

## ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.622

О. Д. Азаров<sup>1</sup>, О. І. Черняк<sup>1</sup>, В. В. Залізецький<sup>1</sup>

## ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПРАЦЮВАННЯ ДАНИХ ДИСТАНЦІЙНО-РОЗПОДІЛЕНИХ СИСТЕМ ТА ПОШУКУ ОБ'ЄКТІВ НА МІСЦЕВОСТІ

<sup>1</sup>Вінницький національний технічний університет

**Анотація.** В умовах євроінтеграції, подорожі за кордон стали більш доступними для українців і актуальною стає потреба добре орієнтуватись на незнайомій місцевості. Звісно зараз є безліч застосувань, що поліпшують життя туристам, наприклад безкоштовні Google карти вбудовані в більшість сучасних смартфонів на платформі Android. Але інколи потрібна інформація про розташування більше ніж 20-ти найближчих об'єктів, що не дозволяють зробити Google карти. Окрім того сейсмозвідка вимагає розміщення сенсорів на певній відстані один від одного з покриттям певної території. Потрібні засоби для контролю процесу розміщення сенсорів та відображення відсотку покриття території з врахуванням радіусу дії сенсорів. Також останнім часом, безпілотні літальні апарати активно завойовують своє місце в різних сферах життя. Потрібні інструменти для зручного керування такими апаратами. У статті пропонується програмне забезпечення для опрацювання даних дистанційно-розподілених систем та пошуку об'єктів на місцевості, що може використовуватись для вирішення вищеописаних задач. Запропоноване рішення розроблено з використанням сучасного стеку технологій, є надійним, незалежним від платформи, відносно дешевим та оснащено власним API.

**Ключові слова:** GPS, Google API, Web-сервіс, API, геолокація, сейсмозвідка, дистанційно-розподілені системи, пошук об'єктів на місцевості.

**Аннотация.** Благодаря евроинтеграции, путешествия за границу стали более доступными для украинцев и актуальной становится потребность хорошо ориентироваться на незнакомой местности. Конечно сейчас есть множество приложений, улучшающих жизнь туристам, например, бесплатные Google карты встроены в большинство современных смартфонов на платформе Android. Но иногда нужна информация о расположении более 20-ти ближайших объектов, что не позволяет сделать Google карты. Кроме того, сейсмозвездка требует размещения сенсоров на определенном расстоянии друг от друга с покрытием определенной территории. Нужны средства для контроля процесса размещения сенсоров и отображения процента покрытия территории с учетом радиуса действия сенсоров. Также в последнее время, беспилотные летательные аппараты активно завоевывают свое место в различных сферах жизни. Необходимы инструменты для удобного управления такими аппаратами. В статье предлагается программное обеспечение для обработки данных дистанционно-распределенных систем и поиска объектов на местности, что может использоваться для решения вышеописанных задач. Предложенное решение разработано с использованием современного стека технологий, является надежным, независимым от платформы, относительно дешевым и оснащено собственным API.

**Ключевые слова:** GPS, Google API, Web-сервис, API, геолокация, сейсмозвездка, дистанционно-распределенные системы, поиск объектов на местности.

**Abstract.** Thanks to the European integration, traveling abroad became more accessible for Ukrainians and the need to navigate the unfamiliar terrain is becoming urgent. Of course now there are many applications that improve the lives of tourists, for example, free Google maps are built into most modern smartphones on the Android platform. But sometimes you need information about the location of more than 20 nearby objects that do not allow you to make Google maps. In addition, seismic prospecting requires the placement of sensors at a certain distance from each other with the coverage of a certain area. We need tools to control the placement of sensors and display the percentage coverage of the territory, taking into account the range of sensors. Also recently, unmanned aerial vehicles are actively gaining their place in various spheres of life. Tools are needed for convenient control of such devices. The article proposes software for processing data from remote-distributed systems and searching for objects on the ground, which can be used to solve the above-described tasks. The proposed solution is developed using a modern stack of technologies, it is reliable, platform independent, relatively cheap and equipped with its own API.

**Keywords:** GPS, Google API, Web-service, API, geolocation, seismic exploration, remotely-distributed systems, objects locating.

**DOI:** <https://doi.org/10.31649/1999-9941-2018-42-2-10-15>.

## Вступ

В умовах євроінтеграції, подорожі за кордон стали більш доступними для українців і актуальною стає потреба добре орієнтуватись на місцевості та швидко дізнатись де знаходиться найближчий банк, кафе, стоматолог чи автостоянка. Звісно на сьогодні є безліч застосувань, що поліпшують життя. Є різні навігатори, смартфони із вбудованими GPS датчиками та Google чи іншими картами. Усе це створює можливість для комфортного подорожування [1]. Проте інколи виникають специфічні потреби, наприклад отримати інформацію про розташування усіх об'єктів певного типу в межах певної місцевості, для якоїсь звітності, аналізу чи статистики. Google карти не дозволяють вивантажувати та відображати такі масиви інформації. Та це й не потрібно для такого типу застосування.

Окрім того сейсмозвідка вимагає розміщення сенсорів на певній відстані один від одного з покриттям певної території [2]. При цьому було б дуже зручно бачити в реальному часі скільки території покрито, де розміщено сенсори та де потрібно встановити наступний сенсор.

Також останнім часом безпілотні літальні апарати (БПЛА), які завжди були надбанням військових,

завойовують своє місце ще й у різних сферах цивільного життя [3]. Щодня людство знаходить їм нові застосування, тому потрібне програмне забезпечення (ПЗ) для вирішення цих задач.

### Аналіз останніх наукових досліджень і публікацій

Теоретичним обґрунтуванням процесів геопозиціонування займався Липкін І. А. [4]. Питання розробки систем моніторингу місцезнаходження розглядалися Козловським Е. М., який надав характеристики можливих моделей та способів отримання даних геолокації та описав складності і обмеження таких систем [5]. Кухтій А. та Кухтій С. у своїх роботах обґрунтували доцільність використання інформаційних технологій в навігації для пересічних користувачів та бізнесу [1].

Багатоканальними системами, аналого-цифровими (АЦ) перетвореннями, опрацюванням і аналізом даних з акустичних сенсорів в тім числі й у сейсмозв'язці займається Крупельницький Л. В. [2]. Ефективним вирішенням задач цифрової обробки сигналів з використанням кодів золотої пропорції займається Черняк О. І. [6].

Великий інтерес до навігації та геолокації є у компаній, таких як: Google, SkyRiver, EasyWay [7-10]. Проблемами моніторингу місцезнаходження та навігації займається компанія Google, що надає можливості та сервіси щодо карт і геолокації [7-8]. Компанія SkyRiver - дозволяє спостерігати за місцезнаходженням транспортного засобу або стаціонарного об'єкту з будь-якого сучасного мобільного пристрою [9]. Також варто відзначити український проект для пошуку маршрутів громадського транспорту EasyWay, що допомагає підібрати зручний маршрут з урахуванням фінансових та часових затрат [10].

Питаннями, що покладено в основу статті займалися провідні фахівці як в нашій державі, так і за кордоном. Фізичні та юридичні особи, згадані вище зробили значний вклад у розвиток тем геолокації, навігації, сейсмозв'язки, тощо. Але в існуючих наукових дослідженнях та розробках в цілому не вирішуються питання опрацювання даних дистанційно-розподілених систем та пошуку об'єктів на місцевості, та вимагається впровадження сучасних мережевих технологій збору, збереження та опрацювання даних.

### Структура і характеристики програмного забезпечення

Структурна організація ПЗ для опрацювання даних дистанційно-розподілених систем та пошуку об'єктів на місцевості зображена на рис. 1.

ПЗ для опрацювання даних дистанційно-розподілених систем та пошуку об'єктів на місцевості розроблено з використанням сучасного стеку технологій: Java, Spring Framework, Maven, MongoDB, Swagger, Docker, Google Maps Services. Обраний стек є добре задокументованим та вільним у використанні, що дозволяє зробити ПЗ надійним, незалежним від платформи та відносно дешевим [11].

Розроблене ПЗ дозволяє:

- знаходити та відображати на карті найближчі до місцезнаходження користувача об'єкти;
- здійснювати пошук відносно заданої координати та радіусу пошуку;
- здійснювати пошук в межах певної місцевості (населеного пункту, району і т.д.);
- відображати межі заданої території та де здійснюється пошук в реальному часі;
- отримувати усі типи об'єктів, передбачені сервісом Google Places API, наприклад ресторани, аеропорти, кінотеатри, а також користувацькі об'єкти, такі як: сейсмометри, БПЛА (за умови інтеграції з системою);
- визначати координати для наступної області пошуку, для розміщення сейсмографу, або ж координату та напрямок для переміщення БПЛА.

Користувачі мають змогу використовувати функціональні можливості даного ПЗ через Web-інтерфейс, або ж інтегрувати їх у свої розробки через API. Оскільки, функціонал реалізовано у вигляді мікро-сервісу, то існує можливість об'єднати його з іншими мікро-сервісами, що також мають своє API. Це дозволяє розширити кількість функціональних можливостей зібраних в інтерактивному Swagger документі.

На рисунку 1 зображено декілька сервісів, а саме CGE та CGP. CGE – аббревіатура від City Geo Explorer. Це власне і є ПЗ для опрацювання даних дистанційно-розподілених систем та пошуку об'єктів на місцевості. Інший сервіс зображений на рисунку CGP – аббревіатура від Codes of Golden Proportion. Це сервіс для віддаленого виконання арифметичних і логічних операцій в кодах золотої пропорції, якому відповідають публікації [12, 13, 14]. Для обробки даних дистанційно-розподілених систем (БПЛА та сейсмометрів обладнаних GPS-трекерами) стане у нагоді простота виконання арифметичних і логічних операцій та надлишковість кодів золотої пропорції. Це створює можливості для ряду напрямків, зокрема, підвищенні точності, синхронізації, контролю й діагностики [12]. Тому вирішено використати наявні напрацювання для віддаленого виконання арифметичних і логічних операцій в кодах золотої пропорції і об'єднати їх з тематикою цієї статті. Тим більше, що для керування запитами, збору документації з сервісів та авторизації користувачів потрібен API-шлюз, що також розроблено в попередніх публікаціях.

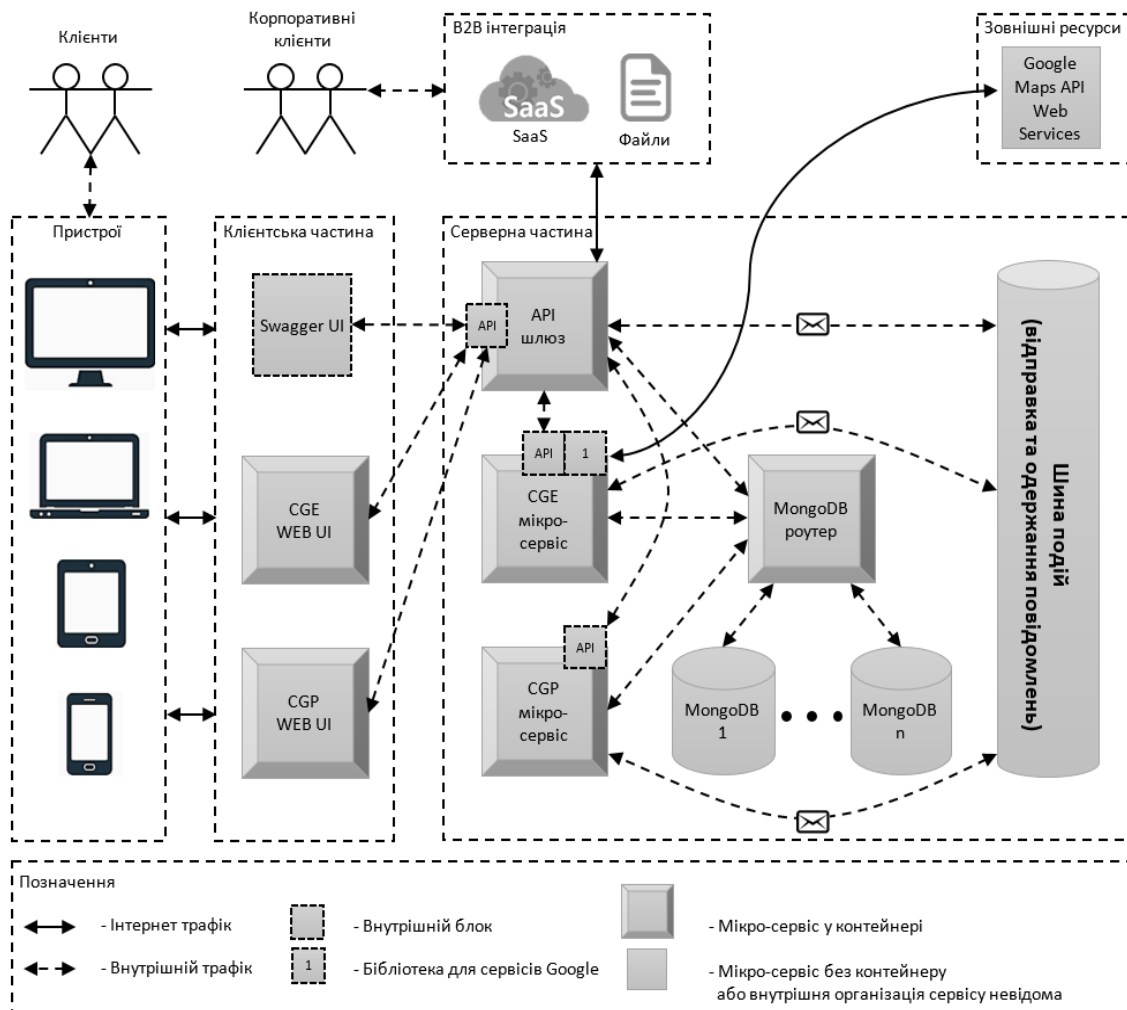


Рисунок 1 – Структурна організація розробленого ПЗ для опрацювання даних дистанційно-розподілених систем та пошуку об'єктів на місцевості

API-шлюз та CGP мікро-сервіс розроблено з використанням Scala, PlayFramework, SBT, MongoDB, Swagger, Apache Kafka, що дещо відрізняється від основного стеку ПЗ для опрацювання даних дистанційно-розподілених систем та пошуку об'єктів на місцевості. Спільним є мікро-сервісна архітектура, база даних MongoDB та генератор інтерактивної документації Swagger. Відмінності в стеку не є проблемою, адже сервіси ізольовані і комунікують лише за допомогою http запитів, json файлів, обмінюються повідомлення через Kafka чи RabbitMQ. Принцип комунікації між сервісами за допомогою Kafka чи RabbitMQ зображено на рис. 2.

Комунікація через такі засоби як Kafka або RabbitMQ необхідна тому, що ці засоби дозволяють створювати чергу повідомлень, які поступово будуть оброблятися. У випадку, якщо якийсь сервер був тимчасово недоступний, обробка буде продовжена з того місця де зупинилися. Темі можуть мати багато розділів і в кожного свій власний індекс з якого потрібно продовжити обробку. Відправник може надсилати повідомлення одразу на декілька тем, або розділів. Так само одержувачів підписаних на одну тему може бути багато, щоб обробити необхідну кількість даних. Все це не можливо при використанні звичайних http запитів до сервісів.

Як видно з рис. 1, сервіси ізольовано у контейнерах. Для цього використовується контейнерна віртуалізація на базі Docker. Відмінності обраного способу віртуалізації від звичайної віртуалізації зображено на рис. 3.

Використання контейнерної віртуалізації дає такі переваги:

- додатковий захист. Кожен сервіс безпечно ізольовано в окремому контейнері. На зовні ми віддаємо, лише те, що потрібно;
- сервіси можуть використовувати різні версії залежностей і не конфліктувати між собою;
- Docker надає в користування безліч готових конфігурацій та образів для підняття контейнерів;

- завдяки технології контейнерної віртуалізації, що працює без додаткового навантаження гіпервізора і дозволяє спільно використовувати ресурси батьківської ОС, можна запускати багато контейнерів на одному сервері одночасно;
  - легко обмінюватись готовим сконфігурованим контейнером та запускати його на різних серверах з різною апаратурою та ОС;
  - контейнери можуть працювати на локальній машині, як реальній так і на віртуальній машині в дата центрі чи в хмарі;
- можна піднімати декілька екземплярів сервісів і розподіляти навантаження між ними.

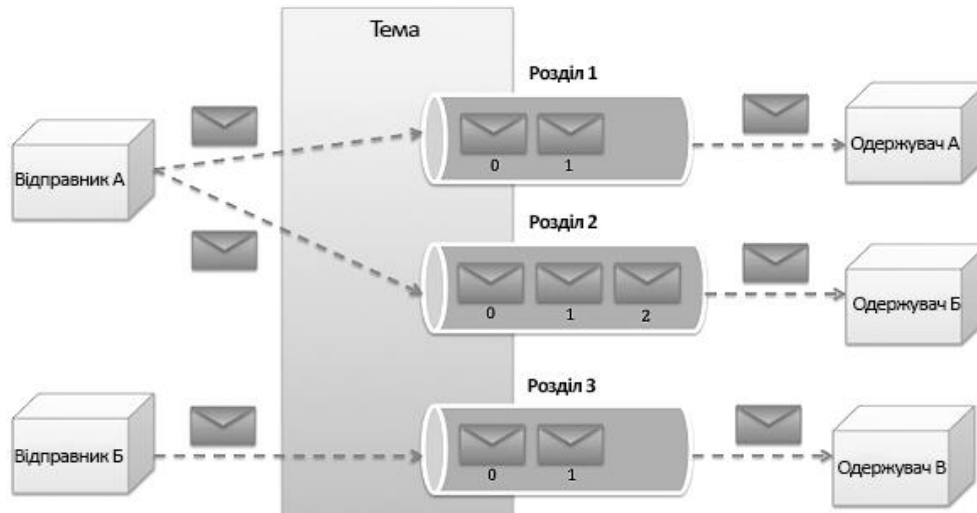
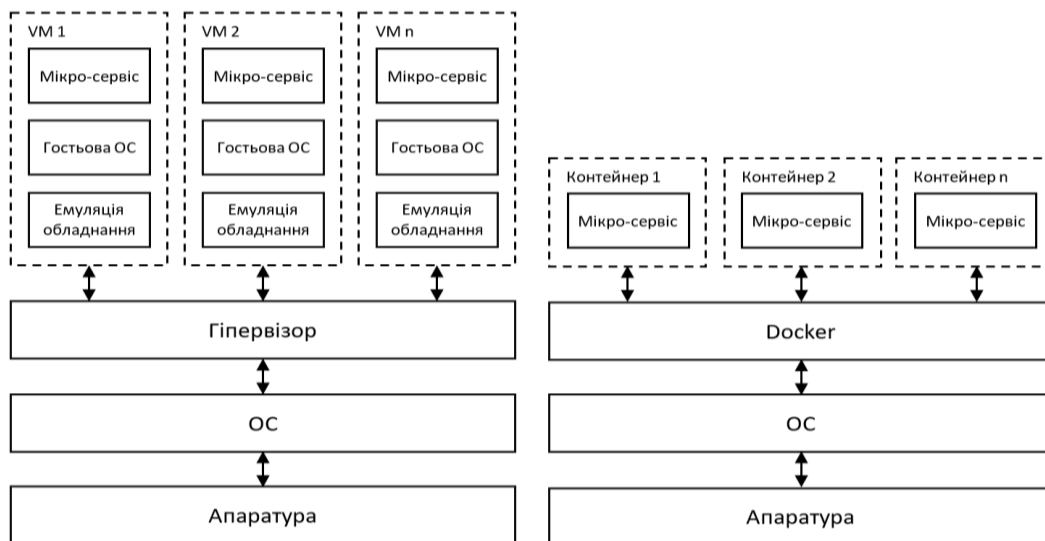


Рисунок 2 – Комунікація між сервісами



Звичайна (гіпервізорна) віртуалізація

Контейнерна Docker віртуалізація

Рисунок 3 – Порівняння контейнерної та звичайної віртуалізації

На рис. 1 зображено декілька баз даних і роутер. Це пояснюється тим, що база даних спроектована для обробки великої кількості даних із забезпеченням високої продуктивності. Це можливо завдяки горизонтальному масштабуванню. В термінології баз даних це називають шардингом. Щоб не перевантажувати один сервер ми розподіляємо інформацію на декілька серверів, а роутер визначає до якої бази даних звернутись за потрібними даними.

Завдяки такому способу організації ПЗ маємо ряд суттєвих переваг та можливостей [14]:

- можливість інтегрувати сервіси в кластер мікро-сервісів з різним функціоналом призначеним для моделювання різних процесів, обробки та аналізу даних отриманих з АЦ систем;
- можливість працювати з будь-якого пристрою, що має доступ до мережі Інтернет;

- можливість розгортання розробленої системи в хмарній інфраструктурі;
- можливості аналітики з інформацією про те, що відбувається в будь-якому сегменті багатовимірної і багатоаспектної ланцюжка сервісів у рамках цифрового продукту або послуги;
- можливість розповсюджувати програмний продукт за принципом Software as a Service (SaaS), де замовники платять не за володіння програмами як такими, а за їх використання;
- можливість створити модель партнерської цифрової екосистеми, а також спрощення інтеграції продуктів між собою.

### Висновки

ПЗ розроблено з використанням сучасного стеку технологій: Java, Spring Framework, Maven, MongoDB, Swagger, Docker, Google Maps Services та є надійним, незалежним від платформи та відносно дешевим. Користувачі мають змогу використовувати функціональні можливості через Web-інтерфейс, або ж інтегрувати функціонал у свої розробки через спеціальне API.

Створене ПЗ для опрацювання даних дистанційно-розподілених систем та пошуку об'єктів на місцевості дозволяє:

- знаходити та відображати на карті найближчі до місцезнаходження користувача об'єкти;
- здійснювати пошук відносно заданої координати та радіусу пошуку;
- здійснювати пошук в межах певної місцевості (населеного пункту, району і т.д.);
- відображати межі заданої території та де здійснюється пошук в реальному часі;
- отримувати усі типи об'єктів, передбачені сервісом Google Places API, наприклад ресторани, аеропорти, кінотеатри, а також користувацькі об'єкти, такі як: сейсмометри, БПЛА (за умови інтеграції з системою);
- визначати координати для наступної області пошуку, для розміщення сейсмографу, або ж координату та напрямок для переміщення БПЛА.

Завдяки запропонованій архітектурі ПЗ маємо ряд суттєвих переваг та можливостей:

- можливість інтегрувати сервіси в кластер мікро-сервісів з різним функціоналом призначеним для моделювання різних процесів, обробки та аналізу даних отриманих з АЦ систем;
- можливість працювати з будь-якого пристрою, що має доступ до мережі Інтернет;
- можливість розгортання розробленої системи в хмарній інфраструктурі;
- можливості аналітики з інформацією про те, що відбувається в будь-якому сегменті багатовимірної і багатоаспектної ланцюжка сервісів у рамках цифрового продукту або послуги;
- можливість розповсюджувати програмний продукт за принципом SaaS, де замовники платять не за володіння програмами як такими, а за їх використання;
- можливість створити модель партнерської цифрової екосистеми, а також спрощення інтеграції продуктів між собою.

### Література

- [1] А. Кухтій, та С. Кухтій, Формування туристичних маршрутів з використанням сучасних інформаційних технологій. Львів, Україна: ЛДУФК, 2014, 242 с.
- [2] Л. В. Крупельницький, «Характеристики і структури багатоканальних АЦ-систем, що самокорегуються, для аналізу аудіо сигналів,» на V Міжнар. наук.-практ. конф. Методи та засоби кодування, захисту й ущільнення інформації, Вінниця, 2016, с. 129-133.
- [3] Arkadiy Sedov, «Огляд сфер використання БПЛА в повсякденному житті», [Електронний ресурс] – Режим доступу: <http://www.50northspatial.org/ua/uavs-everyday-life/>. Дата звернення: 26.08.18.
- [4] И. А. Липкин, Спутниковые навигационные системы. Москва: Россия: Вузовская книга, 2001, 86 с.
- [5] Е. М. Козловский, Искусство позиционирования. Москва, Россия: Вокруг света, 2006, 280 с.
- [6] О. І. Черняк, «Потокові методи і засоби повнофункціональної побітової арифметики зі зменшеними витратами обладнання,» дис. канд. техн. наук., фак-т інфор. техн. і комп. Інженер., Він. нац. техн. ун-т, Вінниця, 2013. - 20 с.
- [7] Developer's Guide [Електронний ресурс]. – Режим доступу: <https://developers.google.com/api-client-library/java/google-api-java-client/dev-guide>. Дата звернення: 26.08.18.
- [8] Java client library for Google Maps API Web Services [Електронний ресурс]. – Режим доступу: <https://github.com/googlemaps/google-maps-services-java>. Дата звернення: 26.08.18.
- [9] SkyRiver [Електронний ресурс]. – Режим доступу: <http://skyfleet.com.ua/>.
- [10] Система пошуку маршрутів громадського транспорту EasyWay [Електронний ресурс]. – Режим доступу: <https://www.eway.in.ua/>. Дата звернення: 26.08.18.
- [11] О. Д. Азаров, Л. В. Крупельницький, О. І. Черняк, та В. В. Залізецький «Система дистанційної колективної самопідготовки,» Інформаційні технології та комп'ютерна інженерія. №2(36), с. 15-20, 2016.

[12] О. Д. Азаров, О. І. Черняк, В. В. Залізецький «Програмне забезпечення для віддаленого виділення цілої і дробової частин чисел у кодах золоті пропорції,» на VI Міжнар. наук.-практ. конф. Методи та засоби кодування, захисту й ущільнення інформації, Вінниця, 2017. – с. 163-166.

[13] Залізецький В. В. «Програмне забезпечення для віддаленого виконання арифметичних і логічних операцій в кодах золоті пропорції,» Матеріали конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2018) [Електронний ресурс], Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2018/paper/view/3756>. Дата звернення: 26.08.18.

[14] Залізецький В. В. «Програмне забезпечення для віддаленого додавання та віднімання кодів золоті пропорції в порозрядному режимі,» Матеріали XLVII науково-технічної конференції підрозділів [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2018/paper/view/5269>. Дата звернення: 26.08.18.

Стаття надійшла: 27.08.18.

#### Відомості про авторів

**Азаров Олексій Дмитрович**, д.т.н., професор, декан факультету інформаційних технологій та комп'ютерної інженерії Вінницького національного технічного університету, заслужений працівник освіти України.

**Черняк Олександр Іванович**, к. т. н., доцент кафедри обчислювальної техніки Вінницького національного технічного університету; адреса: 21021.

**Залізецький Василь Володимирович**, аспірант кафедри обчислювальної техніки Вінницького національного технічного університету.

O. D. Azarov<sup>1</sup>, O. I. Chernyak<sup>1</sup>, V. V. Zalizetskiy<sup>1</sup>

## SOFTWARE FOR PROCESSING DATA OF DISTANCE-DISTRIBUTED SYSTEMS AND SEARCHING FOR OBJECTS ON A LOCATION

<sup>1</sup>Vinnitsia National Technical University