник Олексій Сергійович (Linyuk Oleksii) – Національний технічний університет «Харківський політехнічний інститут», магістр комп'ютерних наук; тел: (096)701-66-97; e-mail: virich1997@gmail.com.

Іванченко Ксенія Вікторівна (Ivanchenko Kseniia) – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри комп'ютерного моделювання процесів та систем; тел.: (097) 437-89-74; e-mail: xeniya.ivanchenko@gmail.com, ORCID: <https://orcid.org/0000-0003-0960-1200>