

Д. О. КРАСУЛЯ, К. В. ІВАНЧЕНКО, Д. А. ІВАНЧЕНКО

РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ПЕРЕВІРКИ РІВНЯ ЗНАТЬ АНГЛІЙСЬКОЇ МОВИ З ВИКОРИСТАННЯМ МОВИ ПРОГРАМУВАННЯ SWIFT ТА ФРЕЙМВОРКУ UIKIT

Ця наукова робота спрямована на розробку та впровадження інноваційного мобільного застосунку, спрямованого на оцінку рівня знань іноземних мов. Основна мета дослідження полягала в створенні зручного та швидкого інструменту, який дозволяє користувачам не лише визначити свій рівень мовної компетенції, а й виявляти недоліки та визначити оптимальний шлях для подальшого вдосконалення.

В ході дослідження було ретельно проаналізовано існуючі методики та системи оцінювання мовних навичок. На основі цього аналізу були сформульовані докладні функціональні та нефункціональні вимоги до розроблюваного програмного продукту. Вибір технічних та програмних інструментів, зокрема мови програмування Swift та операційної системи iOS, був обґрунтований їхньою актуальністю та можливістю створювати застосунки для всіх пристроїв у екосистемі Apple.

Суттєвою частиною роботи було створення бази даних, використовуючи нереляційну систему Realm для мобільних пристроїв. Застосунок взаємодіє з даними у форматі JSON, що дозволяє ефективно завантажувати та відображати списки питань для користувачів. В процесі розробки використовувалися такі бібліотеки, як Foundation, UIKit, RealmSwift та інші, з метою забезпечення оптимальної продуктивності та надійності застосунку.

Особлива увага приділялася створенню інтуїтивно зрозумілого та зручного інтерфейсу для користувачів. Процес розробки включав в себе ретельне проектування логіки та структури мобільного застосунку з метою забезпечення максимальної зручності під час використання.

Однією з ключових частин дослідження було тестування розробленого застосунку. Здійснювались UI тестування для перевірки коректності та зручності інтерфейсу, тестування сумісності для впевненості у роботі на різних пристроях та функціональне тестування для перевірки відповідності всіх визначених вимог.

В результаті дослідження була успішно розроблена інформаційна система для автоматизованого тестування, яка відповідає високим стандартам та вимогам. Отримані результати можуть служити основою для інтеграції програми у навчальний процес або для надання студентам можливості отримати практичні навички в оцінці своєї мовної компетенції.

Ключові слова: мобільний застосунок, рівень знань, swift, ios, json.

This scientific work is aimed at the development and implementation of an innovative mobile application aimed at assessing the level of knowledge of foreign languages. The main goal of the study was to create a convenient and fast tool that allows users not only to determine their level of language competence, but also to identify shortcomings and determine the optimal path for further improvement.

In the course of the study, the existing methods and systems for evaluating language skills were carefully analyzed. Based on this analysis, detailed functional and non-functional requirements for the developed software product were formulated. The choice of technical and software tools, in particular the Swift programming language and the iOS operating system, was justified by their relevance and the ability to create applications for all devices in the Apple ecosystem.

A significant part of the work was the creation of a database using the non-relational Realm system for mobile devices. The application interacts with data in JSON format, which allows you to efficiently load and display lists of questions for users. In the development process, such libraries as Foundation, UIKit, RealmSwift and others were used in order to ensure optimal performance and reliability of the application.

Special attention was paid to creating an intuitive and convenient interface for users. The development process included a careful design of the logic and structure of the mobile application in order to ensure maximum convenience during use.

One of the key parts of the research was testing the developed application. UI testing was carried out to check the correctness and usability of the interface, compatibility testing to ensure operation on different devices, and functional testing to check compliance with all defined requirements.

As a result of the research, an information system for automated testing that meets high standards and requirements was successfully developed. The obtained results can serve as a basis for the integration of the program into the educational process or to provide students with the opportunity to gain practical skills in assessing their language competence.

Keywords: mobile application, level of knowledge, swift, ios, json.

Вступ.

У сфері освіти мобільні застосунки все більше набувають популярності, забезпечуючи швидкий та зручний спосіб вивчення, засвоєння та перевірки знань. Тому створення мобільного застосунку, який буде вмщати комплексні питання для перевірки знань з іноземної мови є актуальною задачею. Існуючі застосунки для перевірки знань мають питання, які не є комплексними. Тобто це питання або лише на знання граматики, або лише на перевірку словарного запасу. Питання на аудіювання, читання, граматику, а також знання перекладу розширює як комплексність застосунку так і його функціональність. Мобільний застосунок для контролю рівня знань з іноземної мови – це зручний та швидкий спосіб перевірити свій рівень, виявити помилки та задати подальший вектор навчання тієї чи іншої мови.

Проходячи тести у розробленому мобільному

застосунку користувач зможе виявити свій поточний рівень знань обраної ним мови. Такий застосунок може бути корисним і для людей, які здають екзамени з англійської мови, такі як, наприклад, ЗНО. Питання, з яких складається тест генеруються кожен раз, коли користувач буде переходити на вкладку тестування.

Актуальність створення мобільного застосунку для контролю рівня знань іноземної мови обумовлено ще тим, що дуже часто, вивчаючи будь-яку іноземну мову люди не можуть самостійно визначити рівень своїх знань. А саме за допомогою цього мобільного застосунку користувач зможе швидко визначити свій рівень та виявити, що необхідно ще вивчити та чому необхідно надати більше уваги.

Постановка проблеми.

Метою роботи є створення мобільного застосунку, який буде вмщати комплексні питання для перевірки знань з іноземної мови, а також дозволяти

користувачеві перевірити свої знання з іноземної мови, а також користувач повинен мати можливість відслідковувати свій прогрес.

Для досягнення поставленої мети необхідно виконати декілька завдань:

- провести порівняльний аналіз існуючий на ринку аналогів, виявити їх плюси та мінуси;
- вибрати операційну систему, на якій буде створений мобільний застосунок та мову програмування;
- обрати мову (або декілька мов), рівень знань якої користувач може перевірити;
- розробити структуру та логіку мобільного застосунку, із чого він буде складатися та які можливості будуть у користувача;
- розробити алгоритм, який дозволяє користувачеві перевірити свої знання;
- визначити джерело інформації, звідки в застосунку будуть з'являтися тести;
- визначити потрібний функціонал застосунку, виходячи з наявної інформації і можливостей, які надає API потрібного сервісу;
- необхідно розробити інтуїтивно зрозумілий інтерфейс, який буде відповідати гайдлайнам обраної операційної системи (IOS чи Android);
- необхідно визначити, яка база даних буде використовуватися у застосунку, які дані будуть зберігатися;
- здійснити розробку мобільного застосунку, враховуючи інформацію, що була отримана на попередніх етапах виконання проекту;
- реалізувати адаптивність застосунку під всі актуальні моделі смартфонів лінійки Apple;
- здійснити тестування розробленого застосунку з метою знаходження наявних дефектів, недоліків і інших відхилень у поведінки застосунку у порівнянні з результатом, який планувався.

Аналіз останніх досліджень і публікацій.

Існує дуже багато різних методик перевірки знань з іноземної мови. В процесі розробки була поставлена задача обрати найкращу методику, яка б дозволила користувачеві швидко та якісно перевірити свій рівень знань іноземної мови. Вивчивши літературу [1] та проаналізувавши методики оцінювання і перевірки знань з будь якої іноземною мовою, що також описується у виданні [2], можна сказати, що не існує єдиної та чіткої методики, яка б дозволяла однозначно перевірити та визначити точний рівень володіння тієї чи іншою іноземною мовою.

На даний момент існує система оцінювання володіння мовою, яка називається CEFR - Загальноєвропейська система оцінки рівнів володіння іноземною мовою. Страсбурзький університет разом із радою з питань співпраці в галузі культури видав вказівки [3] із загальноєвропейськими рекомендаціями, щодо мовної освіти. Саме там й було запропоновано використовувати систему оцінювання CEFR. У виданні надані рекомендації щодо вивчення, викладання, а також оцінювання рівня володіння тієї чи іншою іноземною мовою.

Існує ціле різноманіття операційних систем та оболонок над мобільними операційними системами, проте, у своїй роботі V.V. Subrahmanyam [4] проаналізував існуючі мобільні операційні системи, та виокремив дві з них: це операційна система Android від Google, яка була представлена у 2008 році, та операційна система IOS від Apple, яка була представлена на рік раніше. Саме ці дві операційні системи займають лідируючі позиції на світовому ринку операційних систем для мобільних застосунків.

Ознайомившись із великою кількістю літератури на тему розробки мобільних застосунків та проаналізувавши сучасні методології перевірки знань з іноземної мови можна зробити такі висновки:

По-перше, в сучасному світі існує дві лідируючі мобільні операційні системи: це Google Android та Apple IOS. Проте, після ознайомлення із статтями вчених, можна зробити висновок, що саме IOS є найбільш надійною та безпечною операційною системою із усіх можливих. Тому, саме ця операційна система буде використовуватись у моїй дипломній роботі. Окрім цього, після аналізу сучасного ринку мобільних застосунків можна з упевненістю сказати, що застосунків аналогів на платформі iOS значно менше, ніж на платформі Android.

По-друге, для того, щоб створити мобільний застосунок на IOS, можна використовувати одну з двох мов програмування, а саме: Objective-C, або Swift [5-7]. Це дві офіційні мови програмування, які були представлені компанією Apple, але для більш швидкої та адаптивної роботи мобільного застосунку краще використовувати більш нову мову програмування Swift, а так як саме ця мова дає можливість створювати не лише мобільні застосунки, а й застосунки для усіх девайсів з екосистеми Apple – вона була обрана мною для написання застосунку для контролю рівня знань з іноземної мови.

По-третє, проаналізувавши літературу по мобільним базам даних, можна сказати, що база даних Realm [8] більш швидка та інтуїтивно зрозуміла, то ж було прийнято рішення використовувати саме цю базу даних для зберігання результатів тестування користувача та подальшої формування статистики занять.

Основна частина.

Увесь застосунок складається із двох основних та декількох допоміжних екранів. Екран із запитанням при кожному натисканні на варіант відповіді перезавантажується з уже новим запитанням. Вся логіка роботи програми базується на тестах, які завантажуються із файлу у JSON форматі, що має масив із завданнями та відповідями до цих завдань.

Кожна відповідь являє собою модель Answer, яка містить поле type, так як відповідь може бути як у форматі тексту, так і у форматі картинки. Поле value, яке відповідає за текст самої відповіді, або у випадку із картинкою – містить посилання на картинку. А також булеве поле isRight, за допомогою якого перевіряється чи являється ця відповідь правильною.

Після отримання даних відбувається декодування

у моделі, а також відображається перше запитання. Питання із масиву запитань витягуються рандомно.

Після того, як користувач завершить тестування, результати тесту зберігаються у локальну базу даних, у клас типу Object. Саме такі класи зберігаються у використовуваній локальній базі даних Realm.

Далі, на основі отриманих результатів формується статистика і користувач має можливість її переглянути. Якщо даних для статистики ще немає, то на екрані статистики відображається Empty State.

Загальну логіку роботи програми наведено на кросфункціональній діаграмі (рис. 1)

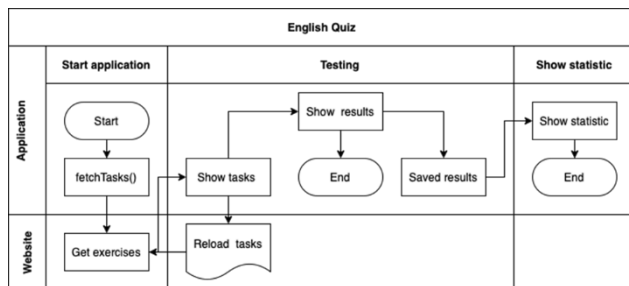


Рис. 1 – Кросфункціональна діаграма застосунку

Структура застосунку:

1 Екран тестування. Екран поділяється на три блоки. Перший блок – задача. Це може бути або текст, або картинка із завданням, або відео. Другий блок – це блок питання. Безпосередньо саме запитання. Третій блок – блок можливих відповідей. У кожному запитанні є лише одна правильна відповідь.

2 Екран статистики – екран, на якому відображається статистика тестувань користувача, вхідними даними є кількість правильних відповідей з кожного тестування, та дата самого тестування. Статистика представлена у вигляді стовпчастого графіку. Формується по дням.

На рис. 2 представлена UML-діаграма активності, що відображає роботу користувача, програми та локальної бази даних.

При кожному початку тестування із масиву запитань випадково дістається запитання, тож кожного разу список запитань та їх послідовність буде відрізнятись.

Також користувач може в будь який момент завершити тестування, тоді поточні результати тесту будуть збережені і тестування завершиться, а користувач зможе перейти на вкладку із статистикою.

На всіх екранах із запитаннями є кнопка перезавантажити список запитань, що дозволяє оновити список запитань та заново почати тестування.

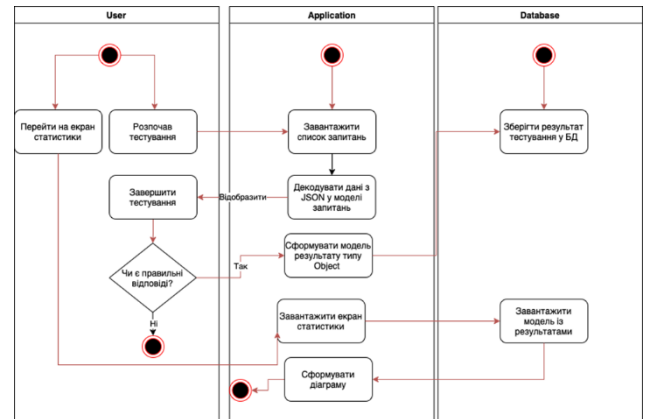


Рис. 2 – UML-діаграма активності у застосунку

При створенні застосунку необхідно було вирішити, яка архітектура застосунку, саме кодової частини програми буде використана.

Проаналізувавши необхідні вимоги до застосунку було вирішено розглядати архітектури MV(X).

Основні частини MV(X) архітектур – це, так чи інакше, моделі, види та контролери.

- Модель - відповідає за використання предметних (бізнес, domain) даних

- View - відповідає за шар подання (GUI). Саме цей слой відповідає за відрисовку UI компонентів. View не знає нічого про бізнес-логіку застосунку, він лише малює те, що йому говорить контроллер.

- Контролер/Презентер/ViewModel – так чи інакше відповідають за зв'язок моделі з контролером. Здебільшого займаються тим, що прокидають події моделі на вигляд, а події виду – на модель, відповідним чином їх перетворюючи та обробляючи. Наприклад, змінюючи модель у відповідь на дії користувача на екрані (виді), або змінюють вигляд у відповідь на зміни моделі. Зазвичай є пасивними учасниками процесу і реагують лише зовнішні стимули (події від Виду чи Моделі).

При запуску програми головний View контролер завантажує список запитань. Екран формується із одного запитання та поділяється на три блоки. Цей екран можна побачити на рисунку 3.

Для створення можливості контролювати свої результати у застосунку було вирішено створювати статистику із відповідями на запитання. Для того, щоб ця статистика не видаляється при кожному закритті застосунку було вирішено використовувати локальну базу даних, яка б запам'ятовувала всі результати та на базі цих даних формувалася статистика.

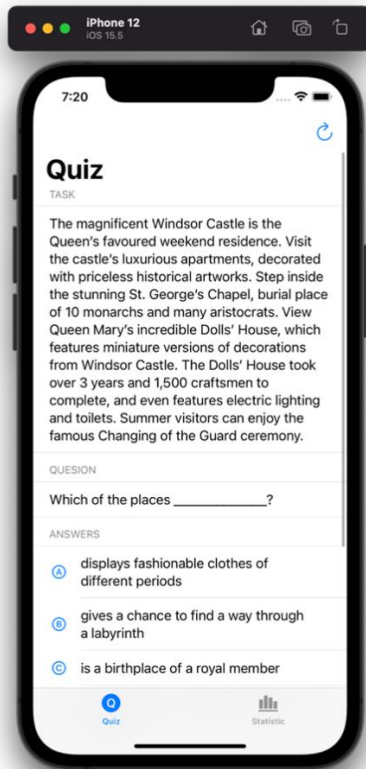


Рис. 3 – Екран із запитанням

Для цього було вирішено використовувати базу даних Realm. Для запису та витягання даних з неї повинні бути створені об'єкти (класи) типу Object.

Даний клас повинен вмщати в себе всі необхідні поля моделі, які необхідно зберігати та які в подальшому використовуються.

Як вже було сказано, для перевірки рівня знань було використано метод тестування із різними видами запитань. Всі запитання беруться із файлу типу JSON. Для завантаження списку запитань був створений метод `getExercise()` (рис. 4).

```
private func getExercises() {
    guard let jsonData = self.readLocalFile(for: "questions") else { return }
    do {
        self.exercises = try JSONDecoder().decode([Exercise].self, from: jsonData)
        self.results = []
        self.updateExercise()
    } catch {
        showAlert(error: error)
    }
}
```

Рис. 4 – Метод завантаження завдань

Сама таблиця бере дані, які їй надійшли із файлу, ці дані представлені у вигляді масиву моделей завдань. Після чого витягується рандомним образом завдання, перевіряється його тип та виходячи з цього типу таблиця формує або коміркою із відео-файлом, або з фото, або просто із текстом. Те ж саме відбувається і з коміркою, що містить відповіді. Там також перевіряється тип відповіді і в залежності від цього формується комірка (Рис. 5).

Для перевірки правильності обраної користувачем відповіді береться індекс комірки із відповіддю, після цього в конкретній моделі відповіді, так званій моделі під назвою Answer, яка була заповнена із JSON файлу,

є поле булевого типу під назвою `isRight`. Саме це поле відповідає за правильність даної відповіді.

В кожному запитанні є лише одна правильна відповідь. Тож лише у однієї моделі Answer поле `isRight` буде дорівнювати `true`.

Будь який мобільний застосунок перед виведенням його на всебічне користування необхідно протестувати. У застосунку повинен бути адаптивний під різні діагоналі пристроїв UI, у застосунку не повинно бути збоїв, тобто життєвий цикл застосунку не повинен перериватися. Цикл повинен завершуватися лише тоді, коли користувач вийде з застосунку.

По-перше, необхідно було перевірити швидкість роботи застосунку та його вплив на оперативну та постійну пам'ять телефону. Результати тестування наведені на рисунку 6.

Для перевірки адаптивності інтерфейсу застосунків до різних діагоналей пристроїв було протестовано застосунок на iPhone 12, що має середню довжину діагоналі екрану, iPhone 13 Pro Max із найбільшим екраном, та iPhone SE першого покоління, який має найменшу довжину діагоналі екрану.

```
switch model {
case .task(let task):
    switch task.type {
    case .video:
        let cell = tableView.dequeueReusableCell(cellClass: VideoCell.self, forIndexPath: indexPath)
        cell.model = model
        return cell
    case .picture:
        let cell = tableView.dequeueReusableCell(cellClass: ImageCell.self, forIndexPath: indexPath)
        cell.setValues(model: model)
        return cell
    default:
        return cell
    }
case .answer(let answer):
    switch answer.type {
    case .picture:
        let cell = tableView.dequeueReusableCell(cellClass: ImageCell.self, forIndexPath: indexPath)
        cell.setValues(model: model, option: alphabet[indexPath.row])
        return cell
    case .text:
        cell.imageView?.image = UIImage(systemName: "\(alphabet[indexPath.row]).circle")
    default:
        break
    }
}
```

Рис. 5 – Формування таблиці із завданнями

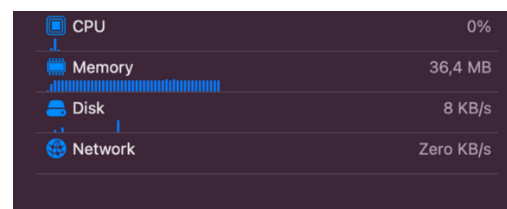


Рис. 6 – Вплив застосунку на CPU, постійну пам'ять, оперативну, та кількість витрачених kb/s

Висновки.

В результаті проведеної роботи було створено мобільний застосунок для контролю рівню знань іноземної мови.

Цей застосунок дає змогу швидко і легко перевірити рівень знань іноземної мови користувача, а також проаналізувати отримані результати та слідкувати за прогресом, що представлений у вигляді статистики, яка зберігається у базі даних та дозволяє користувачеві у будь який час відстежити свій прогрес. Застосунок був створений на платформі IOS за допомогою мови програмування Swift та повністю відповідає вимогам документації програмного продукту.

Розглянуто та використано на практиці роботу із

Cocoa Pods та об'єктно-орієнтованою базою даною RealmSwift. Крім цього, застосунок працює із даними у форматі JSON, що дозволяє завантажувати список запитань та відображати їх користувачеві.

В ході роботи було розроблено методику, за допомогою якої реалізувалася перевірка знань користувача. Це перевірка за допомогою тестувань із різними типами запитань. Також були розглянуті та проаналізовані існуючі на ринку альтернативні застосунки. В ході розробки мобільного застосунку були використані такі бібліотеки, як Foundation, UIKit, RealmSwift та інші. Крім цього, було вивчено та додано у програму роботу з об'єктно-орієнтованою не реляційною базою даних для мобільних девайсів під назвою Realm.

В результаті наукової роботи було розроблено логіку та структуру мобільного застосунку та реалізовано його графічний інтерфейс, проаналізовано переваги та недоліки отриманого результату, а також порівняно застосунок із існуючими аналогами на ринку, а також було проведено UI тестування, тестування сумісності, а також проведено функціональне тестування.

Список літератури

1. Панова Л.С. Методика навчання іноземних мов у загальноосвітніх навчальних закладах /Л.С. Панова, І.Ф. Андрійко, С.В. Тезікова, С.І. Потапенко// Київ Видавничий центр «Академія». - 2010.
2. Присяжна О.Д. Методичні вказівки Перевірка знань іноземної мови (англійської) студентами 1 курсів усіх спеціальностей / О.Д. Присяжна, Л.В. Варчук// Нульовий контроль, 2017.
3. Загальноєвропейські рекомендації з мовної освіти: вивчення, викладання, оцінювання /Науковий редактор українського видання доктор пед. наук, проф. С.Ю.Николаєва. - К.: Ленвіт, 2003.-273с.
4. V.V. Subrahmanyam, Kailasam Swathi, A Study on Mobile Operating Systems and their Recent Advances// Osmania University Campus, Hyderabad, 2011.

5. M. Landman Managing smart phone security risks. Information Security Curriculum Development Conference, ser.InfoSecCD '10. - New York, NY. - 2010. - pp. 145-155. [Online]. Available: <http://doi.acm.org/10.1145/1940941.1940971>

6. Rostislav Fojtik. Swift a New Programming Language for Development and Education. - Digital Science, 2019. - pp.284-295.

7. ZahidaParveen, Nazish. F. Performance Comparison of Most Common High Level Programming Languages, International Journal of Computing Academic Research. - vol. 5, number 5. - 2016. - pp. 246-258.

8. Marin Todorov Realm: Building Modern Swift Apps with Realm Database (Second Edition). - Razeware LLC. - 2019. - 307 p.

References (transliterated)

1. Panova L.S. Metodyka navchannia inozemnykh mov u zahalnoosvitnikh navchalnykh zakladakh /L.S. Panova, I.F. Andriyko, S.V. Tezikova, S.I. Potapenko// Kyiv Vydavnychyy tsentr «Akademii». - 2010.
2. Prysiazna O.D. Metodichni vказivky Perevirka znan inozemnoi movy (anhliiskoi) studentamy 1 kursiv usikh spetsialnostei / O.D. Prysiazna, L.V. Varchuk// Nulovyi kontrol, 2017.
3. Zahalnoievropeiski rekomendatsii z movnoi osvity: vyvchennia, vykladannia, otsiniuvannia /Naukovyy redaktor ukraïnskoho vydannia doktor ped. nauk, prof. S.Iu.Nikolaieva. - K.: Lenvit, 2003.-273s.
4. V.V. Subrahmanyam, Kailasam Swathi, A Study on Mobile Operating Systems and their Recent Advances// Osmania University Campus, Hyderabad, 2011.
5. M. Landman Managing smart phone security risks. Information Security Curriculum Development Conference, ser.InfoSecCD '10. - New York, NY. - 2010. - pp. 145-155. [Online]. Available: <http://doi.acm.org/10.1145/1940941.1940971>
6. Rostislav Fojtik. Swift a New Programming Language for Development and Education. - Digital Science, 2019. - pp.284-295.
7. ZahidaParveen, Nazish. F. Performance Comparison of Most Common High Level Programming Languages, International Journal of Computing Academic Research. - vol. 5, number 5. - 2016. - pp. 246-258.
8. Marin Todorov Realm: Building Modern Swift Apps with Realm Database (Second Edition). - Razeware LLC. - 2019. - 307 p.

Надійшла (received) 13.12.2023

Відомості про авторів / About the Authors

Красуля Дар'я Олексіївна (Krasulia Daria) – Національний технічний університет «Харківський політехнічний інститут», магістр комп'ютерних наук; тел: (066)346-57-14; e-mail: daria.krasulia@infiz.khpi.edu.ua.

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>

Іванченко Дмитро Анатолійович (Ivanchenko Dmytro) – кандидат технічних наук, доцент, Приазовський Державний Технічний Університет, доцент кафедри рухомого складу транспортних систем; тел: (096) 368-12-37; e-mail: ukrsarterps@gmail.com. ORCID: <http://orcid.org/0000-0002-3024-3930>