

УДК 004.65

М.А. БОРОДІН, В.О. МЕТЕЛЬОВ

DOI: 10.20998/2078-9130.2021.2.249801

РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ПОШИРЕННЯ НАВЧАЛЬНОЇ ІНФОРМАЦІЇ

Розглядається задача поширення навчальної інформації та дослідження взаємодії компонентів на розподілення навантаження від користувачів. Мета роботи: реалізувати програмне забезпечення для розв'язання вказаної задачі. Спроекувати базу даних, та структуру проекту. В програмі реалізувати різні права доступу до інформації. Для програмного забезпечення має бути створений, зрозумілий для користувача, інтерфейс. Розробити тести для програми, які перевіряють функціональність та продуктивність програми. За допомогою розроблених тестів провести дослідження на розподілення навантаження від користувачів. Розроблено програму, яка складається з двох частин (серверної та клієнтської), які взаємодіють за допомогою HTTP запитів. Серверна частина опрацьовує дані та проводить їх перевірку. Клієнтська частина отримує та надає дані серверній частині та реалізовує зрозумілий для користувача інтерфейс. Рішення поставленої задачі реалізовано в середовищах IntelliJ Idea 2020.1 (Ultimate Edition) та WebStorm 2020.1. Для реалізації використовувались мови програмування Java 11 і JavaScript (ECMAScript 6). Новизна роботи полягає у створенні програми, яка водночас дає можливість використовувати функції, необхідні для навчання у вищому навчальному закладі. Наприклад, надання студентам доступу до навчальних матеріалів, можливість для всіх користувачів переглядати свій розклад. Викладачі також можуть переглянути розклад груп, які вони викладають у поточному семестрі. Кожен учень може переглянути розклад учителя, якщо він викладає хоча б один предмет. Іншою важливою функцією є управління структурою університету (керування інформацією про інститути, кафедри і навчальні групи). Було також проведено дослідження продуктивності програми, та досліджено можливості покращення швидкості програми у разі значного збільшення активних користувачів.

Ключові слова: поширення матеріалів, розмежування доступу, авторизація, управління, студент, викладач, адміністратор, розклад.

Рассматривается задача распространения обучающей информации и исследования взаимодействия компонентов на распределение нагрузки от пользователей. Цель работы: реализовать программное обеспечение для решения данной задачи. Спроектировать базу данных и структуру проекта. В программе реализовать разные права доступа к информации. Для программного обеспечения должен быть создан пользовательский интерфейс. Разработать тесты для программы, проверяющие функциональность и производительность программы. С помощью разработанных тестов провести исследования по распределению нагрузки от пользователей. Разработана программа, состоящая из двух частей (серверной и клиентской), взаимодействующих с помощью HTTP запросов. Серверная часть обрабатывает данные и производит их проверку. Клиентская часть получает и предоставляет данные серверной части и реализует понятный пользовательский интерфейс. Решение поставленной задачи реализовано в средах IntelliJ Idea 2020.1 (Ultimate Edition) и WebStorm 2020.1. Для реализации использовались языки программирования Java 11 и JavaScript (ECMAScript 6). Новизна работы заключается в создании программы, которая дает возможность использовать функции, необходимые для обучения в высшем учебном заведении. Например, предоставление студентам доступа к учебным материалам, возможность для всех пользователей просматривать свое расписание. Преподаватели могут также просмотреть расписание групп, которые они преподают в текущем семестре. Каждый учащийся может пересмотреть расписание преподавателя, если он преподает хотя бы один предмет. Другой важной функцией является управление структурой университета (управление информацией об институтах, кафедрах и учебных группах). Было также проведено исследование производительности программы и исследованы возможности улучшения скорости программы в случае значительного увеличения активных пользователей.

Ключевые слова: распространение материалов, разграничение доступа, авторизация, управление, студент, преподаватель, администратор, расписание.

The problem of sharing of educational information and research of components interaction on load distribution from users is considered. Purpose: to implement software to solve this problem. Design a database and project structure. Role-based access to information should be implemented. A user-friendly interface must be created for the software. Develop tests for the program that should be implemented to check the functionality and performance of the program. Use the developed tests to conduct research on load distribution from users. The developed program consists two parts (server and client). They interact using HTTP requests. The server part processes the data and verifies them. The client part receives and provides data to the server part and implements a user-friendly interface. The solution of this problem is implemented in the environments IntelliJ Idea 2020.1 (Ultimate Edition) and WebStorm 2020.1. Java 11 and JavaScript (ECMAScript 6) programming languages were used for implementation. The novelty of the work lies in the development of a program that makes it possible to use the functions necessary for studying at a higher educational institution. For example, providing students with access to study materials, the ability for all users to view their schedule. Instructors can also view the schedule of the groups they teach in the current semester. Each student can revise the teacher's schedule if he teaches at least one subject. Another important function is the management of the structure of the university (management of information about institutes, departments and study groups). We also conducted research on the performance of the program and investigated the possibility of improving the speed of the program in the event of a significant increase in active users.

Keywords: distribution of materials, access restriction, authorization, management, student, teacher, administrator, schedule.

Вступ. Протягом швидкого розвитку технологій освіта також має швидко розвиватися. Однією з найважливіших частин навчального процесу є розповсюдження навчальних матеріалів та всієї інформації, необхідної для навчання студентів, наприклад розкладу чи інформації про учасників навчального процесу. Також у наш час дуже важливі безпека та швидкий доступ до інформації, тому програма створена для того, щоб сторонні особи не мали доступу до інформації зареєстрованих користувачів, а самі користувачі могли отримати її з будь-якої точки світу [1].

Розроблена програма допомагає викладачам швидко надати доступ до всіх навчальних матеріалів для учнів. При цьому забезпечується конфіденційність

даних, сторонні особи не можуть отримати доступ до матеріалів. Кожен, хто використовує розроблену програму, має власний обліковий запис з логіном і паролем, що надає тільки йому доступ до інформації.

Ще однією важливою функцією розробленої програми є можливість перегляду розкладу. Кожен користувач, як студент, так і викладач, може переглядати свій розклад з будь-якого місця, де є доступ до Інтернету.

Для студентів розроблена можливість переглядати інформацію з кабінету студента. Так як кожен студент має свій акаунт і в кабінеті студента, перед переглядом потрібно авторизуватись і там. Дані студентів з кабінету студента не зберігаються в

розробленій програмі, вони зберігаються тільки в браузері, щоб студенти могли зручно їх переглядати.

Використовуючи розроблену програму та тести продуктивності для неї було проведено дослідження роботи програми під час великого навантаження.

Постановка задачі. У даній роботі реалізовано програму, яка дає можливість надавати доступ до навчальних матеріалів, можливість переглядати розклад та інформацію з кабінету студента.

Кожен користувач системи має свою особисту сторінку та можливість авторизуватися використовуючи логін і пароль або такі сервери як Microsoft Office, Gmail, та інші. Для використання функціоналу системи та отримання будь-якої інформації потрібно бути авторизованим.

Вчителі повинні надавати лише доступ до власних навчальних матеріалів, вони не повинні жодним чином впливати на навчальні матеріали інших вчителів. Студенти повинні мати можливість переглядати лише ті файли, до яких їм надано доступ. Студенти не повинні мати доступ до сторінок для завантаження нових матеріалів та розповсюдження матеріалів.

Повинен бути реалізований обліковий запис адміністратора, який має можливість керувати структурою університету (додавати інститути, кафедри та групи). Також адміністратор має можливість додавати студентів до навчальних груп і видаляти їх із груп. Однією з найважливіших його функцій є управління розкладом. В викладачів є можливість змінювати тільки власний розклад.

Програма повинна безперебійно працювати і під великим навантаженням. Щоб перевірити поведінку програми під навантаженням мають бути розроблені тести продуктивності. Найбільше навантаження на програму розраховується на перші дні навчального року, тому для розробки був обраний сценарій реєстрації студентів на початку семестру.

Інформаційні системи для вищих навчальних закладів. Інформаційні технології використовуються у всіх галузях, не виключення і галузь освіти. Впровадження інформаційної системи в академічне середовище, таке як реєстрація нового студента, профіль учасника навчального процесу або система оцінювання може покращити якість академічної роботи в середній школі чи університетах.

Atisa Dipamkara – це одна з шкіл Індонезії, яка почала використовувати інформаційні технології для підвищення ефективності та продуктивності управління освітою. Але школа зіштовхнулася з проблемою вимірювання ефективності та результативності введеної системи, визначення задоволення вчителя від використання нової системи оцінювання. Для вирішення цієї проблеми вони обрали анкети, які використовуються для оцінки зручності веб-сайтів опираючись на корисність, задоволеність та простоту використання [2].

В Бурайміському університеті було розроблено мобільний додаток для полегшення спілкування між університетом, його працівниками та студентами. Цей

додаток надає користувачам інформацію про події та новини [3].

Вплив інформаційно-комунікаційні технологій (ІКТ) за дуже короткий час став дуже помітним у всіх аспектах життя особливо в секторі освіти. Для багатьох країн набуття навичок та розуміння понять ІКТ стає необхідним для вдосконалення своєї освітньої системи. Нещодавно було відзначено що ІКТ почали широко відігравати ключову роль в освіті для полегшення процесу викладання. Тому його слід використовувати для збільшення ефективності педагогічного процесу. Більше того, ІКТ використовуються для різних цілей, таких як створення розкладу, управління електронною бібліотекою, керування звітністю та моніторинг відвідуваності [4]. Ці інструменти полегшують адміністрування університетів. У той же час ІКТ мають великий вплив на сфери досліджень та наукової діяльності [5].

Опис логічної структури. Програма для поширення навчальної інформації має клієнт-серверну архітектуру. Така архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного [6-15]. Серверів як і клієнтів може бути багато. Якщо є можливість запустити декілька серверів, то має бути реалізована не тільки функціональність програми, а і сервіс для розподілення навантаження на сервери, який спрямовуватиме запити клієнтів до найменш зайнятих серверів.

Серверна частина програми містить в собі реалізацію функціоналу програми. Клієнтська частина містить програмний код для реалізації інтерфейсу, який дає можливість зручно користуватись функціоналом системи.

В програмі реалізована авторизація користувачів, є можливість реєстрації нових користувачів, як студентів так і викладачів.

Студентам доступна сторінка користувача на якій є основна інформація про студента та розклад на день, сторінка з навчальними матеріалами, згрупованими за предметом та типом (лекції чи завдання) та сторінки з інформацією з кабінету студента.

Викладачам доступна можливість надавати доступ до навчальних матеріалів студентам, структуровано зберігати навчальні матеріали та керувати своїм розкладом.

Також є акаунт для адміністратора, в якого є можливість управління групами, структурою університету та розкладом викладачів та студентів.

Як видно зі схеми зображеної на рис. 1 клієнт з сервером спілкуються за допомогою HTTP запитів (це реалізовано в сервері за допомогою бібліотеки Spring), сервер звертається до БД використовуючи SQL скрипти.

Спроектвана база даних відповідає третій нормальній формі. Так як:

- всі атрибути в таблицях – прості;
- в таблицях відсутні повторення рядків;

– не ключові атрибути неприводимо залежать від первинного ключа (в складі потенційного ключа нема меншої підмножини атрибутів, від якої можна вивести дану функціональну залежність);

– не ключові атрибути нетранзитивно залежать від первинного ключа, всі не ключові поля, вміст яких може відноситись до декількох записів в таблиці винесено в окремі таблиці.

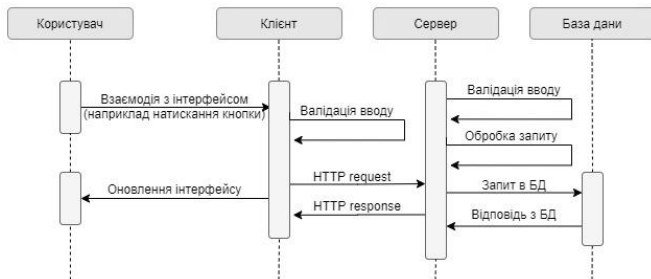


Рис. 1 – Взаємодія програми з користувачем

Серверна частина програми відповідає за управління всіх процесів, в цій частині програми проводиться перевірка даних, створення записів в базу даних, надання потрібної інформації клієнтській частині програми за допомогою HTTP запитів.

Діаграма зображена на рис. 2 показує компоненти програми та її взаємодію з іншими сервісами.

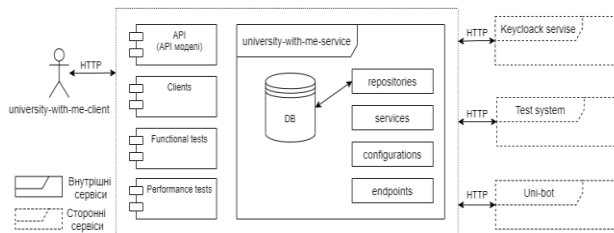


Рис. 2 – Архітектура програми

Компонент з логікою обробки даних можна розділити на 5 головних модулів для авторизації, роботи з файлами, керування розкладом, управління структурою університету та робота з кабінетом студента.

Тестування продуктивності програми. Під час тестування продуктивності важливо перевіряти сценарії не тільки використання системи в звичайному режимі, а в піковому навантаженні. В випадку програм, які використовуються під час навчального процесу це початок навчального року. Адже в цей час студенти першого курсу реєструються в системі і в систему надходить велика кількість запитів на збереженні інформації і паралельно з цим всі дивляться свій новий розклад. Тому був реалізований тестовий сценарій в якому одночасно реєструється велика кількість студентів.

Даний сценарій перевіряє одночасно роботу під великим навантаженням і сервісу авторизації і сервісу електронного навчання.

В крокові 1 виконується тільки звернення до сервісу авторизації так як тільки авторизовані в спільному сервісі авторизації користувачі можуть

продовжити реєстрації в системі електронного навчання. Це потрібно для можливості використовувати один акаунт для всіх систем (електронного навчання, тестування, можливості створення телеграм-ботів).

В кроках 2, 3, 4, 5 відбувається завантаження інформації про університети, інститути, кафедри і групи, щоб обрати свою і зареєструватись в системі. Якщо студент обрав не вірну групу, то він сам зможе змінити це в налаштуваннях, або адміністратор може перемістити його в потрібну групу.

В крокові 6 відбувається реєстрація студента в розробленій програмі з обраними університетом, інститутом, кафедрою та групою. Під час реєстрації програма звертається до сервісу авторизації щоб отримати дані про студента та повернути цю інформації для відображення домашньої сторінки.

В крокові 7 відбувається видалення акаунту з системи електронного навчання, цей крок потрібний для очистки даних після тесту.

Для тестування продуктивності програми розроблений сценарій був запущений на різній кількості користувачів та з різними проміжками часу. На одному, ста користувачах одночасно та протягом трьох хвилин, під час яких кількість користувачів плавно збільшувалась, досягала піку та через деякий час починала зменшуватись.

Зі звіту видно що всі запити були виконані успішно та займають в середньому 310 мс.

На рис. 3 зображено результат сценарію в якому намагається одночасно зареєструватись 100 користувачів. Всі запити, так само, як і в першому сценарії, виконані успішно, але з такою кількістю користувачів займають в середньому 1788мс. Найдовшим був запит в сервіс авторизації (крок 1), він зайняв 9792мс. Найкоротшим в сервіс електронного навчання (крок 3, діставання інформації про інститути), він зайняв 147 мс.



Рис. 3 – Звіт сценарію з 100 користувачів

Опис роботи програми. Кожен користувач має акаунт, тому сторонні особи не мають можливості бачити його розклад та навчальні матеріали до яких він має доступ. Для доступу до навчальних матеріалів

користувач має зареєструватися або увійти в систему, якщо вже має акаунт. На рис. 4 зображена форма реєстрації в сервісі авторизації.

Рис. 4 – Форма для реєстрації

Для зручної навігації по програмі було розроблено бокове меню, яке можна або приховати з екрану або відкрити. Меню відрізняється для різних типів користувачів.

Адміністратору доступні такі функції як управління структурою університету (створення інститутів, кафедр, груп). Адміністратор має можливість переглядати інформацію про створену навчальну групу, редагувати її та додавати нових студентів в навчальну групу або видаляти їх з групи. Після додавання адміністратором студента в групу, йому відразу надається доступ до всіх навчальних матеріалів, доступних цій групі та розкладу.

Також адміністратор має можливість додавати в систему корпуси та аудиторії. Для кожної аудиторії є можливість вказати кількість місць в ній. Однією з найважливіших функцій адміністратора є робота з розкладом. Адміністратор може додавати пари в розклад студентів та переглядати створений розклад. Функцію управління розкладом додано і до акаунту викладача, щоб зменшити навантаження на адміністратора і інформація в системі максимально швидко ставала актуальною.

В викладачів є можливість завантажувати нові навчальні матеріали. Вони можуть створювати розділи з файлами для нових предметів та додавати до них навчальні матеріали. Кожен предмет має два підрозділи для файлів: лекції та завдання. Тому під час додавання нових навчальних матеріалів є можливість обрати їхній тип.

Для студентів реалізована можливість переглядати інформацію з кабінету студента. Для того щоб її переглядати потрібно ввести логін та пароль від нього, тому що для отримання даних з кабінету студента в запиті мають бути логін та пароль користувача, який їх завантажує. Ніяка інформація з кабінету студента не зберігається в розробленій програмі, а тільки в браузері користувача. Це зроблено для того щоб не потрібно було вводити логін та пароль при кожному запиті, та завантажену

інформацію можна було переглядати в офлайн режимі.

Висновки. У рамках роботи розроблено програмний продукт для розповсюдження навчальних матеріалів.

Розроблено сервер, який обробляє та зберігає всю необхідну для програми інформацію (дані про університет, користувачів, усі завантажені файли).

Також розроблено клієнт для сервера, який включає в себе інтерфейс користувача. Клієнт дозволяє підключатися до сервера з будь-якої точки світу, де є Інтернет, і зручно користуватися функціями системи.

HTTP-запити використовуються для зв'язку між сервером і клієнтом.

Для розробки сервера була використана мова програмування Java. Клієнт був розроблений за допомогою JavaScript.

Новизна роботи полягає у створенні програми, яка водночас дає можливість використовувати функції, необхідні для навчання у вищому навчальному закладі. Наприклад, надання студентам доступу до навчальних матеріалів, можливість для всіх користувачів переглядати свій розклад. Викладачі також можуть переглянути розклад груп, які вони викладають у поточному семестрі. Кожен учень може переглянути розклад учителя, якщо він викладає хоча б один предмет. Іншою важливою функцією є управління структурою університету (керування інформацією про інститути, кафедри і навчальні групи).

Список літератури

1. Бородін М.А. Проекування та розробка системи електронного навчання / М.А. Бородін, В. О. Метельов // *XIV Міжнародна науково-практична конференція магістрантів та аспірантів «Теоретичні та практичні дослідження молодих науковців»* (01–04 грудня 2020 року): матеріали конференції / за ред. проф. Є.І. Сокола. – Харків : НТУ «ХПІ», 2020. – С. 74.
2. Hendra S., Kom. M.T., Yulyani Arifin S., Kom M.M. «Web-based Usability Measurement for Student Grading Information System». *3rd International Conference on Computer Science and Computational Intelligence* 2018.
3. Ghalyia ALFarsi, Jasiya Jabbar, Maryam ALSinani, «Implementing a Mobile Application News Tool for Disseminating Messages and Events of AlBuraimi University College» [Електронний ресурс]; 20 October 2018 <https://doi.org/10.3991/ijim.v12i7.9484>
4. Abdalla Eldow, Mohanaad Shakir, mohamad Yahya Abdullah, Sohail Iqbal Malik. «Using Information Communication Technology as a Teaching tool in Sudanese Governmental Universities of Khartoum State» 2017.
5. Susan, M. K. Impact of Information Communication Technology (ICT) on Professional Development and Educational Needs of Library Professionals in the Universities of Kerala, Cochin University of Science and Technology, Faculty of Technology, India, P.1. 2011.
6. Pasrija V., Kumar S., Srivastava P. R. Assessment of Software Quality: Choquet Integral Approach. *Procedia Technology*. 2012. Vol. 6. P. 153–162
7. Горбаченко И. М. Оценка качества программного обеспечения для создания систем тестирования. *Фундаментальные исследования* 2013. № 6 (часть 4). С. 823–827.
8. Клюев Е.И., Гриненко Е.А. Подход к оценке качества программных средств. *Инженерия программного обеспечения* 2014. № 3 (19). С. 5–14.

9. Garousia V., Feldererbe M., Karapıçakç C. M., Yılmazd U. Testing embedded software: A survey of the literature. *Information and Software Technology* 2018. Vol. 104. P. 14–45.
10. Mehmood N., Petersen M. K., Börstler J., Wnuk K. Regression testing for large-scale embedded software development – *Exploring the state of practice*. *Information and Software Technology*. 2020. Vol. 120. Article 106243.
11. Seo J., Choi B., Yang S. Lightweight embedded software performance analysis method by kernel hack and its industrial field study. *Journal of Systems and Software*. 2012. Vol. 85, issue 1. P. 28–42.
12. Бураков В.В. Методика оценки качества программных средств. *Известия высших учебных заведений. Приборостроение*. 2008. Т. 51, № 1. С. 35–41.
13. Пронина О. И., Пятюков Е. Е. Использование нечетких множеств при определении класса автомобиля. *Вісник Нац. техн. ун-ту «ХПИ»: зб. наук. пр. Сер.: Системний аналіз, управління та інформаційні технології*. Харків: НТУ «ХПИ», 2017. № 28 (1250). С. 41–48.
14. Рутковская Д., Пилинский М., Рутковский Л. *Нейронные сети, генетические алгоритмы и нечеткие системы*. Москва: Горячая линия - Телеком, 2006. 452 с.
15. Гриняев Ю. В. *Теория нечетких множеств*. Учебное пособие для студентов. Томск: Томский государственный университет систем управления и радиоэлектроники, 2008. – 141 с
- University of Science and Technology, Faculty of Technology, India, P.1. 2011.
6. Pasrija V., Kumar S., Srivastava P. R. Assessment of Software Quality: Choquet Integral Approach. *Procedia Technology*. 2012, vol. 6, pp. 153–162
7. Gorbachenko I. M. Ocenka kachestva programmnoho obespecheniya dlya sozdaniya sistem testirovaniya [Quality assessment of software for creating testing systems]. *Fundamental'nye issledovaniya [Basic research]*. 2013, no. 6 (part 4), pp 823–827.
8. Klyuev E.I., Grinenko E.A. Podhod k ocenke kachestva programmyh sredstv [Approach to software quality assessment]. *Inzheneriia prohramnoho zabezpechennia [Software Security Engineering]*. 2014, no. 3 (19), pp. 5–14.
9. Garousia V., Feldererbe M., Karapıçakç C. M., Yılmazd U. Testing embedded software: A survey of the literature. *Information and Software Technology*. 2018, vol. 104, pp. 14–45.
10. Mehmood N., Petersen M. K., Börstler J., Wnuk K. Regression testing for large-scale embedded software development – *Exploring the state of practice*. *Information and Software Technology*. 2020, vol. 120, article 106243.
11. Seo J., Choi B., Yang S. Lightweight embedded software performance analysis method by kernel hack and its industrial field study. *Journal of Systems and Software*. 2012, vol. 85, issue 1, pp. 28–42.
12. Burakov V. V. Metodika ocnki kachestva programmyh sredstv [Software Quality Assessment Methodology]. *Izvestiya vysshih uchebnyh zavedenij. Priborostroenie* [News of higher educational institutions. Instrumentation]. 2008, vol. 51, no. 1, pp. 35–41.
13. Pronina O. I., Patykov E. E. Ispol'zovanie nechetkikh mnozhestv pri opredelenii klassa avtomobilya [Using fuzzy sets when determining a car class]. *Visnyk Natsionalnoho tekhnichnoho universytetu «KhPI»: zbirnyk naukovykh prats. Seriya: Systemnyi analiz, upravlinnia ta informatsiini tekhnolohii* [Bulletin of the National Technical University "KhPI": a collection of scientific papers. Series: Systems Analysis, Control and Information Technology]. 2017, no. 28 (1250), pp. 41–48.
14. Rutkovska D., Pilinski M., Rutkovski L. *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*. Warszawa, Łódź, Wydawnictwo Naukowe PWN, 2004. 410 s. (Russ. ed.: Rutkovska D., Pilinski M., Rutkovski L. *Nejronnye seti, genetycheskie algoritmy i nechetkie systemy*. Moscow, Goryachaya liniya - Telekom Publ., 2006. 452 p.).
15. Grinyaev Yu. V. *Teoriya nechetkikh mnozhestv*. Uchebnoe posobie dlya studentov [Theory of fuzzy sets. Study guide for students]. Tomsk, Tomskij gosudarstvennyj universitet sistem upravleniya i radioelektroniki Publ., 2008. 141 p

References (transliterated)

Надійшла (received) 27.12.2021

Відомості про авторів / Сведения об авторах / About the Authors

Бородин Марія Анатоліївна – Національний технічний університет «Харківський політехнічний інститут», студентка гр. І-М220а кафедри комп'ютерного моделювання процесів та систем; тел.: (057)-707-64-54; e-mail: m.borodin.1999@gmail.com.

Бородин Марія Анатольевна – Национальный технический университет «Харьковский политехнический институт», студентка гр. И-М220а кафедры компьютерного моделирования процессов и систем; тел.: (057)-707-64-54; e-mail: m.borodin.1999@gmail.com.

Borodin Maria Anatoliivna – National Technical University "Kharkiv Polytechnic Institute", student of the group I-M220a of the Department of Computer Modeling of Processes and Systems; tel.: (057)-707-64-54; e-mail: m.borodin.1999@gmail.com.

Метельов Володимир Олександрович – кандидат технічних наук, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри комп'ютерного моделювання процесів та систем; тел.: (057)-707-64-54; e-mail: vometel@gmail.com. ORCID: <https://orcid.org/0000-0002-2633-6296>

Метелёв Владимир Александрович – кандидат технических наук, Национальный технический университет «Харьковский политехнический институт», доцент кафедры компьютерного моделирования процессов и систем; тел.: (057)-707-64-54; e-mail: vometel@gmail.com. ORCID: <https://orcid.org/0000-0002-2633-6296>

Mietielov Volodymyr Oleksandrovych – PhD, National Technical University "Kharkiv Polytechnic Institute", Associate Professor of the Department of Computer Modeling of Processes and Systems; tel.: (057)-707-64-54; e-mail: vometel@gmail.com. ORCID: <https://orcid.org/0000-0002-2633-6296>