ISSN 2786-7226
DOI: 10.53920/ITS

Науковий журнал

# ITSYNERGY

2023 Випуск 2 (5)

Київ, 2023

**SCIENTIFIC JOURNAL**
# IT SYNERGY

*Published since 2021 year* *Two time a year*

ISSN 2786-7226 Kyiv, 2023, Issue 2 (5)

**Establishers:** Academician Yuriy Bugay International Scientific and Technical University

**The journal is included in scientometric databases:** Google Scholar, Index Copernicus, CrossRef

**The journal publishes the results of scientific research in the following specialties:**
121 – Software engineering; 122 – Computer science;
172 – Telecommunications and radio engineering.



Recommended for publication by the decision of the Academician Yuriy Bugay International Scientific and Technical University (Ukraine), protocol № 05/2324 from 21.12.2023

**Editorial board address:** Scientific journal «IT SYNERGY», Academician Yuriy Bugay International Scientific and Technical University, provulok Khersonskyi (Mahnitohorskyi), 3, Kyiv, 02094, Ukraine

☎ (066) 353-55-31 journal@istu.edu.ua http://its.istu.edu.ua

Registered by the Ministry of Justice of Ukraine Certificate of state registration of the print media Series KB № 24967-14907P dated 20.09.2021

**НАУКОВИЙ ЖУРНАЛ**
# IT SYNERGY

*Засновано у червні 2021 року* *Виходить 2 рази на рік*

ISSN 2786-7226 Київ, 2023, Випуск 2 (5)

**Засновник:** Заклад вищої освіти «Міжнародний науково-технічний університет імені академіка Юрія Бугая»

**Журнал включено до наукометричних баз:** Google Scholar, Index Copernicus, CrossRef

**У журналі публікуються результати наукових пошуків зі спеціальностей:**
121 – Інженерія програмного забезпечення; 122 – Комп'ютерні науки;
172 – Телекомунікації та радіотехніка.



Рекомендовано до друку рішенням Вченої ради ЗВО «МНТУ»,
протокол №05/2324 від 21.12.2023

**Адреса редакції:** Науковий журнал «IT SYNERGY», ЗВО «МНТУ»,
провулок Херсонський (Магнітогорський), 3, м. Київ, 02094, Україна

☎ (066) 353-55-31 ✉ journal@istu.edu.ua 🌐 http://its.istu.edu.ua

Зареєстровано Міністерством юстиції України
Свідоцтво про державну реєстрацію друкованого засобу масової інформації
Серія КВ № 24967-14907Р від 20 вересня 2021 року

# ЗМІСТ

Леся Анатоліївна ДІТКОВСЬКА, Сергій Миколайович КОВАЛЕНКО,
Олександр Миколайович МАКОВЕЙЧУК

**ФОРМУВАННЯ КОМПЕТЕНТНОСТЕЙ ТЕХНІКА-ПРОГРАМІСТА ВІДПОВІДНО ДО ЗАПИТІВ СУЧАСНОГО РИНКУ ПРАЦІ ..... 6**

Костянтин Олександрович ТКАЧЕНКО, Владислав Юрійович ЯКИМЕНКО

**ДЕЯКІ АСПЕКТИ ВИКОРИСТАННЯ АСИНХРОННОЇ КОМУНІКАЦІЇ У МІКРОСЕРВІСНІЙ АРХІТЕКТУРІ ..... 27**

Олександр Іванович ГОЛУБЕНКО, Андрій Вікторович ЛЕМЕШКО,
Олександр Сергійович ЦВИК, Юрій Валентинович МІШКУР

**ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ В ЛОКАЛЬНИХ МЕРЕЖАХ ЗА ДОПОМОГОЮ КОНТРОЛЮ ТРАФІКУ ..... 44**

Олександр Андрійович ТКАЧЕНКО, Богдан Юрійович ВОЛОХОНЕНКО

**ДЕЯКІ АСПЕКТИ РОЗРОБКИ ТА ВИКОРИСТАННЯ ГЕОІНФОРМАЦІЙНИХ В СІЛЬСЬКОМУ ГОСПОДАРСТВІ ..... 52**

Олександр Іванович ГОЛУБЕНКО, Андрій Вікторович ЛЕМЕШКО,
Максим Володимирович КУЗЬМЕНКО, Євген Олександрович ДЕГТЯРЬОВ

**ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У КІБЕРБЕЗПЕЦІ ..... 71**

Ольга Іванівна ТКАЧЕНКО, Максим Володимирович КОВАЛЬЧУК

**АВТОМАТИЗАЦІЯ РОЗГОРТАННЯ ХМАРНИХ ФУНКЦІЙ З ВИКОРИСТАННЯМ SERVERLESS ФРЕЙМВОРКУ: ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ ..... 82**

Андрій Вікторович ЛЕМЕШКО, Артем Васильович АНТОНЕНКО,
Олександр Максимович МАТВІЙЧУК, Олександр Сергійович ДМИТРЕНКО,
Вадим Юрійович БЕРЕЗДЕЦЬКИЙ

**УПРАВЛІННЯ ТРАФІКОМ В ГІБРИДНІЙ ПРОГРАМНО-ВИЗНАЧЕНІЙ МЕРЕЖІ ..... 98**

Ольга Іванівна ТКАЧЕНКО, Олексій Михайлович ТИШУРА

**ДЕЯКІ АСПЕКТИ РОЗРОБКИ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ COFFEE++ ..... 115**

Юрій Валентинович ДЕМЧЕНКО

**БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ ВАГОННОГО ГОСПОДАРСТВА ..... 134**

# CONTENTS

Lesia DITKOVSKA, Serhii KOVALENKO, Oleksandr MAKOVEICHUK

**DEVELOPMENT OF THE COMPETENCES OF THE TECHNICIAN-PROGRAMMER AS DEMANDED BY THE MODERN LABOR MARKET.....6**

Kostiantyn TKACHENKO, Vladyslav YAKYMENKO

**SOME ASPECTS OF USING ASYNCHRONOUS COMMUNICATION IN MICROSERVICES ARCHITECTURE .....27**

Oleksandr GOLUBENKO, Andrii LEMESHKO, Oleksandr TSVYK, Yurii MISHKUR

**ENSURING INFORMATION SECURITY IN LOCAL NETWORKS USING TRAFFIC CONTROL .....44**

Olexandr TKACHENKO, Bohdan VOLOKHONENKO

**SOME ASPECTS OF THE DEVELOPMENT AND USE OF GEOINFORMATION IN AGRICULTURE .....52**

Oleksandr GOLUBENKO, Andrii LEMESHKO, Andrii POLISHCHUK, Maksym KUZMENKO, Eugene DEGTYAREV

**RESEARCH ON THE APPLICATION OF ARTIFICIAL INTELLIGENCE IN CYBER SECURITY..... 71**

Olha TKACHENKO, Maksym KOVALCHUK

**AUTOMATION OF THE DEPLOYMENT OF CLOUD FUNCTIONS USING THE SERVERLESS FRAMEWORK: PROBLEMS AND PERSPECTIVES .....82**

Andriy LEMESHKO, Artem ANTONENKO, Oleksandr MATVIICHUK, Oleksandr DMYTRENKO, Vadym BEREZDETSKYI

**MODELING OF WIRELESS NETWORKS IN OMNET ++ ENVIRONMENT INVOLVING INET FRAMEWORK.....98**

Olha TKACHENKO, Oleksii TYSHURA

**SOME ASPECTS OF DEVELOPMENT OF WEB-ORIENTED SYSTEM COFFEE++.....115**

Yurij ДЕМЧЕНКО

**SECURITY OF INFORMATION SYSTEMS OF THE CARRIAGE ECONOMY..... 134**

*УДК 377:004*
*DOI: https://doi.org/10.53920/ITS-2023-2-1*

**Леся Анатоліївна ДІТКОВСЬКА,**
кандидат педагогічних наук, доцент,
Заклад вищої освіти «Міжнародний науково-технічний університет
імені академіка Юрія Бугая»
ORCID ID: 0000-0002-2112-4550
**Сергій Миколайович КОВАЛЕНКО,**
кандидат фізико-математичних наук, доцент,
Заклад вищої освіти «Міжнародний науково-технічний університет
імені академіка Юрія Бугая»
ORCID ID: 0000-0002-8315-1589
**Олександр Миколайович МАКОВЕЙЧУК,**
доктор технічних наук,
доцент кафедри комп'ютерних наук
та інженерії програмного забезпечення,
Заклад вищої освіти «Міжнародний науково-технічний університет
імені академіка Юрія Бугая»
ORCID ID: 0000-0003-4425-016X

# ФОРМУВАННЯ КОМПЕТЕНТНОСТЕЙ ТЕХНІКА-ПРОГРАМІСТА ВІДПОВІДНО ДО ЗАПИТІВ СУЧАСНОГО РИНКУ ПРАЦІ

***У дослідженні означено первинні посади, що може займати технік-програміст на сучасному ринку праці. Подано перелік типових спеціалізованих задач у галузі інформаційних технологій, які здатен виконувати фаховий молодший бакалавр з інженерії програмного забезпечення. Проведено порівняльний аналіз загальних та фахових компетентностей фахового молодшого бакалавра з інженерії програмного забезпечення та бакалавра з інженерії програмного забезпечення, визначених відповідними стандартами. Визначено окремі компетентності, що є необхідними для фахових молодших бакалаврів з інженерії програмного забезпечення, але не були враховані у Стандарті фахової передвищої освіти за освітньо-професійним ступенем фаховий молодший бакалавр з галузі знань 12 Інформаційні технології, спеціальність 121 Інженерія програмного забезпечення. Висвітлено практичні аспекти формування ключових компетентностей майбутніх техніків-програмістів***

***у процесі їх навчання за освітньо-професійною програмою фахового молодшого бакалавра з інженерії програмного забезпечення. Означено інструменти їх формування у процесі підготовки майбутніх фахівців IT-сфери.***

***Ключові слова:*** *IT-сфера, технік-програміст, загальні компетентності, фахові компетентності, освітні компоненти, фаховий молодший бакалавр з інженерії програмного забезпечення.*

**Lesia DITKOVSKA**
PhD in Pedagogical Sciences, Associate Professor,
Higher Education Institution «Academician Yuriy Bugay International Scientific and Technical University»
**Serhii KOVALENKO**
PhD in Physical and Mathematical Sciences, Associate Professor,
Higher Education Institution «Academician Yuriy Bugay International Scientific and Technical University»
**Oleksandr MAKOVEICHUK**
Doctor of Technical Sciences,
Higher Education Institution «Academician Yuriy Bugay International Scientific and Technical University»

## DEVELOPMENT OF THE COMPETENCES OF THE TECHNICIAN-PROGRAMMER AS DEMANDED BY THE MODERN LABOR MARKET

***The study highlights the position of a software engineer on the contemporary job market. It provides a list of typical specialized tasks in the field of information technology that a junior bachelor in software engineering is capable of performing. A comparative analysis is conducted between the general and professional competencies of a junior bachelor in software engineering and a bachelor in software engineering as defined by respective standards. It identifies specific competencies necessary for junior bachelor professionals in software engineering that were not considered in the Standards of Higher Education for the educational-professional degree of junior bachelor in the field of knowledge 12 Information Technology, specialization 121 Software Engineering. Practical aspects of forming key competencies for future software engineer-technicians during their education within the educational-professional program for junior bachelor***

***in software engineering are illuminated. The tools for shaping these competencies during the preparation of future IT professionals are outlined.***

***Keywords:*** *IT-sphere, technical programmer, general competences, professional competences, primary disciplines, professional junior bachelor in software engineering.*

**Постановка проблеми.** Стрімкий розвиток інформаційно-комунікаційних технологій та цифровізація суспільства спричинили кардинальні зміни ринку праці. Серед найбільш затребуваних фахівців в останні роки є фахівці ІТ-спеціальностей, що, у свою чергу, актуалізує особливу увагу на їх підготовці. Зважаючи на вагоме значення цифрових технологій у подальшому розвитку усіх сфер життєдіяльності людини, проблеми підготовки фахівців ІТ-спеціальностей повинні стати пріоритетом як для освітньої сфери так і держави в цілому. Серед головних не вирішених проблем підготовки ІТ-фахівців слід зазначити такі: розробка якісних стандартів освіти та освітніх програм; їх постійне удосконалення відповідно до сучасних вимог ринку праці, який є наразі надзвичайно глобалізованим; узгодження стандартів освіти з міжнародними стандартами ІТ-професій; гармонізація стандартів для різних рівнів освіти задля ефективної реалізації принципу сприяння навчанню впродовж життя; визначення придатності працевлаштування фахового молодшого бакалавра з ІТ-спеціальностей, зокрема, спеціальності 121 Інженерія програмного забезпечення, відповідно до вимог сучасного ринку праці.

**Аналіз останніх досліджень і публікацій.** Фахова передвища освіта в Україні наразі перебуває на стадії активного становлення. Тому дослідженню проблем у цій сфері та визначення шляхів підвищення ефективності системи фахової передвищої освіти загалом, а також підготовки конкурентоспроможних фахових молодших бакалаврів із окремих спеціальностей приділяється значна увага науковців. Так, у роботі [13] подано аналітичний огляд ефективності фахової передвищої освіти станом на 2022 рік, а у роботі [12] досліджено стан розробленості проблеми становлення конкурентоспроможності майбутніх фахових молодших бакалаврів з інженерії програмного забезпечення та зроблено висновок про необхідність постійного оновлення змі-

сту освітньої програми, методів, технологій, засобів, підручників. Проте ґрунтовного аналізу сучасного ринку праці ІТ-фахівців та визначення первинних посад, що може займати фаховий молодший бакалавр з інженерії програмного забезпечення у наукових публікаціях не наведено. Тому це потребує додаткових досліджень.

Основою для підготовки фахівців певних спеціальностей є відповідні стандарти освіти. Процедури розробки та затвердження стандартів вищої та фахової передвищої освіти ухваленні законами України «Про вищу освіту» (2014 рік) та «Про фахову передвищу освіту» (2019 рік). Наразі розроблені та затверджені стандарти освіти за переважною більшістю спеціальностей за усіма рівнями вищої освіти та фахової передвищої освіти. Аналіз стандартів зі спеціальності 121 Інженерія програмного забезпечення галузі знань 12 Інформаційні технології фахової передвищої освіти [7] та першого (бакалаврського) рівня вищої освіти [10] показав, що вони мають ряд суттєвих недоліків, які не дають змоги чітко визначити місце фахового молодшого бакалавра з інженерії програмного забезпечення (техніка-програміста за термінологією Державного класифікатора професій) на сучасному ринку праці.

У той же час результати розвідок компанії GlobalLogic як одного з лідерів у сфері цифрової інженерії [5] засвідчили, що топ-10 вакансій у галузі програмної інженерії в Україні та регіоні EMEA (регіон, який включає Європу, Близький Схід та Африку) містять вимоги щодо знання та досвіду використання технологій Automotive (AUTOSAR), DevOps, C++, Java, JavaScript, Python, а основними функціями таких фахівців є участь у створенні програмного забезпечення на різних етапах його життєвого циклу, починаючи з аналізу вимог до програмного продукту і закінчуючи гарантуванням його стійкої роботи та постійного оновлення на вимогу замовників (користувачів). До програмних продуктів можна віднести вебсайти, бази даних, вебдодатки, мобільні додатки, системи управління контентом [10, 5, 7].

**Метою статті** є з'ясування первинних посад, що може займати технік-програміст на сучасному ринку праці, уточнення вимог до його підготовки, узгодження компетентностей фахівців з інженерії програмного забезпечення освітніх ступенів фахового молодшого бакалавра та бакалавра, а також аналіз досвіду формування клю-

чових компетентностей у фахових молодших бакалаврів впродовж їх навчання у закладах фахової передвищої освіти.

**Виклад основного матеріалу дослідження.** Для з'ясування первинних посад, що може займати технік-програміст на сучасному ринку праці, вимог до його підготовки, які б враховували перспективні напрямки розвитку IT-сфери, нами було проведено дослідження ринку праці за допомогою контент аналізу вебплатформ працевлаштування (work.ua, jobs.ua тощо). За результатами можна зробити висновок, що сучасний ринок праці у сфері інженерії програмного забезпечення оперує, зазвичай, іншими професійними назвами робіт (посадами) ніж визначеними Державним класифікатором професій (технік-програміст та інженер-програміст). Найчастіше у пропозиціях на працевлаштування фахівців, які займаються створенням програмного забезпечення на усіх етапах його життєвого циклу, вживають назву «Software Developer» (Розробник програмного забезпечення), рідше «Software Engineer» (Інженер програмного забезпечення). В описаннях вакансій вимоги до Software Developer та Software Engineer, а також пропонований рівень заробітної плати приблизно однакові. Тому щодо різниці між цими найменуваннями професій дотепер ведеться дискусія.

Зазвичай до назви професії Developer додається найменування платформи та/ чи мови програмування, що вказує на спеціалізацію фахівця (наприклад Java Developer, .NET Developer).

У сфері розробки програмного забезпечення також прийнято ділити фахівців на категорії, що утворюють певну ієрархію: Trainee, Junior, Middle та Senior. Ці категорії визначаються не рівнем освіти у дипломі, а рівнем сформованості професійних компетентностей та досвідом роботи за спеціалізацією.

Головними вимогами до фахівця категорії Trainee є знання англійської мови, теоретичне знання HTML/CSS, навички оформлення базових документів, розуміння баз даних, API-тестування та практичні навички роботи з ним, а також такі soft skills як гнучкість, стресостійкість, бажання навчатись, комунікабельність, навички роботи в команді.

Junior повинен мати знання основ програмування, вміти читати й самостійно писати програмний код обраною мовою, дотримуватися затверджених стандартів кодування, уміти базово працювати з системою контролю версій, самостійно виконувати

технічні завдання, а також такі soft skills як енергійність, цілеспрямованість, наполегливість, бажання вдосконалювати свої вміння та адекватно реагувати на критику, стресостійкість, навички роботи в команді.

Middle-розробник зобов'язаний самостійно виконувати поставлені перед ним завдання, ставити технічні завдання, вміти працювати у команді, мати управлінські навички, вирішувати конфліктні ситуації, бути стресостійким.

Основними завданнями фахівця категорії Senior є вміння аналізувати бізнес-процеси, проводити системний аналіз, бачити систему в цілому, управляти проєктами, приймати правильні технологічні рішення, тестувати якість роботи програмного забезпечення компанії, тримати контроль над командою, бути відповідальним за діяльність людей у підпорядкуванні, контролювати дедлайни.

Комбінація професійної назви роботи «Developer», спеціалізації та категорії утворює остаточне найменування професії (посади), наприклад Junior Python Developer, Middle Flash Developer.

Таким чином, можемо стверджувати, що фаховий молодший бакалавр з інженерії програмного забезпечення за умов якісно розробленої освітньо-професійної програми та повноцінного формування передбачених нею компетентностей, може зайняти первинну посаду Junior Developer із спеціалізацією, що визначається включеними до освітньо-професійної програми платформами / мовами програмування. За умов набуття досвіду роботи за фахом та активного персонального розвитку фаховий молодший бакалавр здатен перейти до наступної категорії Middle та розширити спеціалізацію. Проте подальше зростання до категорії Senior неможливе без ґрунтовних знань та практичних навичок з аналізу бізнес-процесів, системного аналізу, управління проєктами, управління якістю та управління персоналом. Ці компетентності, зазвичай, можна сформувати здобувши додаткову освіту можливо на наступному першому (бакалаврському) рівні вищої освіти.

Дослідимо питання щодо забезпечення вітчизняними освітніми стандартами:

1. підготовки розробника програмного забезпечення відповідно до запитів сучасного ринку праці;
2. оптимальної відповідності принципу неперервності освіти та концепції навчання впродовж життя.

Для цього проведемо аналіз згаданих стандартів для освітніх рівнів фахового молодшого бакалавра та бакалавра. Перш за все слід зазначити, що ступінь фахового молодшого бакалавра відповідає п'ятому рівню Національної рамки кваліфікацій [6], а ступінь бакалавр – шостому рівню. На основі опису цих рівнів за визначеними дескрипторами (знання, уміння/навички, комунікація, відповідальність та автономія) у законах України «Про фахову передвищу освіту» та «Про вищу освіту» сформульовані інтегральні компетентності для фахівців відповідного рівня:

- «Рівень фахової передвищої освіти передбачає здатність особи вирішувати типові спеціалізовані задачі в окремій галузі професійної діяльності або у процесі навчання, що вимагає застосування положень і методів відповідних наук та може характеризуватися певною невизначеністю умов; відповідальність за результати своєї діяльності; здійснення контролю інших осіб у визначених ситуаціях» [8, ст. 7, част. 1];
- «Перший (бакалаврський) рівень вищої освіти передбачає набуття здобувачами вищої освіти здатності до розв'язування складних спеціалізованих задач у певній галузі професійної діяльності» [9, ст. 5, част. 1].

Це означає, що основна відмінність освітніх рівнів бакалавра та фахового молодшого бакалавра за однойменною спеціальністю полягає у рівні складності задач, які здатен вирішувати відповідний фахівець. Проте чітких критеріїв для оцінки складності задач у сфері програмної інженерії не існує, а методологію Big O для оцінки складності алгоритмів застосовувати для цих цілей не доцільно.

На нашу думку типовими задачами у сфері інженерії програмного забезпечення можна вважати задачі розробки програмного забезпечення, здебільшого, з використанням програмних бібліотек, для автоматизації однотипних процесів, що можуть характеризуватися певною невизначеністю умов. Зазвичай, типові задачі виникають у результаті декомпозиції більш складних задач, що передбачає необхідність наявності у розробника навичок командної роботи. Складні задачі у сфері інженерії програмного забезпечення полягають в організації розробки комплексних програмних рішень на основі процесно-

го та системного аналізу для автоматизації системних процесів, що характеризуються суттєвою невизначеністю умов. Оскільки складні задачі можна представити у вигляді сукупності взаємопов'язаних простих (типових) задач та реалізуються зазвичай певною командою виконавців, то відповідний фахівець повинен мати компетентності щодо розв'язування типових задач (як фаховий молодший бакалавр), а також базові знання із процесного та системного аналізу, проєктного менеджменту, менеджменту якості програмних продуктів та персоналу.

Принагідно слід звернути увагу на той факт, що згідно із Законом України «Про вищу освіту» п'ятому рівню Національної рамки кваліфікацій відповідає також початковий ступінь вищої освіти «молодший бакалавр» та передбачена можливість вступу фахових молодших бакалаврів на навчання за освітньою програмою молодшого бакалавра. Варто звернути увагу на те, що кваліфікації та інтегральні компетентності таких фахівців абсолютно однакові. Зрозуміти таку правову колізію досить складно, проте зазначена проблема потребує додаткового дослідження.

Тепер проаналізуємо затверджений Міністерством освіти та науки Стандарт фахової передвищої освіти за освітньо-професійним ступенем фаховий молодший бакалавр із галузі знань 12 Інформаційні технології, спеціальності 121 Інженерія програмного забезпечення [7].

Даний Стандарт містить інтегральну компетентність та перелік загальних і фахових компетентностей випускника. Як засвідчує раніше наведений аналіз ринку праці, окремі важливі компетентності техніка-програміста не потрапили до Стандарту, тому він потребує удосконалення. Зокрема, у Стандарті відсутні компетентності, пов'язані з керуванням базами даних, організацією комп'ютерних мереж, захисту інформації в комп'ютерних системах, які, на наш погляд, необхідні для повноцінної роботи техніка-програміста у сучасних умовах.

До цікавих висновків приводить також проведений нами порівняльний аналіз компетентностей фахового молодшого бакалавра та бакалавра з інженерії програмного забезпечення із відповідних стандартів.

Наведемо порівняльну таблицю загальних компетентностей.

**Таблиця 1. Порівняння загальних компетентностей фахового молодшого бакалавра і бакалавра спеціальності 121 «Інженерія програмного забезпечення»**

| Загальні компетентності фахового молодшого бакалавра | Загальні компетентності бакалавра |
|---|---|
| ЗК01. Здатність реалізувати свої права і обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні. | К11. Здатність реалізувати свої права і обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні. |
| ЗК02. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя. | К12. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя. |
| ЗК03. Здатність спілкуватися державною мовою як усно, так і письмово. | К03. Здатність спілкуватися державною мовою як усно, так і письмово. |
| ЗК04. Здатність спілкуватися іноземною мовою. | К04. Здатність спілкуватися іноземною мовою як усно, так і письмово. |
| ЗК05. Знання та розуміння предметної області та розуміння професійної діяльності. | |
| ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел. | К06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел. |
| ЗК07. Здатність застосовувати знання у практичних ситуаціях. | К02. Здатність застосовувати знання у практичних ситуаціях. |

Джерело: складена авторами на основі
Стандартів фахової передвищої освіти та вищої освіти галузі знань 12 Інформаційні технології, спеціальність 121 «Інженерія програмного забезпечення» [7, 10]

Як видно з цієї таблиці загальні компетентності фахового молодшого бакалавра є підмножиною загальних компетентностей бакалавра (відсутність відповідника компетентності ЗК05 у стандарті для бакалавра є скоріше певною неузгодженістю, оскільки без знання та розуміння предметної області та розуміння професійної діяльності не може йти мова про сформованість фахових компетентностей випускника).

Також стандарт вищої освіти містить окремі загальні компетентності, які відсутні у стандарті фахової передвищої освіти: К01. Здатність до абстрактного мислення, аналізу та синтезу, К07. Здатність працювати в команді, К08. Здатність діяти на основі етичних міркувань, К09. Прагнення до збереження навколишнього середовища, К10. Здатність діяти соціально відповідально та свідомо [9]. Ці компетентності однозначно слід додати до стандарту фахової передвищої освіти, оскільки компетентності К01, К08, К09, К10 починають формуватися у процесі здобуття повної загальної середньої освіти. Підтвердженням правильності такого висновку можуть слугувати і далі наведені міркування.

Усі відомі авторам освітньо-професійні програми фахової передвищої освіти за спеціальністю 121 Інженерія програмного забезпечення містять освітні компоненти математичного та філософського циклу, а саме ці компоненти формують компетентність К01.

У Кодексі етики інженерів програмного забезпечення задекларовані принципи професійної діяльності, що забезпечують збереження здоров'я, безпеку та добробут суспільства. Розуміння мети та принципів дотримання норм етики при розробці та експлуатації програмного забезпечення є основою для прийняття моральних рішень. А фаховий молодший бакалавр з інженерії програмного забезпечення, відповідно до здобутих компетентностей, займається розробкою та підтримкою програмного забезпечення.

Для забезпечення сталого розвитку суспільства, здорового клімату в колективі надзвичайно важливим є переконання діяти соціально відповідально та свідомо як громадянин та член трудового колективу.

Про компетентність К07 вже було сказано раніше. Крім того Стандарт фахової передвищої освіти містить результат навчання «РН09. Розуміти основні принципи командної роботи при розроб-

ці програмного забезпечення», хоча зв'язані з цим результатом компетентності не зазначені ні серед загальних ні серед фахових компетентностей. Тому, на нашу думку, варто у Стандарт фахової передвищої освіти з інженерії програмного забезпечення до переліку компетентностей додати таку загальну компетентність як здатність працювати в команді.

У результаті реалізації запропонованих змін переліки загальних компетентностей у Стандартах фахової передвищої освіти та вищої освіти першого (бакалаврського) рівня із галузі знань 12 Інформаційні технології, спеціальність 121 Інженерія програмного забезпечення будуть однакові, що є абсолютно логічним. Наразі ж, максимальною можливою різницею між такими переліками є лише три компетентності з дванадцяти.

Наведемо тепер порівняльну таблицю фахових компетентностей з інженерії програмного забезпечення фахового молодшого бакалавра та бакалавра, визначених відповідними стандартами.

**Таблиця 2. Порівняння фахових компетентностей фахового молодшого бакалавра і бакалавра спеціальності 121 «Інженерія програмного забезпечення»**

| Фахові компетентності фахового молодшого бакалавра | Фахові компетентності бакалавра |
|---|---|
| СК01 Здатність алгоритмічно та логічно мислити. | К26. Здатність до алгоритмічного та логічного мислення. |
| СК02. Здатність вдосконалювати знання і навички в галузі інформаційних технологій та усвідомлення важливості навчання протягом усього життя. | К05. Здатність вчитися і оволодівати сучасними знаннями.<br>К22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя. |
| СК03. Здатність застосовувати теоретичні та емпіричні знання для розроблення, тестування, впровадження та супроводу програмного забезпечення. | К25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення. |
| СК04. Здатність дотримуватися стандартів при розробці програмного забезпечення. | К17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу. |

**Закінчення таблиці 2**

| Фахові компетентності фахового молодшого бакалавра | Фахові компетентності бакалавра |
|---|---|
| СК05. Здатність брати участь у визначенні та формулюванні вимог до програмного забезпечення. | К13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. |
| СК06. Здатність брати участь у проектуванні програмного забезпечення. | К14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. |
| СК07. Здатність розробляти модулі і компоненти програмного забезпечення за допомогою типових алгоритмів та інструментів. | К15. Здатність розробляти архітектури, модулі та компоненти програмних систем. |
| СК08. Здатність забезпечувати інформаційну та функціональну безпеку програмного забезпечення. | К18. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки). |
| СК09. Здатність вибирати та використовувати ефективні інструментальні засоби розробки програмного продукту. | К25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення. |
| СК10 Здатність реалізовувати всі етапи життєвого циклу програмного забезпечення. | К23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення. |

Джерело: складена авторами на основі
Стандартів фахової передвищої освіти та вищої освіти галузі знань 12 Інформаційні технології, спеціальність 121 «Інженерія програмного забезпечення» [7, 10]

Аналізуючи наведену таблицю, доходимо висновку, що бакалавр повинен володіти такими ж компетентностями як і фаховий молодший бакалавр з однойменної спеціальності 121 Інженерія програмного забезпечення, а також такими додатковими фаховими компетентностями:

- К16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами;

- К19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних;
- К20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення;
- К21. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності;
- К24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення..

Отже, для отримання бакалаврського ступеня з інженерії програмного забезпечення на основі фахової передвищої освіти за однойменною спеціальністю здобувач має сформувати ще лише п'ять додаткових фахових компетентностей із двадцяти шести.

Наразі, відповідно до Стандарту вищої освіти України: перший (бакалаврський) рівень, галузь знань 12 Інформаційні технології, спеціальність 121 Інженерія програмного забезпечення, при вступі на навчання за освітнім рівнем бакалавра осіб, що мають ступінь фахового молодшого бакалавра за такою ж спеціальністю, заклад вищої освіти має право визнати та перезарахувати не більше ніж 60 кредитів ЄКТС, отриманих за попередньою освітньою програмою фахової передвищої освіти [10], тобто не більше 25% загального обсягу освітньо-професійної програми підготовки бакалавра. Проте, як було доведено вище, навіть за чинними Стандартами відсоток відповідності програмних компетентностей фахового молодшого бакалавра до програмних компетентностей бакалавра значно більший – майже 79% (30 із 38 компетентностей). Крім того заклади фахової передвищої освіти при розробці освітньо-професійної програми можуть виділити до 50% її обсягу на формування окремих додаткових компетентностей не порушуючи Стандарту. Тому, на нашу думку, обмеження обсягів визнання та перезарахування кредитів є недоречним, порушує принцип автономії закладу вищої освіти та суттєво обмежує права здобувача освіти, а тому маємо пропозицію щодо його скасування.

Викладені у цій статті результати досліджень авторів стали важливим підґрунтям для успішної реалізації та удосконалення

освітньо-професійної програми «Інженерія програмного забезпечення» фахової передвищої освіти за спеціальністю 121 Інженерія програмного забезпечення у Фаховому коледжі Закладу вищої освіти «Міжнародний науково-технічний університет імені академіка Юрія Бугая». Коротко опишемо наш досвід проєктування та удосконалення даної ОПП та сподіваємося, що він буде корисним колегам із освітянської спільноти.

Згадана ОПП була розроблена робочою групою із залучення професіоналів-практиків, роботодавців, випускників та студентів Фахового коледжу. Програма зорієнтована на сучасні досягнення у галузі інженерії програмного забезпечення, враховує специфіку роботи у сфері інформаційних технологій, способи і методи розробки програмного забезпечення, парадигми програмування, тестування та супроводу програмних систем різного призначення, а також ґрунтується на загальновідомих наукових результатах, що враховують сучасний стан інформаційних технологій. Концепція освіти у ЗВО «МНТУ» ґрунтується на принципах навчання впродовж життя. Враховуючи можливість використання сучасного мобільного пристрою як повноцінного комп'ютера з функцією постійного доступу до мережі Інтернет та зростання популярності й ефективності використання вебдодатків, основним фокусом ОПП визначено підготовку здобувачів освіти для професійної діяльності у сфері програмування мобільних додатків та Інтернету речей.

Визначена мета ОПП полягає у підготовці майбутніх техніків-програмістів відповідно до запитів сучасного ринку праці, тенденцій розвитку інженерії програмного забезпечення, фахівців, здатних успішно виконувати типові спеціалізовані задачі в галузі інформаційних технологій.

Зважаючи на результати дослідження ринку праці, аналіз загальних та фахових компетентностей, обговорення зі стейкхолдерами до ОПП були включені освітні компоненти «Системи керування базами даних», «Захист інформації в комп'ютерних системах», а також значно збільшено обсяг вивчення навчальної дисципліни «Англійська мова (за професійним спрямуванням)», яка тепер вивчається протягом усього терміну навчання за ОПП.

Особлива увага при підготовці фахового молодшого бакалавра за вказаною ОПП приділяється вивченню мов програмування, щоб розширити спеціалізацію випускника. Так, у межах навчальної дисциплін «Основи програмування» та «Об'єктно-орієнтоване

програмування» вивчається мова програмування C/C++. До переліку вибіркових освітніх компонент професійної підготовки додані навчальні дисципліни «Основи програмування мовою Python» та «Основи програмування мовою JavaScript».

Перелік вибіркових навчальних дисциплін актуалізується кожного року на основі анкетування здобувачів освіти та обговорення із зацікавленими сторонами, представниками професійної спільноти. При останньому оновленні до переліку вибіркових навчальних дисциплін були додані актуальні навчальні дисципліни «Організація комп'ютерних мереж», «Конструювання програмного забезпечення», «Людино-машинна взаємодія».

При формуванні змісту освітніх компонентів також звертається увага на формування фахових компетентностей, що пов'язані з використанням сучасних технологій і цифрових інструментів – digital skills. Для IT-фахівця це вміння застосовувати спеціальні програми, інструменти та технології у своїй роботі, такі як аналітичні інструменти, хмарні обчислення в Microsoft/Google/Apple, системи штучного інтелекту. Ці напрями є пріоритетними для підвищення кваліфікації викладачів фахових навчальних дисциплін. Так, викладачі циклової комісії інженерії програмного забезпечення та фізико-математичних дисциплін пройшли онлайн-курс «Основи AI» та здобули практичні навички для ефективного застосування штучного інтелекту в освітньому процесі. Також у процесі вивчення дисципліни «Офісні інформаційні технології» студенти знайомляться з технологіями хмарних обчислень, правовими засадами їх застосовування, можливостями офісного пакету Google, порядком отримання, використання та безпечного зберігання електронного підпису.

ОПП містить освітні компоненти, що передбачають проведення досліджень. Це виконання курсової та кваліфікаційної роботи. Тематика робіт розробляється із врахуванням їх актуальності. Досвід публічного захисту курсових робіт засвідчив зацікавленість та відповідальне ставлення здобувачів освіти до виконання робіт, ефективність процедури захисту для формування багатьох soft skills. Варто зазначити, що подальшу апробацію своїх наукових напрацювань у процесі розробки курсової роботи здобувачі фахової передвищої освіти здійснили шляхом представлення їх на Всеукраїнській науково-практичній конференції здобувачів вищої освіти і молодих вчених.

Значна увага в освітньому процесі за ОПП приділяється формуванню та розвитку soft skills – навичок, що не є специфічними для конкретної професії, але відображають особисті якості та здібності людини у спілкуванні з оточуючими. Рівень сформованості таких компетентностей має істотний вплив на конкурсний відбір при працевлаштуванні, формування власної репутації та кар'єрне зростання. Найактуальнішими soft skills для розробників програмного забезпечення є навички роботи в команді, відповідальність, самодисципліна, впевненість у собі, критичне мислення, лідерство; креативність, конструктивне сприйняття та аналіз критичних зауважень до виконаних завдань, вміння встановлювати пріоритети в завданнях. Такі навички формуються зазвичай спеціальними методами навчання, позааудиторними виховними заходами, участю у дослідженнях та наукових конференціях, особистим прикладом викладачів, а також спеціальною вибірковою навчальною дисципліною «Soft skills і способи їх розвитку» та підсилюються під час проходження практики.

Практична підготовки фахівців є надзвичайно важливою складовою частиною освітньої професійної програми, якою передбачено три види практик: навчальна практика «Вступ до спеціальності», виробнича практика та переддипломна практика.

Зокрема, навчальна практика проводиться для закріплення здобувачами освіти теоретичних знань, ознайомлення зі специфікою майбутньої спеціальності, отримання первинних професійних умінь і навичок, ознайомлення із типовими завданнями, які вирішують фахівці за обраним фахом; попередньої пропедевтичної орієнтації на спектр виробничих функцій фахівця, умінь та компетентностей, необхідних для їх реалізації. Досвід проведення навчальної практики показав, що студенти, ознайомившись з інструментами github (один з найбільших вебсервісів для спільної розробки програмного забезпечення) та git (розподілена система контролю версій, яка дозволяє відстежувати історію розробки програмного забезпечення і спільно працювати над складними проєктами), успішно виконали завдання з програмування, наближені до реалізації реальних професійних задач та пройшли весь процес командної розробки, що сприяло формуванню здатності працювати в команді та підсиленні програмного результату навчання «Розуміти основні принципи командної роботи при розробці програмного забезпечення».

У межах навчальної практики студенти для ознайомлення з професією QA-тестувальник переглядають освітній серіал «QA-тестувальник», створений з ініціативи Міністерства цифрової трансформації для національної онлайн-платформи з цифрової грамотності Дія.Освіта [3]. Здобувачі освіти дізнаються чим займається і які задачі виконує тестувальник, які якості важливо мати тестувальнику, якими навичками володіти, якими інструментами та програмами потрібно користуватись, які можливості розвитку у цій професії. Також студенти з використанням інтерактивного симулятора можуть віртуально відвідати робоче місце тестувальника та виконати типові задачі тестувальників програмного забезпечення.

Завданнями виробничої практики є вивчення організації і етапів розробки програмного продукту, набуття практичних навичок програмування, проєктування та реалізації вебпроєктів (блоків «Front-end» та «Back-end»), самостійного вирішення технічних задач на базі сучасних комп'ютеризованих систем, ознайомлення з сучасними технологічними процесами розробки, впровадження та налагодження програмного продукту, з сучасним апаратним та програмним забезпеченням, набуття умінь організаторської роботи за спеціальністю. Для реалізації деяких проєктів студенти об'єднувалися у команди, що також підсилило компетентності командної роботи.

Дослідження ринку праці показали, що роботодавці враховують і різноманітні сертифікати, що отримали претенденти на заняття відповідної посади у рамках формальної чи неформальної освіти. Тому циклова комісія з інженерії програмного забезпечення та фізико-математичних дисциплін, яка є відповідальною за реалізацію ОПП всіляко вітає проходження здобувачами сертифікатних курсів.

Враховуючи те, що безпекова складова переходу суспільного життя в кіберпростір, поява та розвиток Інтернету речей стають ще більш актуальними, здобувачі освіти під керівництвом викладача пройшли курс «Вступ до кібербезпеки», розміщений на платформі мережевої академії CISCO – міжнародної освітньої програми в галузі IT та кібербезпеки. У результаті опанування курсу студенти отримали початкові знання з кібербезпеки; уточнили її вплив на соціальні та виробничі процеси у соціальних групах; узнали про найбільш поширені загрози, атаки та вразливості; зро-

зуміли як компанії захищають свою діяльність від атак; дізналися чому сегмент кібербезпеки продовжує зростати. Як додатковий бонус слід відзначити отримання студентами відповідних сертифікатів академії CISCO.

Варто зазначити, що у результаті впровадження вищезазначених інструментів у освітній процес Фаховий коледж Закладу вищої освіти «Міжнародний науково-технічний університет імені академіка Юрія Бугая» зайняв досить високе місце у рейтингу приватних закладів вищої світи за результатами вступної кампанії у 2022 року за показником «Зараховано на контракт» [2].

**Висновки та пропозиції.** У результаті проведених досліджень уточнено первинні посади, що може займати технік-програміст на сучасному ринку праці,

Сформульовано перелік типових спеціалізованих задач у галузі інформаційних технологій, які здатен виконувати технік-програміст.

Проведено порівняльний аналіз загальних та фахових компетентностей фахового молодшого бакалавра з інженерії програмного забезпечення та бакалавра з інженерії програмного забезпечення, визначених відповідними стандартами.

Означено компетентності, що є необхідними для фахових молодших бакалаврів з інженерії програмного забезпечення, але не були враховані у Стандарті такі як: здатність працювати в команді, здатність діяти на основі етичних міркувань, прагнення до збереження навколишнього середовища, здатність діяти соціально відповідально та свідомо.

Встановлено, що передбачені відповідними Стандартами програмні компетентності фахового молодшого бакалавра на 79% співпадають з програмними компетентностями бакалавра за однойменною спеціальністю 121 Інженерія програмного забезпечення. У зв'язку з цим запропоновано скасувати у Стандарті вищої освіти України для першого (бакалаврського) рівня, галузі знань 12 Інформаційні технології, спеціальності 121 Інженерія програмного забезпечення обмеження обсягу кредитів ЄКТС, що можуть бути визнані та перезараховані для вступників на підставі диплому фахового молодшого бакалавра з такої ж спеціальності. Наразі такі обмеження становлять 60 кредитів ЄКТС.

З досвіду провадження підготовки фахових молодших бакалаврів у Фаховому коледжі ЗВО «МНТУ» запропоновано практичні інструменти, спрямовані на посилення професійних компетентностей техніка-програміста.



**ЛІТЕРАТУРА**

1. Варава І. П. Формування готовності до професійної діяльності майбутніх техніків-програмістів як сучасна психолого-педагогічна проблема. *Вісник Національного авіаційного університету, Серія: Педагогіка. Психологія.* 2021. Вип. 1 (18). С. 18–30. https://doi.org/10.18372/2411-264X.18.15470 (дата звернення: 02.06.2023).

2. Вступ. Освіта.ua: Рейтинг вишів. URL: https://osvita.ua/vnz/rating/vstup-osvita/59065/ (дата звернення: 01.06.2023).

3. Дія. Освіта. URL: https://osvita.diia.gov.ua/ (дата звернення: 02.06.2023).

4. Класифікатор професій ДК 003:2010 від 28.07.2010 № 327. URL: https://zakon.rada.gov.ua/rada/show/va327609-10#Text (дата звернення: 01.06.2023).

5. Кого потребує IT індустрія у 2023 році. URL: https://itukraine.org.ua/who-does-the-it-industry-need-in-2023.html (дата звернення: 01.06.2023).

6. Про затвердження Національної рамки кваліфікацій : Постанова Кабінету Міністрів України від 23.11.2011 № 1341. URL: https://zakon.rada.gov.ua/laws/show/1341-2011-%D0%BF#Text (дата звернення: 01.06.2023).

7. Про затвердження стандарту фахової передвищої освіти зі спеціальності 121 Інженерія програмного забезпечення галузі знань 12 Інформаційні технології освітньо-професійного ступеня «фаховий молодший бакалавр» : Наказ Міністерства освіти і науки України від 21.09.2021 р. № 1006. URL: https://mon.gov.ua/ua/npa/pro-zatverdzhennya-standartu-fahovoyi-peredvishoyi-osviti-zi-specialnosti-121-inzheneriya-programnogo-zabezpechennya-galuzi-znan-12-informacijni-tehnologiyi-osvitno-profesijnogo-stupenya-fahovij-molodshij-bakalavr (дата звернення: 22.05.2023).

8. Про фахову передвищу освіту: Закон України від 06.06.2019. № 2745-VIII. Редакція від 23.03.2023. URL: https://zakon.rada.gov.ua/laws/show/2745-19#Text (дата звернення: 02.06.2023).

9. Про вищу освіту: Закон України від 01.07.2014. № 1556-VII. Редакція від 28.05.2023 URL: https://zakon.rada.gov.ua/laws/show/2745-19#Text (дата звернення: 02.06.2023).

10. Стандарт вищої освіти України: перший (бакалаврський) рівень, галузь знань 12 Інформаційні технології, спеціальність 121 Інженерія програмного забезпечення. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/12/21/121-inzhener.programn.zabezp.bakalavr-1.pdf (дата звернення: 02.06.2023).

11. Топузов О.М., Малихін О.В., Ярмольчук Т.М. Модель стратегії формування готовності майбутніх фахівців з інформаційних технологій до професійної діяльності / *Інформаційні технології і засоби навчання*, 2020, Том 77, №3. С. 205 – 222. URL: https://journal.iitta.gov.ua/index.php/itlt/issue/view/106 (дата звернення: 02.06.2023).

12. Любарець В., Любима А. Становлення конкурентоспроможності майбутніх фахових молодших бакалаврів з інженерії програмного забезпечення / *Вища освіта України*, 2022, № 1-2. С. 38-43. URL: https://journals.udu.kyiv.ua/index.php/vou/article/view/37/28 (дата звернення: 07.06.2023).

13. Радкевич В.О., Лузан П.Г., Пащенко Т.М. Фахова передвища освіта: аналітичний огляд ефективності / *Вісник НАПН України*, 2022, Том 4, №2. С. 1–12. URL: https://visnyk.naps.gov.ua/index.php/journal/article/view/295/360 (дата звернення: 07.06.2023).

## REFERENCES

1. Varava I. P. Formuvannia hotovnosti do profesiinoi diialnosti maibutnikh tekhnikiv-prohramistiv yak suchasna psykholoho-pedahohichna problema. *Visnyk Natsionalnoho aviatsiinoho universytetu, Seriia: Pedahohika. Psykholohiia*. 2021. Vyp. 1 (18). P. 18-30. https://doi.org/10.18372/2411-264X.18.15470 (data zvernennia: 02.06.2023).

2. Vstup. Osvita.ua: Reitynh vyshiv. URL: https://osvita.ua/vnz/rating/vstup-osvita/59065 (data zvernennia: 01.06.2023).

3. Diia. Osvita. URL: https://osvita.diia.gov.ua (data zvernennia: 02.06.2023).

4. Klasyfikator profesii DK 003:2010 vid 28.07.2010 № 327. URL: https://zakon.rada.gov.ua/rada/show/va327609-10#Text (data zvernennia: 01.06.2023).

5. Koho potrebuie IT industriia u 2023 rotsi. URL: https://itukraine.org.ua/who-does-the-it-industry-need-in-2023.html (data zvernennia: 01.06.2023).

6. Pro zatverdzhennia Natsionalnoi ramky kvalifikatsii : Postanova Kabinetu Ministriv Ukrainy vid 23.11.2011 № 1341. URL: https://zakon.rada.gov.ua/laws/show/1341-2011-%D0%BF#Text (data zvernennia: 01.06.2023).

7. Pro zatverdzhennia standartu fakhovoi peredvyshchoi osvity zi spetsialnosti 121 Inzheneriia prohramnoho zabezpechennia haluzi znan

12 Informatsiini tekhnolohii osvitno-profesiinoho stupenia «fakhovyi molodshyi bakalavr» : Nakaz Ministerstva osvity i nauky Ukrainy vid 21.09.2021 r. № 1006. URL: https://mon.gov.ua/ua/npa/pro-zatverdzhennya-standartu-fahovoyi-peredvishoyi-osviti-zi-specialnosti-121-inzheneriya-programnogo-zabezpechennya-galuzi-znan-12-informacijni-tehnologiyi-osvitno-profesijnogo-stupenya-fahovij-molodshij-bakalavr (data zvernennia: 22.05.2023).

8. Pro fakhovu peredvyshchu osvitu : Zakon Ukrainy vid» vid 06.06.2019. URL: https://zakon.rada.gov.ua/rada/show/va327609-10#Text (data zvernennia: 01.06.2023).

9. Pro vyshchu osvitu: Zakon Ukrainy vid 01.07.2014. № 1556-VII. Redaktsiia vid 28.05.2023 URL: https://zakon.rada.gov.ua/laws/show/2745-19#Text (data zvernennia: 02.06.2023).

10. Standart vyshchoi osvity Ukrainy: pershyi (bakalavrskyi) riven, haluz znan 12 Informatsiini tekhnolohii, spetsialnist 121 Inzheneriia prohramnoho zabezpechennia. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/12/21/121-inzhener.programn.zabezp.bakalavr-1.pdf (data zvernennia: 02.06.2023).

11. Topuzov O.M., Malykhin O.V., Yarmolchuk T.M. Model stratehii formuvannia hotovnosti maibutnikh fakhivtsiv z informatsiinykh tekhnolohii do profesiinoi diialnosti / *Informatsiini tekhnolohii i zasoby navchannia*, 2020, Tom 77, №3. P. 205-222 (data zvernennia: 02.06.2023).

12. Liubarets V., Liubyma A. Stanovlennia konkurentospromozhnosti maibutnikh fakhovykh molodshykh bakalavriv z inzhenerii prohramnoho zabezpechennia / *Vyshcha osvita Ukrainy*, 2022, № 1-2. S. 38-43. URL: https://journals.udu.kyiv.ua/index.php/vou/article/view/37/28 (data zvernennia: 07.06.2023).

13. Radkevych V.O., Luzan P.H., Pashchenko T.M. Fakhova peredvyshcha osvita: analitychnyi ohliad efektyvnosti / *Visnyk NAPN Ukrainy*, 2022, Tom 4, №2. S. 1-12. URL: https://visnyk.naps.gov.ua/index.php/journal/article/view/295/360 (data zvernennia: 07.06.2023).

**СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 02.12.2023**

*УДК 004.02, 004.455.2*
*DOI: https://doi.org/10.53920/ITS-2023-2-2*

**Костянтин Олександрович ТКАЧЕНКО,**
кандидат економічних наук, доцент,
доцент кафедри інформаційних технологій,
Державний університет інфраструктури та технологій
ORCID ID: 0000-0003-0549-3396
**Владислав Юрійович ЯКИМЕНКО,**
магістрант кафедри інформаційних технологій,
Державний університет інфраструктури та технологій
ORCID ID: 0009-0007-4752-1259

# ДЕЯКІ АСПЕКТИ ВИКОРИСТАННЯ АСИНХРОННОЇ КОМУНІКАЦІЇ У МІКРОСЕРВІСНІЙ АРХІТЕКТУРІ

***У великих мікросервісних системах асинхронна комунікація є важливим інструментом для забезпечення гнучкості, масштабованості та стійкості. Така комунікація дозволяє мікросервісам працювати разом без необхідності негайної (постійної) взаємодії та сприяє створенню архітектури, що є відмінної від так званих «монолітних» систем. Мікросервіси, використовуючи асинхронну комунікацію, можуть функціонувати незалежно один від одного. Кожен мікросервіс може взаємодіяти з іншими сервісами через асинхронні події або повідомлення, незалежно від рівнів їхньої доступності або стану, в якому вони знаходяться.***

***Метою статті є дослідження та аналіз деяких аспектів використання асинхронної комунікації в мікросервісній архітектурі для визначення оптимальних підходів та рекомендацій щодо її впровадження в сучасних інформаційних системах. В роботі розглянуто можливості підвищення масштабованості, надійності та продуктивності мікросервісних додатків за допомогою асинхронних підходів до здійснення комунікації між сервісами. Проведено аналіз відповідних прикладів практичного застосування запропонованого підходу та підтверджено важливість і актуальність проблем, що розглядаються, в сучасному програмному інжинірингу.***

***Використання результатів проведеного аналізу та розроблених сервісів буде корисне для різних категорій користувачів – розробни-***

***ків відповідного програмного забезпечення (зокрема, програмістів і архітекторів програмного забезпечення). Вони можуть використовувати у своїх проєктах для оптимізації комунікації між сервісами запропонований авторами підхід. Для менеджерів та технічних лідерів програмних проєктів отримані результати проведеного аналізу та запропонований підхід сприятимуть кращому розумінню того, як асинхронна комунікація може вплинути на продуктивність і стабільність мікросервісних додатків. Ці аспекти вказаними категоріями користувачів та розробників можуть враховуватися при прийнятті рішень щодо архітектури своїх проєктів. Запропонований підхід має все необхідне для подальшого розвитку і впровадження в сфері мікросервісної архітектури та асинхронної комунікації.***

***Ключові слова:*** *асинхронна комунікація, мікросервіси, архітектура програмних застосунків, брокери повідомлень, високонавантажені системи, хмарні обчислення, хмарні технології.*

**Kostiantyn TKACHENKO**
PhD of economical sciences, associate professor,
associate professor at the department
of information technologies,
State University of Infrastructure and Technology
**Vladyslav YAKYMENKO**
Undergraduate at the department of information technologies,
State University of Infrastructure and Technology

## SOME ASPECTS OF USING ASYNCHRONOUS COMMUNICATION IN MICROSERVICES ARCHITECTURE

***In large microservices systems, asynchronous communication is an important tool to ensure flexibility, scalability, and resilience. Such communication allows microservices to work together without the need for immediate (constant) interaction and contributes to the creation of an architecture that differs from so-called «monolithic» systems. Microservices, using asynchronous communication, can operate independently of each other. Each microservice can interact with other services through asynchronous events or messages, regardless of their availability levels or states.***

***The purpose of this article is to explore and analyze the features and advantages of using asynchronous communication in microservices***

***architecture to determine optimal approaches and recommendations for its implementation in modern information systems. The paper examines the possibilities of improving the scalability, reliability, and performance of microservices applications through asynchronous communication approaches. An analysis of relevant practical examples of the proposed approach is conducted, confirming the importance and relevance of the issues under consideration in modern software engineering.***

***The use of the results of the analysis and the developed services will be beneficial for various categories of users, including developers of corresponding software (in particular, programmers and software architects). They can incorporate the approach proposed by the authors into their projects to optimize communication between services. For managers and technical leaders of software projects, the results of the analysis and the proposed approach will contribute to a better understanding of how asynchronous communication can affect the performance and stability of microservices applications. These aspects can be taken into account when making decisions regarding the architecture of their projects. The proposed approach has everything necessary for further development and implementation in the field of microservices architecture and asynchronous communication.***

***Keywords:*** *asynchronous communication, microservices, software architecture, message brokers, high-performance systems, cloud computing, cloud technologies.*

**Постановка проблеми.** В мікросервісній архітектурі [1], де програмні додатки поділяються на невеликі незалежні сервіси, асинхронна комунікація [2] стає необхідною складовою при вирішенні цілої низки важливих і актуальних проблем та практичних завдань, усунення наслідків викликів, що виникають.

В цій статті розглянуто та обгрунтовано необхідність використання асинхронної комунікації у мікросервісній архітектурі та здійснення аналізу існуючих проблем, з якими можна стикнутися при її впровадженні.

Актуальність асинхронної комунікації в мікросервісній архітектурі не викликає сумнівів, а її необхідність обумовлена, зокрема, такими факторами як:

- необхідність забезпечення незалежності мікросервісів (кожен сервіс повинен функціонувати самостійно);

- надання можливості сервісам взаємодіяти без блокування та залежностей;
- необхідність підвищення рівнів масштабованості та продуктивності, щоб сервіси мали можливість обробляти події асинхронно та реагувати на них відповідно до навантаження;
- необхідність оптимізації роботи з великим обсягом даних (зменшувати, зокрема, час очікування результатів).

При використанні асинхронної комунікації в мікросервісній архітектурі можуть виникати, зокрема такі проблеми, як:

- складність налагодження та налаштування, оскільки події відбуваються асинхронно і важко відстежити послідовність подій;
- контроль за станом системи та забезпечення консистентності [3], оскільки асинхронні події можуть виникати в непередбачуваний момент;
- складність управління повідомленнями, коли у великих системах асинхронна комунікація може призвести до значного збільшення кількості повідомлень, що потребує ефективних засобів відповідного управління та моніторингу цими повідомленнями;
- необхідність додаткових інфраструктурних рішень, коли для використання асинхронної комунікації слід розгорнути, зокрема, такі додаткові інфраструктурні компоненти, як брокери повідомлень [4] або системи черг [5].

Таким чином дослідження та аналіз використання асинхронної комунікації в мікросервісній архітектурі є актуальною проблемою, вирішення якої може призвести до розробки відповідних оптимальних підходів і рекомендацій щодо подолання цієї проблеми та забезпечення ефективної роботи мікросервісних систем.

**Аналіз останніх досліджень і публікацій.** Дослідження в сфері асинхронних комунікацій та мікросервісної архітектури належать до перспективних напрямків розвитку сучасного програмного інжинірингу, оскільки мікросервіси набувають популярності, і важливо розуміти, як вирішувати проблеми, пов'язані з асинхронною комунікацією.

В [6] досліджуються принципи та шаблони реалізації подійно-орієнтованих (ситуаційно-орієнтованих) мікросервісів, що використовують асинхронні комунікації.

Визначення ключових проблем, з якими можна стикнутися при впровадженні асинхронної комунікації в мікросервісній архітектурі, включаючи питання щодо керування станом, налагодження і безпеки, розглянуто в [7]. В цій роботі запропоновані й шляхи щодо вирішення визначених проблем та досягнення успішної імплементації асинхронної комунікації в мікросервісній архітектурі.

В [8] розглядається конкретний технічний стек реалізації подійно-орієнтованих мікросервісів з використанням Spring Boot (фреймворку на основі Java з відкритим кодом, розробленого компанією Pivotal Software) [9] і Apache Kafka (розподіленого сховищі подій і платформи для їх багатопотокової обробки) [10].

Різноманітні шаблони для мікросервісної архітектури, включаючи асинхронну комунікацію, розглянуто в [11]. Надані в роботі приклади реалізації вказаних шаблонів мовою програмування Java сприяють більш глибокому розумінню принципів та паттернів для мікросервісних додатків

Проведений аналіз обумовив переконаність та впевненість авторів у важливості асинхронної комунікації у мікросервісній архітектурі та обгрунтував необхідність подальших досліджень та розробки оптимальних практичних рішень для вирішення викликів запропонованого підходу.

**Мета статті.** Метою цієї наукової статті є проведення аналізу та дослідження проблем, пов'язаних з використанням асинхронної комунікації в мікросервісній архітектурі, визначення оптимальних стратегій і підходів для вирішення цих проблем та розробки програмного забезпечення.

Досягнення цієї мети забезпечується вирішенням, зокрема, наступних завдань:

- проведення аналізу проблем, пов'язаних з використанням асинхронної комунікації в мікросервісній архітектурі, визначення переваг та недоліків використання такої комунікації;
- визначення класів задач, які можуть бути вирішені більш ефективно за допомогою асинхронної комунікації в мікросервісній архітектурі;

- дослідження інструментарію, програмного забезпечення та технологій, адекватних потребам та можливостям асинхронної комунікації в мікросервісній архітектурі, що дозволяють збирати, аналізувати та використовувати відповідні дані для підвищення ефективності мікросервісних додатків;
- формування практичних рекомендацій (для розробників та архітекторів програмного забезпечення) щодо ефективного впровадження асинхронної комунікації в мікросервісних додатках

Мета і завдання статті спрямовані на просування інноваційних підходів та технологій, які можуть підтримати стійкий розвиток області, що розглядається, в сучасному програмному інжинірингу.

**Виклад основного матеріалу дослідження.** Мікросервісна архітектура виникла як реакція на недоліки монолітних додатків і стала популярною завдяки своїй гнучкості та здатності розширюватися.

Однак разом з перевагами мікросервісів з'явилися виклики, пов'язані з взаємодією між ними. Асинхронна комунікація та брокери повідомлень виникли для вирішення цих викликів і забезпечення надійної та ефективної комунікації між мікросервісами.

Мікросервіси мають функціонувати незалежно один від одного. Асинхронна комунікація дозволяє їм взаємодіяти без необхідності блокування та очікування відповідей в реальному часі. Це забезпечує вищу ступінь незалежності та розділення обов'язків між сервісами.

У великих системах з багатьма сервісами синхронна комунікація може призвести до зайвого очікування відповідей. Асинхронні повідомлення дозволяють сервісам продовжувати роботу, не очікуючи негайної відповіді. Мікросервіси часто функціонують в розподіленому середовищі, де доволі часто можуть виникати затримки та відмови.

Асинхронна комунікація допомагає забезпечити надійну взаємодію між сервісами, незалежно від їх доступності чи стану. На рис. 1 зображено типову архітектуру мікросервісного застосунку, який використовує асинхронний тип комунікації між своїми компонентами.

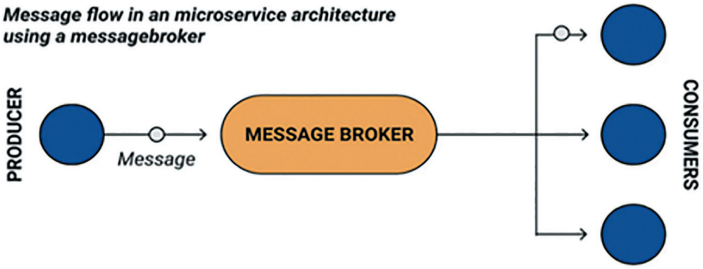


**Рис. 1. Приклад використання брокера повідомлень в мікросервісній архітектурі**

Джерело: [12]

Серед можливих сценаріїв використання для асинхронної комунікації та брокерів повідомлень слід виділити, зокрема такі, як:

*1. Обробка замовлень та транзакцій*. В онлайн-торгівлі асинхронна комунікація може використовуватися для обробки замовлень, підтвердження оплати та оновлення статусу доставки. Кожен етап операції може бути окремим сервісом, які взаємодіють асинхронно, щоб забезпечити швидку та надійну обробку інформації.

*2. Обробка подій користувачів*. Соціальні мережі та додатки задля спільної роботи використовують асинхронну комунікацію для реагування на дії користувачів. Наприклад, коли користувач залишає коментар, система може відправити повідомлення іншим користувачам без очікування відповіді.

*3. Збір та обробка даних в реальному часі*. Аналітичні системи можуть використовувати асинхронну комунікацію для збору та обробки даних в реальному часі. Дані можуть надходити з різних джерел і асинхронно оброблюватися для створення відповідних аналітичних звітів.

*4. Управління конфігурацією*. У системах, де потрібно розподіляти конфігурацію та оновлення параметрів, асинхронна комунікація може допомогти впроваджувати зміни без перезавантаження сервісів.

*5. Інтеграція зі сторонніми сервісами*. При інтеграції зі сторонніми сервісами, такими як платіжні системи чи поштові сервіси, асинхронна комунікація дозволяє ефективно взаємодіяти зі сторонніми API [13] та обробляти відповіді асинхронно.

Розглянемо підходи до реалізації асинхронної комунікації в мікросервісній архітектурі:

*1. Тип комунікації «Сповіщення» (Publish-Subscribe, Pub-Sub)* [14] є одним із найпоширеніших методів асинхронної комунікації в мікросервісній архітектурі. Він полягає у тому, що мікросервіси публікують події чи повідомлення в централізованому брокері повідомлень, а інші сервіси можуть підписатися на ці події та отримувати їх для подальшої обробки. Типову архітектуру мікросервісних застосунків використовуючих Pub-Sub зображено на рис. 2. Розглянемо більш детально «Сповіщення» у мікросервісній архітектурі, звертаючи увагу, зокрема, на те, що:

- мікросервіси, що публікують в брокері повідомлень власну наявну інформацію та події, якими можуть користуватися іншими сервісами. Наприклад, сервіс, що відповідає за обробку замовлень, може публікувати подію про нове замовлення. Інші мікросервіси можуть підписатися на ті типи подій, які мають для них інтерес. Наприклад, сервіс оповіщень може підписатися на події про нові замовлення, щоб надсилати сповіщення своїм користувачам. На один і той же тип події може бути підписано декілька різних сервісів (наприклад, кілька сервісів можуть підписатися на подію про оновлення профілю користувача, і реакція в кожному сервісі може бути різною);
- важливим у запропонованому підході є наявність централізованого брокеру повідомлень, наприклад, таких як Apache Kafka, RabbitMQ [15]. Брокер відповідає за збереження подій та їх розсилку, що забезпечує надійну доставку подій та розділення між виробниками (сервіси, що публікують події) і споживачами (сервіси, що підписані на події);
- є можливість більш чітко розділити обов'язки між сервісами (сервіси, які публікують події, спеціалізуються на своїх функціях, а сервіси-споживачі реагують на події, до яких в них є інтерес, без необхідності знати деталі роботи інших сервісів), що сприяє створенню реалізацій в реальному часі, бо події можуть бути оброблені майже миттєво після їх публікації.


**Рис. 2. Приклад архітектури Publish-Subscribe**

Джерело: [16]

*2. Тип комунікації через HTTP/REST API* [17] з асинхронними запитами дозволяє мікросервісам взаємодіяти асинхронно, використовуючи протокол HTTP/REST. Такий тип комунікації, також відомий як Webhook [18], зображено на рис. 3. Цей підхід відрізняється від звичайної синхронної комунікації через REST тим, що він дозволяє мікросервісам надсилати запити та отримувати відповіді асинхронно, коли це необхідно.

У цьому підході мікросервіси можуть надсилати HTTP-запити до інших сервісів, не отримуючи відповіді відразу. Замість блокуючого очікування відповіді, сервіс, що надсилає запит, може продовжувати свою роботу без очікування результату. Це особливо корисно, коли обробка запиту може зайняти багато часу. Сервіс, який отримує такий запит, може обробляти його відокремлено та відправляти відповідь тоді, коли вона готова. Отримувач може підписатися на отримання відповіді та отримати її, коли вона буде доступною.

Такий підхід особливо корисний для операцій, які можуть тривати довго, наприклад, завантаження великих файлів або складні обчислення. Сервіс може прийняти асинхронний запит, розпочати операцію і повідомити про її завершення тоді, коли вона вже виконана. В асинхронних запитах стан запиту може бути збережений для його подальшого використання. Наприклад, сервіс може надіслати асинхронний запит на обробку великої кількості даних, а потім періодично запитувати стан обробки і отримувати відповіді.

Цей тип асинхронної комунікації може бути використаний для реалізації систем сповіщень та подій. Мікросервіси можуть надсилати асинхронні запити для підписки на певні події та отримувати сповіщення про їх настання.

Відкладені запити допомагають уникнути блокування і звільняють в системі ресурси для виконання інших завдань. Це може підвищити продуктивність та ефективність системи. Використання асинхронних запитів дозволяє системі бути більш гнучкою і оперативно реагувати на зміни та навантаження.

Webhook сприяє ефективній взаємодії мікросервісів, особливо при обробці довгих операцій або некритичності точного часу, що надається на формування відповіді. Використання Webhook є доцільним при створенні гнучких і реактивних інформаційних систем.

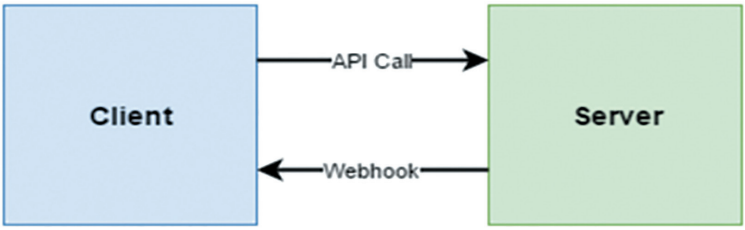


**Рис. 3. Приклад HTTP/REST API з асинхронними запитами**

Джерело: [18]

*3. WebSocket* [19] – протокол для асинхронної двосторонньої комунікації між клієнтом і сервером через одне з'єднання TCP/IP [20]. Він розроблений спеціально для вебдодатків, які потребують низького рівня затримки та високого рівня ефективності при обміні даними в режимі реального часу.

Для обміну даними між клієнтом та сервером через WebSocket слід спочатку встановити початкове з'єднання та виконують handshake [21] (так зване «рукостискання»). Handshake –процес взаємного підтвердження підтримки WebSocket і обрання версії протоколу, побудованого на принципі постійного з'єднання, що відрізняє його від традиційного HTTP, де кожен запит відкриває та закриває нове з'єднання. Постійне з'єднання дозволяє клієнту і серверу спілкуватися в реальному часі без затримок, шо пов'язані

з відкриттям/закриттям з'єднань. Діаграма послідовності операцій для Websocket зображена на рис. 4.

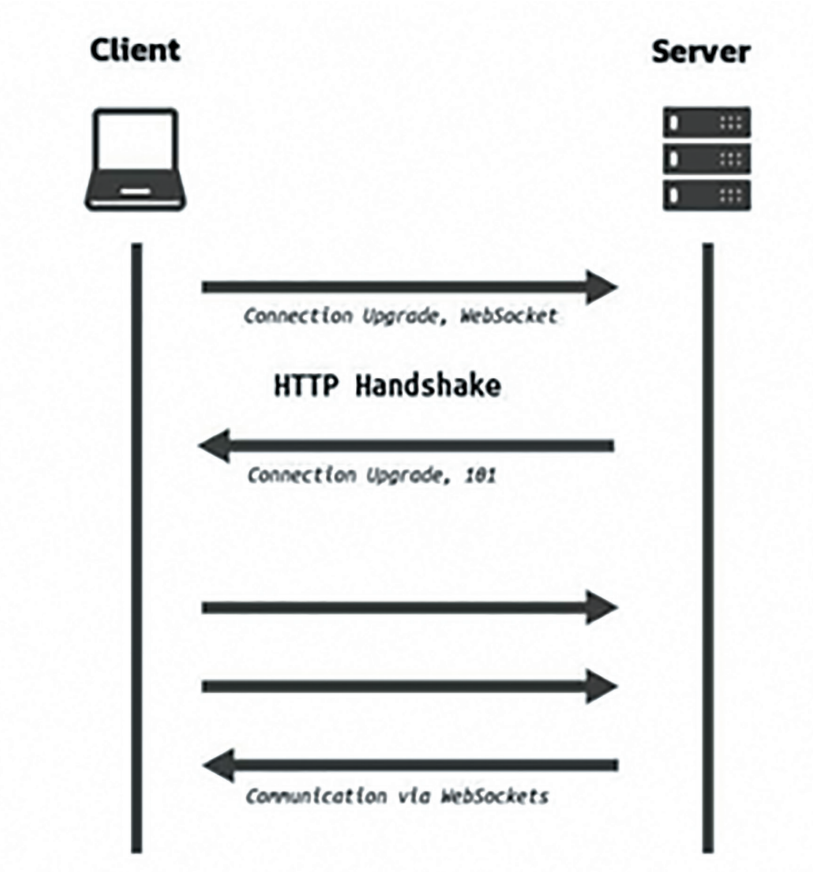


**Рис. 4. Приклад використання WebSocket**

Джерело: [19]

WebSocket, зокрема:

- підтримує асинхронний обмін даними в обидва напрямки (клієнт -> сервер та сервер -> клієнт), тобто і клієнт, і сервер можуть ініціювати відправку даних без очікування запиту від іншої сторони;

- підтримує передачу як бінарних, так і текстових даних, що робить його універсальним для різноманітних застосувань (від передачі текстових повідомлень до передачі зображень, відео та інших бінарних даних);
- має менший обсяг налагодження порівняно з HTTP через зменшення накладних витрат використання заголовків та оптимізований протокол обміну даними.

WebSocket ідеально підходить для додатків, які потребують режиму реального часу, таких як онлайн-чати, сповіщення, моніторинг. Він дозволяє миттєво відправляти та отримувати оновлення без необхідності постійних запитів. Для його використання у вебдодатках розробники можуть користуватися спеціалізованими бібліотеками та фреймворками, такими як Socket.IO, Spring WebSockets, або WebSocket API в браузері.

Важливим при виборі способу комунікації між мікросервісами є популярність підходу, яка дозволяє використовувати широкий спектр сучасних бібліотек та фреймворків. Все це сприяє прискоренню процесу розробки програмного продукту, що є важливим для підприємств, основним джерелом доходу яких є відповідні інформаційні системи.

Наведемо приклади різних підходів до використання популярної технології асинхронної комунікації в мікросервісній архітектурі – Apache Kafka.

На рис. 5 наведено приклад імперативного підходу до взаємодії з брокером повідомлень, з використанням нативного клієнту для Kafka.

```
while (running) {
  ConsumerRecords<K, V> records = consumer.poll(Long.MAX_VALUE);
  process(records); // application-specific processing
  try {
    consumer.commitSync();
  } catch (CommitFailedException e) {
    // application-specific rollback of processed records
  }
}
```

**Рис. 5. Приклад використання нативного клієнта Apache Kafka на мові програмування Java**

Джерело: авторська система

На рис. 6 зображено приклад такої ж операції – отримання повідомлень з Kafka топіку, але вже з використанням фреймворку Spring Boot.

```
@Component
@KafkaListener(id = "multiGroup", topics = "multitype")
public class MultiTypeKafkaListener {

    @KafkaHandler
    public void handleGreeting(Greeting greeting) {
        System.out.println("Greeting received: " + greeting);
    }

    @KafkaHandler
    public void handleF(Farewell farewell) {
        System.out.println("Farewell received: " + farewell);
    }

    @KafkaHandler(isDefault = true)
    public void unknown(Object object) {
        System.out.println("Unkown type received: " + object);
    }

}
```

**Рис. 6. Приклад використання бібліотеки Spring Kafka**

Джерело: авторська система

В прикладі на Рис. 6 використовується декларативний підхід, і завдяки цьому можна перенести реалізацію технічних аспектів взаємодії з брокером до фреймворку і зосередитися вже безпосередньо на вирішенні бізнес-проблем, що суттєво спрощує і прискорює розробку вебзастосунків.

Конфігурацію підключення до брокера повідомлень можна визначити в окремих файлах налаштувань що дозволяє розмежувати код застосунку від конфігураційних файлів, дозволяючи зберігати ці конфігурації окремо або використовувати анотації [23].

Sping Kafka забезпечує спрощення процесу розробки завдяки, зокрема:

- підтримці транзакцій, коли бібліотека надає підтримку транзакцій, що дозволяє виконувати атомарні операції над Kafka та іншими джерелами даних;
- спрощеному тестуванню через надання засобів тестування коду (для впевненості в правильності коду), що використовується Kafka;

- інтеграції з компонентами Spring Framework, такими як Spring Boot, Spring Cloud, і Spring Integration [22]; це дозволяє легко інтегрувати Kafka у вебдодатки та мікросервіси, використовуючи всю потужність Spring.

**Висновки та пропозиції.** В статті досліджено та проаналізовано особливості та переваги використання асинхронної комунікації в мікросервісній архітектурі. Було визначено основні аспекти цього типу комунікації, відзначено їх важливість для розробників, архітекторів, менеджерів проектів і науковців.

Таким чином, грунтуючись на результатах проведеного дослідження, можна зробити, зокрема, наступні висновки:

- асинхронна комунікація важлива для мікросервісів, забезпечуючи гнучкість, масштабованість та стійкість системи, дозволяючи мікросервісам працювати незалежно один від одного;
- використання асинхронної комунікації надає можливість для зниження негативних наслідків затримок у взаємодії між мікросервісами, коли замість блокування процесу до завершення запиту, сервіс може продовжувати свою роботу і опрацьовувати інші запити;
- асинхронна комунікація сприяє підвищенню надійності та масштабованості мікросервісів, дозволяючи легко додавати та видаляти сервіси, збільшуючи стійкість до відмов та розподіленої обробки завдань.

Крім того, розробляючи проєкти з використанням асинхронної комунікації у мікросервісних системах, слід, зокрема:

- виконати оцінювання специфічних потреб та вимог до проєкту, враховуючи фактори масштабованості, стійкості та швидкості взаємодії;
- здійснити вибір технології реалізації асинхронної комунікації, враховуючи вимоги до проєкту;
- забезпечити моніторинг і керування асинхронною комунікацією для відстеження стану та надійності системи.

**ЛІТЕРАТУРА**

1. What are microservices? URL: https://microservices.io/ (дата звернення: 22.09.2023).

2. Asynchronous Communication: Definition and How to Use It. URL: https://www.getguru.com/reference/synchronous-vs-asynchronous-communication (дата звернення: 22.09.2023).

3. Consistency model. URL: https://en.wikipedia.org/wiki/Consistency_model (дата звернення: 24.09.2023).

4. What are message brokers? URL: https://www.ibm.com/topics/message-brokers. (дата звернення: 24.09.2023).

5. What is Message Queueing? URL: https://www.cloudamqp.com/blog/what-is-message-queuing.html (дата звернення: 23.09.2023).

6. What do you mean by «Event-Driven»? URL: https://martinfowler.com/articles/201701-event-driven.html (дата звернення: 11.09.2023).

7. Building a Robust Microservice Architecture: Understanding Communication Patterns. URL: https://identio.fi/en/blog/building-a-robust-microservice-architecture-understanding-communication-patterns/ (дата звернення: 10.09.2023).

8. Spring Boot with Kafka for Event-Driven Microservices. URL: https://medium.com/@AlexanderObregon/spring-boot-with-kafka-for-event-driven-microservices-5f9c0e51256e. (дата звернення: 11.09.2023).

9. Spring Boot Official Documentation. URL: https://spring.io/projects/spring-boot (дата звернення: 21.09.2023).

10. Apache Kafka Official Website. URL: https://kafka.apache.org/ (дата звернення: 26.09.2023).

11. Richardson C. Microservices Patterns: With examples in Java. URL: https://www.manning.com/books/microservices-patterns (дата звернення: 12.09.2023).

12. RabbitMQ with Java, Spring and Docker, asynchronous communication between microservices. URL: https://levelup.gitconnected.com/rabbitmq-with-java-and-spring-asynchronous-communication-between-microservices-c087595c500b. (дата звернення: 09.09.2023).

13. API. URL: https://en.wikipedia.org/wiki/API (дата звернення: 19.09.2023).

14. Publisher-Subscriber pattern. URL: https://www.enjoyalgorithms.com/blog/publisher-subscriber-pattern (дата звернення: 22.09.2023).

15. RabbitMQ Official Website. URL: https://www.rabbitmq.com/ (дата звернення: 25.09.2023).

16. Event-Driven Systems: A Deep Dive into Pub/Sub Architecture. URL: https://levelup.gitconnected.com/event-driven-systems-a-deep-dive-into-pubsub-architecture-39e416be913c (дата звернення: 12.09.2023).

17. What is REST? URL: https://www.codecademy.com/article/what-is-rest (дата звернення: 14.09.2023).

18. What is a Webhook? URL: https://www.markheath.net/post/basic-introduction-webhooks. (дата звернення: 09.09.2023).

19. WebSockets Security: Main Attacks and Risks. URL: https://www.vaadata.com/blog/websockets-security-attacks-risks. (дата звернення: 09.09.2023).

20. TCP/IP. URL: https://en.wikipedia.org/wiki/TCP/IP (дата звернення: 19.09.2023).

21. TCP handshake. URL: https://developer.mozilla.org/en-US/docs/Glossary/TCP_handshake (дата звернення: 19.09.2023).

22. Spring Integration Official Documentation. URL: https://spring.io/projects/spring-integration (дата звернення: 17.09.2023).

23. An Introduction to Annotations and Annotation Processing in Java. URL: https://reflectoring.io/java-annotation-processing/ (дата звернення: 27.09.2023).

## REFERENCES

1. What are microservices?, available at: https://microservices.io/ (Accessed 22 September 2023).

2. Asynchronous Communication: Definition and How to Use It, available at: https://www.getguru.com/reference/synchronous-vs-asynchronous-communication (Accessed 22 September 2023).

3. Consistency model, available at: https://en.wikipedia.org/wiki/Consistency_model (Accessed 24 September 2023).

4. What are message brokers? available at: https://www.ibm.com/topics/message-brokers (Accessed 24 September 2023).

5. What is Message Queueing? available at: https://www.cloudamqp.com/blog/what-is-message-queuing.html (Accessed 23 September 2023).

6. What do you mean by "Event-Driven"? available at: https://martinfowler.com/articles/201701-event-driven.html (Accessed 11 September 2023).

7. Building a Robust Microservice Architecture: Understanding Communication Patterns, available at: https://identio.fi/en/blog/building-a-robust-microservice-architecture-understanding-communication-patterns/ (Accessed 10 September 2023).

8. Spring Boot with Kafka for Event-Driven Microservices, available at: https://medium.com/@AlexanderObregon/spring-boot-with-kafka-for-event-driven-microservices-5f9c0e51256e (Accessed 11 September 2023).

9. Spring Boot Official Documentation, available at: https://spring.io/projects/spring-boot (Accessed 21 September 2023).

10. Apache Kafka Official Website, available at: https://kafka.apache.org/ (Accessed 26 September 2023).

11. Richardson C. Microservices Patterns: With examples in Java, available at: https://www.manning.com/books/microservices-patterns (Accessed 12 September 2023).

12. RabbitMQ with Java, Spring and Docker, asynchronous communication between microservices, available at: https://levelup.gitconnected.com/rabbitmq-with-java-and-spring-asynchronous-communication-between-microservices-c087595c500b (Accessed 09 September 2023).

13. API, available at: https://en.wikipedia.org/wiki/API (Accessed 19 September 2023).

14. Publisher-Subscriber pattern, available at: https://www.enjoyalgorithms.com/blog/publisher-subscriber-pattern (Accessed 22 September 2023).

15. RabbitMQ Official Website, available at: https://www.rabbitmq.com/ (Accessed 25 September 2023).

16. Event-Driven Systems: A Deep Dive into Pub/Sub Architecture, available at: https://levelup.gitconnected.com/event-driven-systems-a-deep-dive-into-pubsub-architecture-39e416be913c (Accessed 12 September 2023).

17. What is REST? available at: https://www.codecademy.com/article/what-is-rest (Accessed 14 September 2023).

18. What is a Webhook? available at: https://www.markheath.net/post/basic-introduction-webhooks (Accessed 09 September 2023).

19. WebSockets Security: Main Attacks and Risks, available at: https://www.vaadata.com/blog/websockets-security-attacks-risks (Accessed 09 September 2023).

20. TCP/IP, available at: https://en.wikipedia.org/wiki/TCP/IP (Accessed 19 September 2023).

21. TCP handshake, available at: https://developer.mozilla.org/en-US/docs/Glossary/TCP_handshake (Accessed 19 September 2023).

22. Spring Integration Official Documentation, available at: https://spring.io/projects/spring-integration (Accessed 17 September 2023).

23. An Introduction to Annotations and Annotation Processing in Java, available at: https://reflectoring.io/java-annotation-processing/ (Accessed 17 September 2023).

**СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 25.09.2023**

*УДК 004.77*
*DOI: https://doi.org/10.53920/ITS-2023-2-3*

**Олександр Іванович ГОЛУБЕНКО,**
кандидат технічних наук, доцент,
т. в. о. завідувача кафедри комп'ютерних наук
та інженерії програмного забезпечення,
ЗВО «Міжнародний науково-технічний університет
імені академіка Юрія Бугая»
ORCID ID: 0000-0002-1776-5160
**Андрій Вікторович ЛЕМЕШКО,**
доктор філософії, доцент,
доцент кафедри комп'ютерної інженерії,
Державний університет інформаційно-комунікаційних технологій
ORCID ID: 0000-0001-8003-3168
**Олександр Сергійович ЦВИК,**
аспірант кафедри комп'ютерної інженерії,
Державний університет інформаційно-комунікаційних технологій
ORCID ID: 0000-0001-7786-1712
**Юрій Валентинович МІШКУР,**
аспірант кафедри комп'ютерної інженерії,
Державний університет інформаційно-комунікаційних технологій
ORCID ID: 0009-0004-6807-6914

# ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ В ЛОКАЛЬНИХ МЕРЕЖАХ ЗА ДОПОМОГОЮ КОНТРОЛЮ ТРАФІКУ

***Розглянуто основи контролю трафіку й можливості збільшення потужності моніторингу мережі. Проаналізовано способи завдання шкоди мережі й створення інцидентів безпеки. Визначено основні аспекти безпеки, що повинні бути включені в моніторинг за мережею. Визначено, як контроль трафіку працює без додаткових сил моніторингу й основні можливості з новими технологіями. Був проведений аналіз основної мережі з використанням контролю трафіку й з вдосконаленням за допомогою додаткових засобів, що сприяло збільшенню можливостей моніторингу мережі й покращення контролю трафіку. Були розглянуті основні додаткові заходи, щодо використання адміністративних та технічні аспектів, що можливо використовувати в мережі задля безпеки. Розглянуто основні принципи контролю трафіку завдяки Network Traffic Analysis (NTA) й Network Behavior Analysis (NBA), завдяки чого***

***з'являється можливість виявлення шкідливого програмного забезпечення (ПЗ) й аномалій в мережі. Проаналізований спосіб моніторингу мережі за допомогою Data Leak Prevention (DLP) й приведені основні привелегії його використання та недоліки. Проаналізовані також системи моніторингу логів Security Information and Event Management (SIEM) й Intrusion Detection System (IDS), завдяки яких є можливість аналізувати додатки й периметр мережі для виявлення загроз.***

***Ключові слова:*** *контроль трафіку, мережа, моніторинг, інформаційна безпека, програмне забезпечення.*

**Oleksandr GOLUBENKO**
Candidate of technical sciences, associate professor,
Temporary acting head of the Department
of Computer Science and Software Engineering,
IHE «Academician Yuri Bugay international science
and technical university»
**Andrii LEMESHKO**
Doctor of Philosophy, Associate Professor,
associate professor of the departmentof computer engineering,
State University ofinformation and communication technologies
**Oleksandr TSVYK**
graduate student of the Department of Computer Engineering,
State University of information and communication technologies
**Yurii MISHKUR**
graduate student of the Department of Computer Engineering,
State University of information and communication technologies

## ENSURING INFORMATION SECURITY IN LOCAL NETWORKS USING TRAFFIC CONTROL

***The basics of traffic control and the possibility of increasing the power of network monitoring are considered. Methods of causing damage to the network and creating security incidents are analyzed. The main security aspects that should be included in network monitoring are defined. Defined how traffic control works without additional monitoring forces and basic capabilities with new technologies. An analysis of the core network using traffic control was performed and enhanced with additional tools, which contributed to increased network monitoring capabilities and improved traffic control. The main additional measures regarding the use of administrative and technical aspects that can be used***

***in the network for security were considered. The main principles of traffic control through Network Traffic Analysis and Network Behavior Analysis are considered, thanks to which the possibility of detecting malicious software and anomalies in the network appears. The method of network monitoring using Data Leak Prevention (DLP) is analyzed and the main advantages and disadvantages of its use are given. Security Information and Event Management (SIEM) and Intrusion Detection System (IDS) log monitoring systems were also analyzed, thanks to which it is possible to analyze applications and the network perimeter to detect threats.***

***Keywords:*** *traffic control, network, monitoring, information security, software.*

**Постановка проблеми.** Однією з основних проблем інформаційної безпеки (ІБ) є користувачі, які не повністю розуміють «кібер-гігієну» й можуть спровокувати надзвичайну ситуацію з витоком корпоративних даних. Проблемою є й зовнішній вплив на локальну мережу, хакерами чи зловмисниками, які намагаються добратися до даних компанії чи користувачів, та з кожним роком вдосконалюються й випробують різні способи для нанесення шкоди локальній мережі й зменшення супротиву для входу до його ядра – серверу. Тому постійно треба вдосконалювати мережу й основні принципи її безпеки, в тому числі й контроль трафіку.

Традиційна модель контролю трафіку не має переваг перед новими загрозами, що постійно з'являються, й має основну проблему – вона має принцип контролювання периметру мережі, який дозволяє отримувати весь трафік, але не забезпечує безпеку при деяких виняткових ситуаціях. До таких ситуацій відноситься: несанкціонований вхід не з корпоративних пристроїв, підключення флеш-пам'яті до мережевих пристроїв, WEB-сайти з «cryptojacking» або «miner» й передача сек'юрних даних 3-м особам через веб-сайти чи іншим способом. Так для вирішення даних проблеми потрібно використовувати додаткові заходи безпеки й це буде розглянуто в статті.

**Аналіз останніх досліджень і публікацій.** Більшість компаній, що надають послуги з безпеки мережі, проводять дослідження в яких є можливість визначити основні проблеми мережі на даний час. Також великі інфраструктурні IT компанії, проводять дослідження й надають багато інформації про випадки інцидентів безпеки. Компанія Embroker провела дослідження з кібербезпеки,

яка визначила основні проблеми інформаційної безпеки за останні роки (2020 – 2023), й в 2023 році виділила основні інциденти, які відбулися в мережі й завдали шкоду компаніям. Серед них можна виділити такі, як:

- Порушення регламентів ІБ (знайдено у 100% компаній);
- Підозрілу мережну активність (знайдено у 90% компаній);
- Активність шкідливого програмного забезпечення (знайдено у 68% компаній);
- Спроби експлуатації вразливості у ПЗ (знайдено у 31% компаній);
- Спроби підбору пароля (знайдено у 26% компаній);
- Спроби експлуатації веб-вразливостей (знайдено у 3% компаній);

Завдяки цьому аналізу можемо зробити висновки, як були шляхи завдання шкоди. З цього можемо виділити основу проблеми – недостатній аналіз трафіку й безпеки мережі, й це описано в [1] й вказано, на яких ділянках були основні проблеми.

**Мета статті** – це дослідження забезпечення інформаційної безпеки в локальних мережах за допомогою контролю трафіку.

**Виклад основного матеріалу дослідження.** Оскільки на сьогодні дуже важливим елементом є актуальна інформаційна безпека, то відбувається постійно вдосконалення програм й пристроїв, за допомогою яких й забезпечується безпека. Але оновлення відбувається на основі аналізу того, що зловмисниками раніше було розроблено та застосовано з метою не санкціонованого доступу до мережі, зламу мережі через сегменти, які погано захищені. Тому інциденти з безпеки відбуваються щоденно й не можуть бути зупинені. Завдяки застосуванню різних інструментів безпеки – Firewall, VPN, WLAN, контроль трафіку, антивірусів – ми маємо безпечну мережу, й у кожного є свої прошивки, оновлення й нові версії, за допомогою якої ми маємо вдосконаленні засоби захисту, й з кожною новою версією того ж самого Firewall ми й маємо нові спроби злому мережі.

Локальні мережі з кожним роком стають більш технологічними, з'являються нові сервіси та послуги, а разом з ними розвиваються й нові способи загрози інформаційній безпеці. Тому компанії намагаються з кожним разом знайти новий спосіб, як запобігти критичним інцидентам й зробити мережу безпечною. Основний з способів регулювання безпеки є контроль трафіку.

Перш за все розглянемо середовище порушень регламентів ІБ. Можна виділити основні проблеми:

- використання ПЗ для віддаленого доступу;
- використання незахищених протоколів передачі даних;
- використання BitTorrent;
- завантаження з подальшим встановленням потенційно небезпечного стороннього ПЗ;
- встановлення підміненого ПЗ (з прихованим функціоналом);
- застосування протоколів LLMNR, NetBios.

До підозрілої мережної активності можна віднести: приховування трафіку, множинну неуспішну автентифікацію, сканування внутрішньої мережі, спроби підключення до зовнішніх мереж, спроби віддаленого запуску програми, копіювання та передача даних у великих обсягах.

Завдяки вразливості програмного забезпечення зловмисник може використовувати функціонал програми у своїх цілях. У локальній мережі поширені спроби підбору пароля з допомогою методів перебору, це відбувається завдяки «брутфорсу» й може відбуватися як всередині мережі так й поза, завдяки віддаленим серверам. Веб-вразливості можуть бути використані для доступу до веб-сервера та виконання шкідливого коду.

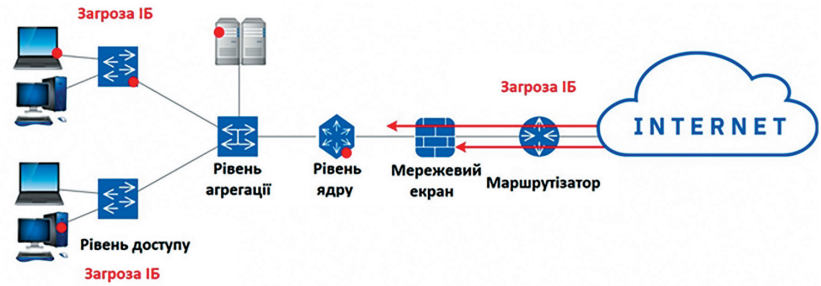


**Рис. 1. Вразливі місця локальних мереж**

Традиційна модель контролю трафіку на периметрі мережі не справляється з усіма загрозами ІБ з однієї простої причини - вона призначена для контролю периметра, а крім основного шляху, на якому стоїть міжмережевий екран (МЕ), існують обхідні канали отримання доступу до локальної мережі. Такими каналами можуть

бути підключені заражені мобільні пристрої, ноутбуки, флешки та диски. Особливу небезпеку становлять несанкціоновані підключення модемів та Wi-Fi адаптерів до пристроїв локальної мережі. Вони можуть відкрити канали передачі даних, які не контролюються МЕ.

Маючи доступ до всього трафіку в мережі, аналізуючи його та зіставляючи, ми можемо створити поведінковий профіль вузла. У цей профіль входитиме така інформація: які протоколи використовує, з якими сегментами мережі взаємодіє, скільки трафіку передає/отримує тощо. У разі виявлення нетипової поведінки (аномалії) подається сигнал тривоги. Поява нового вузла в мережі не залишиться непоміченою.

Завдяки розподіленій системі моніторингу трафіку локальної мережі є можливість детально провести аналіз поширення загроз ІБ по периметру в пошуку місця їх виникнення. Якщо додати потужності для зберігання трафіку, є можливість зробити аналіз у майбутніх інцидентах і розібратися у минулих. Завдяки цьому легко визначити вузькі місця, неправильну конфігурацію обладнання чи його несправність. Завдяки отриманим результатів є можливість удосконалення мережі й її безпеки.

Для запобігання порушенням регламентів ІБ, підозрілої мережевої активності, використання шкідливого ПЗ та експлуатації вразливості в ПЗ використовуються адміністративні та технічні заходи. Це відбувається за допомогою технічного підрозділу.

Для аналізу трафіку, взятого на периметрі і в самій мережі, на наявність заборонених протоколів і сервісів, використовуються Network Traffic Analysis і Network Behavior Analysis. За допомогою порівняння сигнатур та поведінкового аналізу система NTA/NBA зможе розпізнати роботу шкідливого ПЗ. Завдяки функціональним можливостям роботи з шифрованим трафіком та аналізом протоколів можна виявити аномалії в трафіку. Поведінковий аналіз та машинне навчання системи NTA/NBA дозволить виявити дивну активність у роботі мережі та спроби підбору облікових даних, що породжують велику кількість неуспішних аутентифікацій. Цей факт буде видно у трафіку.

Виявити загрози безпеці мережі та провести аналіз поведінки користувачів можна за допомогою системи Data Leak Prevention, яка використовує збір даних із точок моніторингу локальної мережі та з DLP-агентів, встановлених на мережному та абонентському

устаткуванні. Однак використання DLP-агентів може викликати труднощі, адже є пристрої, на які їх неможливо встановити, й на те є різні причини (відсутній доступ до ОС, немає підтримки з боку ОС, тощо). Існує ризик несанкціонованого підключення до мережі, коли користувачі можуть підключати свої особисті пристрої до мережі. Варто зазначити, що шкідливе програмне забезпечення може обходити моніторинг DLP-агента. Отримуючи трафік з точок моніторингу локальної мережі, система DLP зможе проводити аналіз поведінки всіх пристроїв у мережі незалежно, чи встановлений на них DLP-агент чи ні.

Система Security Information and Event Management збирає, аналізує логи різних додатків і на їх підставі може робити висновки про наявність загроз ІБ: підозрілий трафік та атаки, випадки зараження шкідливим програмним забезпеченням, порушення регламентів ІБ користувачами. Робота системи SIEM із вбудованим модулем NetFlow/sFlow полягає в наступному: трафік з точок моніторингу мережі надходить на модуль NetFlow/sFlow, модуль NetFlow/sFlow аналізує трафік, формує події та відправляє на SIEM, де проводиться аналіз та робиться висновок про наявність загрози ІБ. Таким чином, модуль NetFlow/sFlow збирає мережеву статистику з отриманого трафіку, а застосування його спільно з SIEM дозволяє здійснювати незалежний від інших програм моніторинг мережі на предмет загроз ІБ.

Система Intrusion Detection System призначена для виявлення вторгнень. Серед найпоширеніших видів IDS можна виділити Perimeter Intrusion Detection Systems (PIDS) та Network Intrusion Detection System (NIDS). Система PIDS аналізує трафік, взятий з периметра мережі (даний трафік може бути отриманий як з активного мережного обладнання, так і з моніторингу на мережі). Але не всі загрози ІБ можна виявити на периметрі мережі, вони можуть виникнути всередині мережі та не виходити назовні. Такі загрози найчастіше виникають під час підключення до мережі несанкціонованих пристроїв користувачів, підключення модемів, Wi-Fi адаптерів, зараження локальних пристроїв вірусами. Система NIDS отримує та аналізує трафік з точок моніторингу, розташованих по всій локальній мережі. Використання системи NIDS дозволяє виявити всі описані загрози ІБ.

**Висновки та пропозиції.** Визначивши основні загрози мережі, з'являється можливість їх нейтралізації й вдосконалення безпеки.

В цій статті були описані основні принципи контролю трафіку і визначені додаткові засоби до моніторингу мережі. Тому такі методи, як NTA/NBA, DLP, системи SIEM й IDS рекомендовано використовувати разом з основним контролем трафіку для запобігання виникнення інцидентів безпеки й для кращого регулювання потоку мережі. Вся система моніторингу, яка вказана вище, допоможе не тільки контролювати трафік, а й робити аналіз мережі з подальшим вдосконаленням мережі й покращенням інформаційно безпеки.



**ЛІТЕРАТУРА**

1. Ефективна та можлива техніка керування перевантаженнями для High. PerformanceMINs з розподіленою маршрутизацією на основі тегів, Транзакції IEEE у паралельних і розподілених системах, 243.

2. Автономна інженерія трафіку для надійності мережі. IEEE Journal on Selected Areas in Communications, 76.

3. Контроль перевантаження для високошвидкісної дротової мережі: систематичний огляд літератури. Журнал мережевих і комп'ютерних програм, 182.

4. Досягнення в запобіганні перевантаження TCP. 11-й міжнародний симпозіум IEEE з прикладного машинного інтелекту та інформатики, 129.

5. Управління ризиками. Загрози кібербезпеці у 2023 році https://www.embroker.com/blog/top-cybersecurity-threats/.

**REFERENCES**

1. An Effective and Feasible CongestionManagement Technique for High. PerformanceMINs with Tag-Based Distributed Routing, IEEE Transactions on Parallel and Distributed Systems, 243.

2. Autonomic traffic engineering for network robustness. IEEE Journal on Selected Areas in Communications, 76.

3. Congestion control for highspeed wired network: A systematic literature review. Journal of Network and Computer Applications, 182.

4. Advances in TCP Congestion Prevention. IEEE 11th International Symposium on Applied Machine Intelligence and Informatics, 129.

5. Risk Management Cybersecurity Threats in 2023 https://www.embroker.com/blog/top-cybersecurity-threats/.

**СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 01.12.2023**

*УДК 004.02, 004.455.2*
*DOI: https://doi.org/10.53920/ITS-2023-2-4*

**Олександр Андрійович ТКАЧЕНКО,**
кандидат фізико-математичних наук, доцент,
доцент кафедри інформаційних технологій,
Державний університет інфраструктури та технологій
ORCID ID: 0000-0001-6911-2770
**Богдан Юрійович ВОЛОХОНЕНКО,**
магістрант кафедри інформаційних технологій,
Державний університет інфраструктури та технологій
ORCID ID: 0009-0004-3031-4980

# ДЕЯКІ АСПЕКТИ РОЗРОБКИ ТА ВИКОРИСТАННЯ ГЕОІНФОРМАЦІЙНИХ В СІЛЬСЬКОМУ ГОСПОДАРСТВІ

***В наш час системи інформаційні системи використовуються в багатьох сферах економіки, науки, освіти, в нашому повсякденному житті. Особливе місце займають геоінформаційні інформаційні системи, серед задач яких, зокрема, є задачі, пов'язані з обробкою географічних даних (в тому числі й просторових). Сучасні геоінформаційні системи широко застосовуються в різних сферах, зокрема, в геології (для вивчення природних ресурсів, геологічних утворень і кліматичних змін), сільському господарстві (для управління земельними ресурсами, прогнозування врожаїв, оптимізації сільськогосподарських бізнес-процесів), міському господарстві (для планування розвитку і функціонування міст, забезпечення функціонування транспортної системи міста тощо), логістиці (для вирішенні задач маршрутизації, оптимізації доставки товарів, організації оптимальних пасажиро- і вантажопотоків), медицині (для відстеження поширення епідемій та пандемій, допомоги при визначенні осередків природних катастроф, тощо).***

***Метою роботи є аналіз та дослідження проблем щодо розробки програмного забезпечення відповідної геоінформаційної системи у сільському господарстві для оптимізації процесів управління земельними ресурсами, росту врожайності та підвищення стійкості галузі до внутрішніх і зовнішніх факторів. Мета і завдання статті спрямовані на просування інноваційних підходів та технологій, які можуть підтримати стійкий розвиток сільського***

***господарства та забезпечити високий рівень продуктивності та якості продукції.***

***Використання розробленої авторської геоінформаційної системи підтримує, зокрема, такі функції, як просторовий контекст (розгляд даних в контексті їх розташування на мапі), візуалізація даних на мапі робить інформацію більш зрозумілою і доступною для широкого кола користувачів, прогнозування різних явищ (наприклад, зміни клімату, повені, врожайності), планування ресурсів (наприклад за допомогою пенетрометра можна в режимі реального часу отримувати дані про щільність і якість грунту). Геоінформаційні системи залишаються незамінним інструментом, який допомагає вирішувати проблеми та приймати обгрунтовані рішення з урахуванням географічного контексту.***

***Ключові слова:*** *геоінформаційна система, географічні дані, мапа, просторові геодані, задачі та бізнес-процеси сільського господарства, інтерфейс системи.*

**Olexandr TKACHENKO**
PhD of physical and mathematical sciences, associate professor,
associate professor at the department
of information technologies,
State University of Infrastructure and Technology
**Bohdan VOLOKHONENKO**
Undergraduate at the department of information technologies,
State University of Infrastructure and Technology

## SOME ASPECTS OF THE DEVELOPMENT AND USE OF GEOINFORMATION IN AGRICULTURE

***Nowadays, information systems are used in many areas of the economy, science, education, and in our everyday life. A special place is occupied by geoinformation information systems, among the tasks of which, in particular, there are tasks related to the processing of geographic data (including spatial data). Modern geoinformation systems are widely used in various fields, in particular, in geology (for the research of natural resources, geological formations and climate changes), agriculture (for land management, crop forecasting, optimization of agricultural business processes), urban management (for development planning and the functioning of cities, ensuring***

***the functioning of the city's transport system, etc.), logistics (for solving routing problems, optimizing the delivery of goods, organizing optimal passenger and cargo flows), medicine (for tracking the spread of epidemics and pandemics, helping to identify centers of natural disasters, etc.).***

***The purpose of the work is the analysis and research of problems related to the development of software for the appropriate geoinformation system in agriculture to optimize the processes of land resource management, yield growth and increase the industry's resilience to internal and external factors. The purpose and objectives of the article are aimed at promoting innovative approaches and technologies that can support the sustainable development of agriculture and ensure a high level of productivity and product quality.***

***The use of the developed author's geoinformation system supports, in particular, such functions as spatial context (viewing data in the context of their location on the map), visualization of data on the map makes information more understandable and accessible to a wide range of users, forecasting of various phenomena (for example, climate change, floods, yield), resource planning (for example, with the help of a penetrometer, you can receive data on the density and quality of the soil in real time). Geographic information systems remain an indispensable tool that helps solve problems and make informed decisions taking into account the geographic context.***

***Keywords:*** *geoinformation system, geographic data, map, spatial geodata, tasks and business processes of agriculture, system interface.*

**Постановка проблеми.** В наш час географічні інформаційні системи (ГІС) [1, 2] стали необхідним інструментом, модернізуючи та трансформуючи способи, якими збираються, обробляються, зберігаються та використовуються різноманітні географічні дані. ГІС – це комплекс програмних і апаратних засобів, які дозволяють аналізувати та візуалізувати інформацію на мапах, пов'язуючи її з відповідними географічними координатами [3].

Іншими словами можна сказати, що ГІС – це система збору, зберігання, обробки, аналізу та візуалізації географічних даних. Важливою властивістю ГІС є можливість представлення просторових географічних даних [4]. Це означає, що інформація відображається на мапі, де географічні об'єкти, такі як дороги, річки, міста, розміщуються згідно з відповідними координатами на по-

верхні Землі. Сучасні ГІС знайшли широке застосування в різних сферах життєдіяльності держави та суспільства, зокрема, в:

- *геології*, де ГІС використовуються для вивчення природних ресурсів, геологічних утворень та кліматичних змін;
- *сільському господарстві* для управління земельними ресурсами, прогнозування врожаїв, оптимізації різноманітних сільськогосподарських бізнес-процесів;
- *господарстві міст* при розробці планів розвитку і функціонування міст, управлінні інфраструктурою міста та забезпечення ефективного функціонування транспортної системи міста (наприклад, прокладання транспортних маршрутів, моніторингу транспортного руху);
- *логістиці* при вирішенні багатьох класів задач маршрутизації та моніторингу логістичних і транспортних систем, оптимізації доставки товарів, організації оптимальних пасажиро- і вантажопотоків;
- *екології* для здійснення моніторингу і аналізу впливу на навколишнє середовище діяльності людини (як її безпосередньої життєдіяльності, так і опосередковано діяльності промислових, енергетичних, сільськогосподарських підприємств);
- *маркетингових дослідженнях* для визначення, наприклад, потенційних ринків та місць розташування магазинів;
- *медицині* при відстеженні поширення епідемій та пандемій, допомозі при визначенні осередків природних катастроф, боротьбі з наслідками цих катастроф та визначення найбільш небезпечних (патогенних) місць їхнього виникнення.

В наш час спостерігається постійний розвиток ГІС завдяки, зокрема, використанню:

- Big Data;
- штучного інтелекту (зокрема, нейромережевих технологій);
- різних технологій обробки великих обсягів даних;
- хмарним технологій [5];
- вебтехнологій;
- різних мобільних рішень та технологій.

Тому актуальність проблем, пов'язаних з розробкою сучасної ГІС для вирішення задач сільськогосподарського спрямування, не викликає сумнівів. Такі ГІС можуть використовуватися, зокрема,

для розв'язання проблем, пов’язаних зі змінами клімату, управління водними ресурсами в межах окремого регіону чи країни в цілому.

Таким чином ГІС повинно стати потужним інструментом, який допоможе краще розуміти нашу планету і приймати обгрунтовані оптимальні рішення при вирішенні багатьох актуальних проблем.

**Аналіз останніх досліджень і публікацій.** Інформатика та геоінформатика — теоретичний фундамент ГІС, які набули масштабного, широкого застосування в сучасних інформаційних технологіях і мають стрімкий та потужний розвиток [1]. ГІС поєднали в собі апаратні та технічні засоби, програмне та інформаційне забезпечення [2].

Сучасні ГІС та геоінформаційні технології (ГІТ) стали надзвичайно популярними і розповсюдженими в усіх галузях і сферах життя. Вони використовуються для розв'язання різноманітних завдань, включаючи планування, моделювання та управління на різних рівнях, від місцевого до національного. ГІС допомагають аналізувати природно-економічний потенціал, створювати транспортні магістралі та нафтопроводи, проводити екологічний та економічний моніторинг. Важливим є ще й те, що вони надають можливість забезпечу вати безпеку громадянського життя [3].

Сучасні ГІС передбачають наявність наступних підсистем [1 – 3]:

*1. Підсистема збору даних* в ГІС відіграє критичну роль у процесі створення та аналізу геопросторових інформаційних ресурсів, вона відповідає за збір та попередню обробку даних з різних джерел, що можуть включати, зокрема:

- аерофотознімання;
- цифрове дистанційне зондування;
- геодезичні вимірювання;
- словесні описи;
- статистичні дані.

Ця підсистема дозволяє створювати базу даних, яка містить точкові, лінійні та площинні об'єкти, їхні атрибути та географічні координати.

Завдяки використанню комп'ютерів і сучасних електронних пристроїв, таких як дигітайзери і сканери, підготовка початкових даних є більш ефективною. Це дозволяє записувати чи кодувати точки, лінії та області для подальшого використання в аналізі та

візуалізації інформації. Зокрема, можна використовувати готові цифрові мапи, цифрові моделі рельєфу, цифрові ортофотознімки та інші цифрові ресурси для підвищення точності та обсягу геопросторових даних [4].

Таким чином, створення підсистеми збору даних у ГІС є ключовим етапом у отриманні та управлінні географічною інформацією, яка використовується у багатьох галузях: від містобудування до наукових досліджень.

*2. Підсистема зберігання і вибірки даних в ГІС* дозволяє, зокрема:

- ефективно зберігати геопросторових даних;
- оновлювати та редагувати геопросторові дані;
- формулювати запити, щоб отримувати потрібну та контекстно-пов'язану інформацію [5].

Основна функція цієї підсистеми полягає в перенесенні уваги із загального аналізу інформації на точне формулювання запитів. Основна функція підсистеми – збереження геометричних координат, лінійних і площадкових об'єктів, а також їхні відповідні атрибути. Підсистема забезпечує зручний доступ до цих даних та можливість виконання різних операцій над ними.

*3. Підсистема обробки і аналізу даних в ГІС* виконує різноманітні завдання, базуючись на наданих різноманітних даних. Вона впорядковує, розподіляє, встановлює параметри та обмеження, а також проводить моделювання. Підсистема аналізу розроблена для спрощення та полегшення процесу обробки просторових даних, усуваючи потребу в ручних розрахунках та сприяючи автоматизації цього процесу для користувача.

*4. Підсистема виведення даних в ГІС* відображає геопросторову інформацію у різних форматах, включаючи табличний, діаграмний та картографічний. Ця підсистема перетворює дані у зручний для користувача вигляд, незалежно від того, чи має справу це з паперовою картографією, чи з цифровою картографією, використовуючи комп'ютерні засоби.

Головним результатом роботи підсистем є створення мап та інших візуалізацій географічних даних.

У структурі ГІС інформація організована у вигляді різних інформаційних шарів. Кожен шар є сукупністю геопросторових об'єктів, пов'язаних з певною темою чи класом об'єктів на певній території та в системі координат, яка є спільною для всіх шарів [6].

Під час створення ГІС важливо вибрати базові дані, які будуть використовуватися для об'єднання всіх даних.

Підсистема ГІС також надає можливість обробки польових геодезичних даних, включаючи імпорт даних з різних джерел та їх подальший аналіз. Це дозволяє вирішувати проблеми землеустрою та геодезії більш ефективно та точно.

ГІС є потужним інструментом організації, аналізу та візуалізації географічних даних, що знаходять широке застосування у різних сферах: від геології до містобудування.

Перевагою ГІС над звичайними «паперовими» методами досліджень є автоматизація, створення і використання просторових тривимірних моделей даних [7].

**Мета статті** полягає в аналізі та дослідженні проблем розробки програмного забезпечення ГІС та використання ГІТ і ГІС у сільському господарстві для оптимізації процесів управління земельними ресурсами, росту врожайності та підвищення стійкості галузі до внутрішніх і зовнішніх факторів.

Досягнення цієї мети забезпечується вирішенням, зокрема, наступних завдань:

- визначення потреб сільського господарства у відповідних ГІС та ГІТ;
- визначення класів задач, які можуть бути вирішені більш ефективно за допомогою відповідної ГІС;
- дослідження інструментарію, програмного забезпечення та технологій, адекватних потребам та можливостям сільськогосподарського сектора, що дозволяють збирати, аналізувати та використовувати географічні дані для покращення ефективності сільськогосподарського виробництва;
- розробка авторської ГІС для вирішення низки проблем сільського господарства (зокрема, проблем планування та розміщення різних сільсько-господарських об'єктів, проблем управління земельними ресурсами, проблем моніторингу стану посівів, тощо).

Мета і завдання статті спрямовані на просування інноваційних підходів та технологій, які можуть підтримати стійкий розвиток сільського господарства та забезпечити високий рівень продуктивності та якості продукції.

**Виклад основного матеріалу дослідження.** Важливою властивістю ГІС є можливість робити різні обчислення та аналіз

географічних даних, використовуючи спеціальні просторові функції, які дозволяють виконувати різні обчислення, в залежності від систем координат (наприклад, такі як обчислення площі, довжини, периметру, кутів, тощо).

Також важливою властивістю геоінформаційного забезпечення є можливість створення тривимірних просторових моделей для подальшого аналізу та візуалізації геоданих.

Зміст інформації, яка не є базовою та яку включають до предметно-орієнтованих ГІС, визначається за їхнім призначенням. Обробляючи та аналізуючи географічні дані, ГІС використовуються для вирішення різних проблем, зокрема, таких як:

- аналіз земельних ресурсів;
- міське планування;
- екологічний моніторинг;
- моделювання в транспортній сфері.

Просторові дані являють собою інформацію, яка пов'язана з конкретними географічними об'єктами або місцями на Землі і містить в собі інформацію про їхнє розташування, форму, розмір, атрибути та взаємодію.

Просторові дані можуть бути представлені у різних форматах і включати:

- *Геометричну інформацію*, яка є частиною просторових даних, описує геометричні аспекти об'єктів, такі як координати точок, ліній і полігонів, їхні розміри, форми і відношення до інших об'єктів. Геометричні дані дозволяють ГІС розуміти просторову структуру та географічні відносини між об'єктами.
- *Атрибутивну інформацію* – дані про характеристики об'єктів, такі як назви, описи, категорії, числові значення, дати та інші властивості. Атрибутивна інформація дозволяє ГІС виконувати аналіз і запити до даних, робити вибірку і фільтрацію об'єктів за їхніми властивостями.
- *Зв'язані дані* – дані, які встановлюють взаємозв'язки між різними об'єктами в просторі. Наприклад, це можуть бути лінії доріг, які з'єднують точки розташування, або полігони, що представляють адміністративні межі регіонів.
- *Часові дані*. Деякі просторові дані також містять інформацію про час, пов'язаний з об'єктами. Це може включати в себе час створення або зміни об'єкту, та інші часові параметри, які можуть бути важливі для аналізу динаміки подій.

Просторові дані у ГІС дозволяють збирати, зберігати, аналізувати та відображати інформацію про реальний світ на мапі. Вони використовуються в різних галузях, таких як геологія, сільське господарство, містобудування, екологія, транспортне планування, маркетинг та багато інших, для вирішення різноманітних проблем та прийняття відповідних управлінських рішень на основі географічних даних (рис. 1).

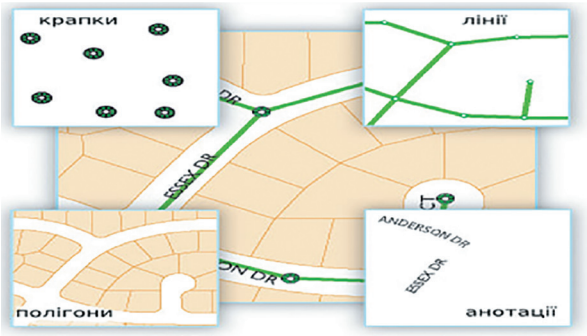


**Рис. 1. Просторові дані в ГІС ArcGis**

Джерело: [7]

Існує багато ГІС з різними можливостями та різним функціоналом для роботи з просторовими даними. Найбільш поширеними ГІС є: ArcGIS [7], Esri Ukraine [8] та QGIS [9].

ArcGIS – комерційна система від компанії Esri, яка підтримує широкий спектр функцій та має багато можливостей по роботі з просторовими геоданими. ArcGIS має багато користувачів, але є досить дорогою.

QGIS є безкоштовною та відкритою системою, що має меншу кількість функцій, ніж ArcGIS, але спектр її функцій можна розширити за допомогою плагінів. QGIS є популярним вибором для некомерційної та дослідницької роботи.

Якщо порівнювати ці дві сучасні ГІС, то можна дійти висновку, що ArcGIS має фк переваги, так і недоліки, зокрема:

- багатофунціональний інтерфейс та інтуїтивне користувацьке середовище;
- великий спектр функціоналу, а саме:
  – геопроцесінг;
  – аналіз та візуалізація даних;
  – підтримка різних форматів даних;

- обмежену безкоштовну версію;
- високу вартість ліцензії;
- високі вимоги до обладнання та ресурсів комп'ютера;
- складне налаштування та розгортання системи.

QGIS в цьому плані має, зокрема:

- безкоштовну версію;
- легко налаштовується та розгортається;
- підтримує великий вибір форматів даних;
- функціонал надає підтримку геопроцесінгу, візуалізації та аналізу даних.

QGIS має не такий багатофункціональний та менш інтуїтивний інтерфейс порівняно з ArcGIS, ирму може виникати проблема з сумісністю з деякими форматами даних. Менш відомою та використовуваною є ГІС GRASS GIS [16].

GRASS GIS – безкоштовна та відкрита система, яка зосереджується на аналізі та обробці геоданих. Вона є потужною та гнучкою, але для її використовування необхідні спеціалізовані знання та навички роботи з геоданими та іншими ресурсами.

Описані вище системи можуть співпрацювати з просторовими даними, але не підходять під вирішення проблеми для сільського господарства. Тому виникла проблема розробки авторської ГІС для вирішення широкого колі практичних завдань в сільському господарстві.

Просторові дані в ГІС мають великий потенціал для застосування в сільському господарстві і можуть бути використані для вирішення різноманітних завдань та оптимізації процесів у цій галузі [10].

На основі проведеного аналізу використання сучасних ГІС та з урахуванням класів проблем, які треба вирішувати у сільському господарстві, була розроблена авторська ГІС.

Завдяки просторовим даним і створено авторської системи були вирішені, зокрема, такі проблеми сільського господарства:

*Землевпорядкування і управління земельними ресурсами.* Геоінформаційна система дозволила детально проаналізувати земельні ділянки, їхні характеристики, вологість, відстань до водних джерел і багато інших параметрів [11]. Завдяки цій інформації, система допомагає сільським господарствам раціонально використовувати землю, планувати вирощування рослин і встановлювати оптимальні межі сільськогосподарських ділянок.

Під час дослідження був використаний пенетрометр (рис. 2).

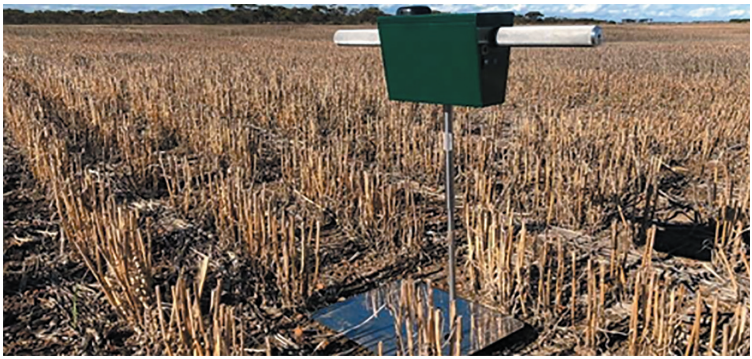

**Рис. 2. Грунтовий пенетрометр Superagronom**

Джерело: [12]

*Пенетрометр* — пристрій для інженерно-геологічних досліджень, який дозволяє вивчати щільність ґрунту на різних глибинах. Завдяки пенетрометру ми маємо можливість проконтролювати міцність і якість ґрунту перед зведенням житлових, промислових та агрономних об'єктів [12].

Принцип роботи пенетрометра полягає в визначенні проникності ґрунту на основі глибини проникнення стандартного інструмента при підведенні певного навантаження. Пенетрометр може точно вимірювати проникність ґрунту в міліметрах, що дозволяє отримати докладний профіль ґрунтових шарів на певній ділянці.

Існують два основних типи ґрунтових пенетрометрів: статичні, які використовують статичне навантаження для вимірювання опору ґрунту, і динамічні, в яких вимірювальний інструмент занурюється в ґрунт через серію ударів.

Пристрій надсилає виміряну щільність, опір ґрунту на сервер, який зберігає надіслані дані і відображає отриману інформацію на мапі.

*Моніторинг стану посівів*. Створена авторська ГІС дозволила вести моніторинг врожайності та стану посівів на польових ділянках.

За допомогою датчиків можна вчасно виявляти проблеми, такі як шкідники, хвороби або стресові умови для рослин і вживати заходів для їхнього контролю. Користувач самостійно вводить інформацію про стан посівів та врожайності і отримує аналіз.

*Оптимізація використання добрив і ресурсів*. ГІС дозволила проаналізувати різні параметри ґрунту і води, щоб розрахувати

оптимальні дози добрив і розташування систем поливу. Це потенційно зменшує витрати та підвищує врожайність.

Під час розрахунку внесення добрива враховується наступні фактори:

- врожайність,
- необхідний рівень якості продукту;
- родючість грунту;
- тип грунту;
- обробка поживних решток.

Метод прямого використання результатів польових дослідів є точним, якщо родючість ґрунтів наукової установи та господарства однакова [12]. Через різницю у ґрунтовій родючості на окремих полях, рекомендовані дози добрив для сільськогосподарських культур обчислюються за допомогою відповідних коефіцієнтів. Це потребує витрат часу та коштів на проведення дослідів і обмежує застосування цього методу за зональними межами. Однак існують інші методи розрахунку норм добрив, які включають, зокрема:

а) *Метод елементного балансу*, який базується на розрахунку норм добрив на основі вмісту поживних речовин у ґрунті та очікуваного виносу цих елементів сільськогосподарськими культурами.

б) *Метод розрахунку норм добрив* на основі підвищення врожайності.

Суть методу з використанням елементного балансу полягає в тому, що дози добрив визначаються як різниця між виносом поживних елементів плановим врожаєм культур та можливим їх використанням із ґрунту. При цьому враховується доступність поживних елементів у ґрунті та добрив.

Вказані вище методи дозволяють більш точно визначити добривні норми для підвищення врожайності сільськогосподарських культур [13]. Дози добрив обчислюються за формулою:

$$Д_у = В – (П – К_п)/К_у \ ,$$

де $Д_у$ – доза елемента добрива, кг/га,

В – винос елемента живлення плановим уражаєм, кг/га,

П – вміст рухомих форм елемента в шарі грунту, кг/га,

$К_п$ – коефіцієнт використання елемента рослинами з грунту,

$К_у$ –коефіцієнт використання елемента рослинами з добрива

Метод достатньо простий, але точність визначення оптимальної дози елемента добрива не є високою, оскільки усі показники, які були використані для розрахунку норми, варіюються.

Наприклад, коефіцієнти використання елементів живлення з грунту і добрив та їх витрати значно варіюють в залежності від декількох факторів: сортові особливості рослини, погодні умови, родючість грунту. Варіювання параметрів може досягати до 35%, а в деяких випадках навіть до 50%.

*Маркетинг та аналіз ринку.* Авторська ГІС дозволила проаналізувати ринки збуту для сільськогосподарської продукції та визначити найбільш прибуткові регіони для постачання. Авторська ГІС допомагає визначити також попити, пропозиції та тенденції на ринку [14].

Дослідження та аналіз ринку відбувається за відповідним планом:

- на першому етапі аналізується загальний стан ринку – аналіз тенденцій ринку, розрахунок ємності ринку, загальні показники, сегментація та структуризація ринку, сировинна база [15];
- на другому етапі після отримання загальної характеристики ринку відбувається державне регулювання галузі, яке передбачає основні закони, що регулюють галузь, податки та збори на ринку, принципи проведення державних закупівель.

Виробництво та динаміка реалізацій має цілу низку показників, зокрема таких, як обсяг виробництва та реалізації в кількісних та вартісних показниках, а також динаміка зростання виробництва і реалізації товарів та послуг.

- на третьому етапі після аналізу первинної інформації, йде глибокий аналіз операторів ринку:
  – перелік основних операторів;
  – структурування операторів;
  – частки ринку операторів;
  – ступінь конкуренції, ризики;
  – факторний аналіз часток ринкових конкурентів.

Обсяг продажів на ринку є головною метою дослідження ринку – експорт та імпорт (аналіз обсягів, структури, цін, географії поставок по країнах). Завдяки аналізу ціноутворень на ринку в авторській ГІС можна вирахувати майбутні ціни на продукти: динаміка цін на продукцію, опис факторів що впливають на формування цін, собівартість продукції та її рентабельність.

- на четвертому етапі після глибокого аналізу ринку, проводиться аналіз різних категорій споживачів та їхніх вподобань, зокрема:

– формування портрету споживача;
– визначення вподобань різних категорій щодо того чи іншого товару;
– аналіз інтернет-аудиторії в конкретній галузі;
– аналіз державних тендерних закупівель (обсяги, структури по регіонам, організатори та учасники, сегментація споживання, потреба у товарі та ступінь задоволеності споживачів).

- на п'ятому етапі після дослідження ринку виконується підведення підсумків, зокрема:
  – генеруються прогнозні тенденції розвитку ринку;
  – формуються гіпотези розвитку ринку;
  – надаються рекомендації щодо розвитку на ринку.

*Планування і моніторинг інфраструктури.* Авторська ГІС допомагає планувати розміщення ферм, доріг, зерносховищ і інфраструктури для сільського господарства (рис. 3, рис. 4). Вона також дозволяє здійснювати моніторинг стану сільськогосподарської інфраструктури для забезпечення ефективного її функціонування.


**Рис. 3. Приклад відображення збережених на сервері даних з пенетрометра**

Джерело: авторська система


**Рис. 4. Планування розміщення ферм**
Джерело: авторська система

*Управління різними видами культур.* Авторська ГІС дозволяє сільським господарствам визначати найкращі види культур для конкретних умов, враховуючи кліматичні умови, грунти та доступні ресурси.

Авторська ГІС має також моніторингову систему для ведення кількості наявних ресурсів та оновлюється в режимі реального часу.

Авторська система має змогу заздалегідь дізнаватись можливі погодні явища і сповіщати користувача при небезпеці, якщо погодні умови будуть загрожувати функціям життєдіяльності рослин.

**Висновки та пропозиції.** ГІС є потужним інструментом, який сприяє збору, аналізу та візуалізації географічної інформації в реальному часі, використовуючи її інтеграцію з іншими численними видами інформації, такими як соціальні, економічні, екологічні тощо. ГІС забезпечує можливість приймати обгрунтовані рішення в різних галузях виробництва.

Розроблена авторська ГІС дозволяє, зокрема, підтримку наступних функцій:

- *Просторовий контекст:* ГІС дозволяє розглядати дані в контексті їх розташування на мапі. Це сприяє кращому розумінню інформації і виявленню географічних взаємо-

зв'язків. Для того, аби можливо було використовувати і розташовувати дані на мапі використовують просторові дані (крапки, лінії, полігони та анотації).

- *Збір та аналіз даних:* ГІС дозволяє збирати дані з різних джерел і аналізувати їх, щоб виявити тенденції, залежності та шаблони. Це важливо для прийняття обгрунтованих рішень. Система аналізує і досліджує ринок за рядом факторів, що є її перевагою.
- *Візуалізація:* Можливість візуалізації даних на мапі робить інформацію більш зрозумілою і доступною для широкого кола користувачів. Завдяки інтеграції з онлайн-мапами система в режимі реального часу показує де знаходяться зареєстровані користувачем об'єкти.
- *Прогнозування*: ГІС може бути використана для прогнозування різних явищ, таких як зміни клімату, повені або врожайності. Вона дозволяє враховувати географічні фактори при прогнозуванні. При загрозі для рослин система активує «небезпеку» і повідомляє користувачам що треба зробити, щоб уникнути небажаних наслідків.
- *Планування ресурсів:* ГІС допомагає управляти природними ресурсами та інфраструктурою, що є важливим аспектом сталого розвитку. За допомогою пенетрометра можна в режимі реального часу отримувати виміряні дані про щільність і якість грунту та зберігати це все на сервісі.
- *Спільна робота:* ГІС підтримує спільну роботу різних користувачів, що дозволяє ефективно обмінюватися даними та спільно працювати над проектами.
- *Збереження часу і коштів:* ГІС дозволяє збільшити продуктивність і ефективність роботи, що зменшує витрати часу і коштів. Однак важливо враховувати, що ГІС також має свої обмеження, такі як високі вимоги до обладнання та програмного забезпечення, необхідність постійного оновлення даних.

Таким чином, можна вважати, що ГІС залишається незамінним інструментом для сучасного суспільства, який допомагає вирішувати складні проблеми та приймати обгрунтовані рішення з урахуванням географічного контексту. Система продовжує розвиватися та знаходити нові застосування в різних сферах діяльності.

**ЛІТЕРАТУРА**

1. Донченко М.В., Коваленко І.І. Геоінформаційні системи. Миколаїв: Вид-во ЧНУ ім. Петра Могили, 2021. 132 с. ISBN 978-966-336-245-8.

2. GIS (Geographic Information System). URL: https://education.nationalgeographic.org/resource/geographic-information-system-gis/ (дата звернення: 11.09.2023).

3. Decadal Monitoring of the Hydrothermal System of Stromboli Volcano, Italy. DOI: https://doi.org/10.1029/2023GC010931 (дата звернення: 11.09.2023).

4. Зацерковний В.І., Бурачек В.Г., Железняк О.О., Терещенко А.О. Геоінформаційні системи і бази даних. Ніжин: НДУ імені М.В. Гоголя, 2017. 237 с.

5. Хмарні технології для збору геоданих. URL: https://ngc.com.ua/ua/info/GIScloud.html (дата звернення: 04.09.2023).

6. Опара В.М., Бузіна І.М., Хайнус Д.Д. Ландшафтно-екологічні дослідження екосистем сучасними методами. *Проблеми безперервної географічної освіти і картографії*, 2019. Вип. 29. С. 55 – 63. DOI: https://doi.org/10.26565/2075-1893-2019-29-06.

7. ArcGis. GeoDatabase. URL: https://pro.arcgis.com/en/pro-app/latest/help/data/geodatabases/overview/feature-class-basics.html (дата звернення: 25.09.2023).

8. Esri Ukraine. URL: https://esri.ua/sarticle.php?id=1 (дата звернення: 14.09.2023).

9. QGIS. Documentation. URL: https://qgis.org/uk/docs/index.html (дата звернення: 14.09.2023).

10. The Bio Geosystem Strategy for Sustainable Irrigated Agriculture in Africa. URL: https://journalcjast.com/index.php/CJAST/article/view/4061 DOI: https://doi.org/10.9734/cjast/2023/v42i44061 (дата звернення: 24.09.2023).

11. Loiskandl W, Nolz R. Requirements for sustainable irrigated agriculture. Agronomy. DOI: https://doi.org/10.3390/agronomy11020306 (дата звернення: 19.09.2023).

12. Пенетрометр. URL: https://superagronom.com/slovnik-agronoma/penetrometr-id20091 (дата звернення: 22.09.2023).

13. Норми внесення добрив. URL: https://agro-business.com.ua/ahrarni-kultury/item/10797-normy-vnesennia-dobryv.html (дата звернення: 23.09.2023).

14. Аналітичне дослідження ринку. URL: https://pro-consulting.ua/ua/issledovanie-rynka/analiz-rynka-selskohozyajstvennoj-otrasli-ukrainy-2022-god (дата звернення: 18.09.2023).

15. Український експорт продукцій. URL: https://landlord.ua/reytingi/top-15-pozytsii-ukrainskoho-eksportu-ahrarnoi-produktsii/ (дата звернення: 25.09.2023).

16. GRASS GIS. Bringing advanced geospatial technologies to the world. URL: https://grass.osgeo.org (дата звернення: 23.09.2023).

**REFERENCES**

1. Donchenko, M.V., Kovalenko, I.I. (2021). Geoinformation systems. Mykolaiv: Publishing House of the ChNU named after Petra Mohyly. 132 p. ISBN 978-966-336-245-8.

2. GIS (Geographic Information System), available at: https://education.nationalgeographic.org/resource/geographic-information-system-gis/ (Accessed 11 September 2023).

3. Decadal Monitoring of the Hydrothermal System of Stromboli Volcano, Italy, available at: https://doi.org/10.1029/2023GC010931 DOI: 10.1029/2023GC010931 (Accessed 11 September 2023).

4. Zatserkovny, V.I., Burachek, V.G., Zheleznyak O.O., Tereshchenko, A.O. (2017). Geoinformation systems and databases. Nizhin: NSU named after M.V. Gogol, 237 p.

5. Cloud technologies for geodata collection, available at: https://ngc.com.ua/ua/info/GIScloud.html (Accessed 04 September 2023).

6. Opara, V.M., Buzina, I.M., Hainus, D.D. (2019). «Landscape and ecological studies of ecosystems using modern methods», *Problems of continuous geographical education and cartography*, Vol. 29: 55 – 63. DOI: https://doi.org/10.26565/2075-1893-2019-29-06.

7. ArcGis. GeoDatabase, available at: https://pro.arcgis.com/en/pro-app/latest/help/data/geodatabases/overview/feature-class-basics.html (Accessed 25 September 2023).

8. Esri Ukraine, available at: https://esri.ua/sarticle.php?id=1 (Accessed 14 September 2023).

9. QGIS. Documentation, available at: https://qgis.org/uk/docs/index.html (Accessed 14 September 2023).

10. The Bio Geosystem Strategy for Sustainable Irrigated Agriculture in Africa, available at: https://journalcjast.com/index.php/CJAST/article/view/4061 DOI: https://doi.org/10.9734/cjast/2023/v42i44061 (Accessed 24 September 2023).

11. Loiskandl, W, Nolz, R. Requirements for sustainable irrigated agriculture. Agronomy, available at: https://doi.org/10.3390/agronomy11020306 DOI: 10.3390/agronomy11020306 (Accessed 19 September 2023).

12. Penetrometer, available at: https://superagronom.com/slovnik-agronoma/penetrometr-id20091 (Accessed 22 September 2023).

13. Norms of application of fertilizers, available at: https://agro-business.com.ua/ahrarni-kultury/item/10797-normy-vnesennia-dobryv.html (Accessed 23 September 2023).

14. Analytical research of the market, available at: https://pro-consulting.ua/ua/issledovanie-rynka/analiz-rynka-selskohozyajstvennoj-otrasli-ukrainy-2022-god (Accessed 18 September 2023).

15. Ukrainian export of products, available at: https://landlord.ua/reytingi/top-15-pozytsii-ukrainskoho-eksportu-ahrarnoi-produktsii/ (Accessed 25 September 2023).

16. GRASS GIS. Bringing advanced geospatial technologies to the world, available at: https://grass.osgeo.org (Accessed 23 September 2023).

**СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 25.09.2023**

*УДК 004.8*
*DOI: https://doi.org/10.53920/ITS-2023-2-5*

**Олександр Іванович ГОЛУБЕНКО,**
кандидат технічних наук, доцент,
т. в. о. завідувача кафедри комп'ютерних наук
та інженерії програмного забезпечення,
ЗВО «Міжнародний науково-технічний університет
імені академіка Юрія Бугая»
ORCID ID: 0000-0002-1776-5160
**Андрій Вікторович ЛЕМЕШКО,**
доктор філософії, доцент,
доцент кафедри комп'ютерної інженерії
Державний університет інформаційно-комунікаційних технологій
ORCID ID: 0000-0001-8003-3168
**Андрій Русланович ПОЛІЩУК,**
магістр кафедри комп'ютерної інженерії,
Державний університет інформаційно-комунікаційних технологій
ORCID ID: 0009-0000-7965-6499
**Максим Володимирович КУЗЬМЕНКО,**
магістр кафедри комп'ютерної інженерії,
Державний університет інформаційно-комунікаційних технологій
ORCID ID: 0009-0009-8146-8038
**Євген Олександрович ДЕГТЯРЬОВ,**
магістр кафедри комп'ютерної інженерії,
Державний університет інформаційно-комунікаційних технологій
ORCID ID: 0009-0002-7219-9437

# ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У КІБЕРБЕЗПЕЦІ

***Схема штучного інтелекту на ринку кібербезпеки допомагає організаціям контролювати, виявляти, повідомляти про кіберзагрози та протидіяти їм, щоб зберегти конфіденційність інформації. Зростаюча обізнаність людей, прогрес в інформаційних технологіях, модернізація розвідувальних і поліцейських рішень для роботи, а також збільшення обсягу знань, зібраних з численних джерел, вимагають використання надійних і вдосконалених рішень кібербезпеки в усіх згаданих галузях. Збільшення інтенсивності та складності кібератак акцентує увагу на потребі розвитку кіберсистем, підтриманих штучним інтелектом. Ростуть***

***численність та масштаб кіберінцидентів на глобальному рівні, що заставляє організації активізувати заходи захисту своєї інформації. Мотивація атак може бути різноманітною: від політичної конкуренції до крадіжки міжнародної інформації та радикальних ідеологічних мотивацій.***

***Ключові слова:*** *штучний інтелект, кібербезпека, комп'ютерна система, алгоритм, машинне навчання, інтернет речей, мережа, середовище.*

**Oleksandr GOLUBENKO**
Candidate of technical sciences, associate professor,
Temporary acting head of the Department
of Computer Science and Software Engineering,
IHE «Academician Yuri Bugay international science
and technical university»
**Andrii LEMESHKO**
Doctor of Philosophy, Associate Professor,
associate professor of the department of computer engineering,
State University of information and communication technologies
**Andrii POLISHCHUK,**
**Maksym KUZMENKO,**
**Eugene DEGTYAREV**
Masters of computer engineering department,
State University of information and communication technologies

## RESEARCH ON THE APPLICATION OF ARTIFICIAL INTELLIGENCE IN CYBER SECURITY

***The AI framework in the cybersecurity market helps organizations monitor, detect, report, and counter cyber threats to preserve information privacy. The growing awareness of people, the progress in information technology, the modernization of intelligence and police work solutions, as well as the increase in the amount of knowledge collected from numerous sources, require the use of reliable and advanced cyber security solutions in all the mentioned industries. The increase in the intensity and complexity of cyber-attacks emphasizes the need for the development of cyber systems supported by artificial intelligence. The number and scale of cyber incidents at the global level are growing, forcing organizations to step up measures to protect their information. The motivation of the attacks can be diverse: from political competition to the theft of international information and radical ideological motivations.***

***Keywords:*** *artificial intelligence, cyber security, computer system, algorithm, machine learning, Internet of Things, network, environment.*

**Постановка проблеми.** Сучасний конкурентний бізнес, активно використовує інноваційний підхід до організації та провадження діяльності, заснований на технологіях штучного інтелекту. Ця технологія дозволяє електронним автоматизованим системам (комп'ютерам) не лише самостійно приймати управлінські рішення, але й виявляти, аналізувати та запобігати кібератакам у режимі реального часу. Штучний інтелект в кібербезпеці стає ключовим для автоматизації захисних механізмів та стратегічного управління ризиками. Водночас, незважаючи на потенційні переваги, існує необхідність ефективно протидіяти новим кіберзагрозам, що виникають у контексті цієї інноваційної динаміки.

**Аналіз останніх досліджень і публікацій.** Кібербезпека відноситься до комплексу заходів, спрямованих на захист електронних даних та систем. Аналогічно до закону Мура, який передбачає зменшення розмірів компонентів інтегральних схем із збільшенням їх продуктивності, кіберзлочинці постійно покращують ефективність своїх атак, витрачаючи на це все менше коштів з часом. Важливо зауважити, що основною метою більшості кібератак є фінансова вигода [1]. За різними джерелами та оцінками, світові витрати на кібербезпеку між 2016 та 2021 роками склали понад 1 трильйон доларів. Витрати на кібербезпеку з 2016 року вже зросли більш ніж на 40 відсотків.

**Мета статті** – дослідження застосування штучного інтелекту для покращення кібербезпеки.

**Виклад основного матеріалу дослідження.** Штучний інтелект – це система, створена на основі комп'ютерних технологій, яка намагається моделювати певні аспекти людського менталітету та функціонування. Ця система може взаємодіяти з навколишнім середовищем, наприклад, розпізнавати голос та перетворювати його на різні мови, наслідуючи при цьому людські здібності. Він базується на різних науках, таких як математика, інформатика та філософія, і його головною метою є створення систем, які можуть демонструвати певні аспекти людської інтелектуальної діяльності. Термін «штучний інтелект» часто використовується для опису систем, які здатні емалювати основні функції сприйняття та розуміння, що є характерними для людського мислення рисунок 1 [2].


**Рис. 1. Використання штучного інтелекту**

Інтеграція штучного інтелекту у системи виявлення вторгнень (IDS) набуває великої актуальності. У роботі X. A. Larriva-Novo та співавторів [4] пропонуються алгоритми для покращення ефективності IDS, зокрема в контексті конкретних сценаріїв. Для цього проводиться категоризація наборів даних кібербезпеки, що дозволяє групувати їх у специфіковані категорії. В роботі розглядаються різні моделі нейронних мереж, такі як багатошарові та рекурентні, використовуються різноманітні функції активації та алгоритми навчання, щоб досягти оптимальної точності в залежності від характеристик бази даних.

Нарешті, результати були використані, щоб визначити, яка категорія набору даних про кібербезпеку є більш важливою для виявлення вторгнень і найбільш адекватної конфігурації алгоритму машинного навчання для мінімізації навантаження на обчислення. Існують також значні ризики для безпеки у взаємозв'язках, необхідних для використання певних переваг автоматизованих систем. У статті описана система виявлення вторгнень, заснована на концепціях некерованих автоматизованих систем.

Представлено модель машинного навчання безпеки на основі дерева (IntruD Tree), яка оцінює функцію безпеки на основі її

важливості та створює загальну модель виявлення вторгнень. Ця модель не тільки має точність прогнозування для непривабливих тестових випадків, вона також мінімізує комп'ютерну складність моделі, зменшуючи вимірювання функцій. Нарешті, набори даних кібербезпеки та вимірювання точності, використовуються для перевірки ефективності нашої моделі IntruDTree. Автори також порівнюють результати IntruDTree з різними традиційними, звичайними підходами машинного вчителя, такими як наївна система класифікації, логістична регресія, векторні системи підтримки та найвужчий сусід, щоб оцінити ефективність отриманої моделі безпеки.

У статті було запропоновано нейроморфний когнітивний обчислювальний підхід для системи виявлення вторгнень у мережу (IDS) кібербезпеки глибокого навчання (DL). Алгоритмічна потужність DL була поєднана зі швидкими та високоефективними нейроморфними процесорами кібербезпеки. Дані були пронумеровані для навчання за допомогою технік ретельного навчання без нагляду, які називаються автоматичним кодувальником під час процесу навчання (AE). Вагові коефіцієнти AE, створені для етапу навчання під керівництвом, використовуються як початкові вагові коефіцієнти для нейронних мереж. Остаточна вага перетворюється на дискретну вагу, синаптичну вагу та порогові значення для нервових клітин за допомогою дискретної факторизації векторів (DVF). Нарешті, згенеровані поперечні ваги, синаптичні ваги, пороги та витоки були зіставлені в поперечні смуги та нейрони. Під час контрольної точки, закодовані зразки перетворюються в центральну форму за допомогою гібридних методів кодування. Для реалізації та тестування використовували IBM Neurosynaptic Core Simulator (NSCS) і новий нейросинаптичний чіп True North. Для виявлення вторгнення в кібербезпеку нейроморфного чіпа результати тесту вказують на точність приблизно 90,12 відсотка. Крім того, автори переглянули запропоновану структуру не тільки для виявлення шкідливих пакетів, але й для класифікації цих типів атак і досягли точності 81,31%. Нейроморфна реалізація забезпечує дивовижну точність у виявленні та класифікації високопотужного виявлення мережевого вторгнення. Для виявлення вторгнення в кібербезпеку нейроморфного чіпа результати тесту вказують на точність приблизно 90,12 відсотка. Крім того, автори переглянули запропоновану структуру не тільки для виявлення

шкідливих пакетів, але й для класифікації цих типів атак і досягли точності 81,31%.

Нейроморфна реалізація забезпечує дивовижну точність у виявленні та класифікації високопотужного виявлення мережевого вторгнення. Для виявлення вторгнення в кібербезпеку нейроморфного чіпа результати тесту вказують на точність приблизно 90,12 відсотка. Крім того, автори переглянули запропоновану структуру не тільки для виявлення шкідливих пакетів, але й для класифікації цих типів атак і досягли точності 81,31%. Нейроморфна реалізація забезпечує дивовижну точність у виявленні та класифікації високопотужного виявлення мережевого вторгнення.

Технологія машинного навчання популярна в багатьох сферах, а технологія машинного навчання має багато застосувань для кібербезпеки. Приклади шкідливого програмного забезпечення включають аналіз шкідливого програмного забезпечення, зокрема виявлення зловмисного програмного забезпечення нульового дня, аналіз загроз, виявлення аномалій вторгнення та багато інших. У багатьох продуктах кібербезпеки вчені використовують виявлення машинного навчання через неефективність підходів на основі сигнатур у виявленні неденних атак або навіть незначних варіантів існуючих атак. У цьому [7] дослідженні, в якому машинне навчання є методом, автори обговорюють різні сфери кібербезпеки. Щоб маніпулювати навчанням і дослідженнями класифікації даних, автори також мають певний досвід несприятливої атаки на алгоритми машинного навчання, тому ці підходи не працюють.

Запобігання кібератакам і загрозам за допомогою ШІ. Штучний інтелект був лише різновидом комп'ютеризованої версії людського інтелекту. Те, як функціонує штучний інтелект, схоже на навчання, як це роблять люди, ітеративно знову і знову. Загроза ландшафту безсумнівно розвивається в цьому столітті. Кібер-зловмисники ґрунтуються виключно на фінансових стимулах. Але департамент знайшов новий спосіб запобігання атакам до того, як вони відбудуться, оскільки він більше не може залежати від старих звичайних методів. У цій статті [4] підкреслюється необхідність розвитку навичок кібербезпеки та те, як використання штучних нейронних мереж і алгоритмів машинного навчання може означати покращення навичок. Також включено огляд і визначення соціальної інженерії, роль, яку вона відіграє в мережі та кіберпограбуванні, а також причини та вплив на кіберзлочини.

Рекомендовано превентивні дії та потенційні рішення щодо загроз і вразливостей у соціальній інженерії, виходячи з висновків наведених у статті [5], вразливість залежить від поведінки людини, розумових імпульсів і психологічних схильностей, хоча технології допомагають зменшити вплив атак соціальної інженерії. Хоча література підтверджує інвестиційні ризики в організаційних освітніх таборах через чутливість соціальної інженерії, оптимістично можна сказати, що напади на соціальну інженерію можна зменшити.

Втрати мільярдів доларів спричинені кіберзлочинністю, збоями операційних систем, знищенням секретної інформації, порушенням безпеки мережі та секретності. Безпека комп'ютерних систем стала необхідною для мінімізації впливу та, імовірно, стримування кіберзлочинів у світлі цих злочинів, які здійснюються щодня. У статті наведена дискусія щодо останніх досягнень у використанні наборів даних кібербезпеки для оцінки систем виявлення вторгнень машинного навчання та інтелектуального аналізу даних. Встановлено, що сучасні стандарти кібербезпеки більше не є надійними, оскільки їхні бази даних більше не відповідають сучасним розробкам комп'ютерних технологій. Отже, у 2013 році було запропоновано новий набір даних ADFA Linux (ADFA-LD) для порівняння кібербезпеки, щоб відповідати поточним світовим досягненням у комп'ютерних технологіях для аналізу машинного навчання інтелектуального аналізу даних і систем виявлення вторгнень.

До ADFA-LD включено кращі визначення їхніх атрибутів. Дослідницьке співтовариство використає це дослідження, щоб відмовитися від поточних наборів даних порівняльного аналізу кібербезпеки та почати використовувати нещодавно впроваджений набір даних порівняльного аналізу для ефективної та систематичної оцінки комп'ютера та системи виявлення вторгнень інтелектуального аналізу даних.

Аналіз соціального та інтернет-трафіку важливий для ідентифікації та захисту від кіберзагроз. Розширені підходи до автоматизованого машинного навчання замінюють традиційні підходи, які повертаються до визначених вручну правил. Ця революція прискорюється завдяки масивним наборам даних, які забезпечують моделі машинного навчання з вищою ефективністю. У статті [8] аналізується нещодавнє аналітичне дослідження кібертрафіку

через соціальні мережі та Інтернет, використовуючи набір загальних принципів схожості, відношення та колективних індикацій у контексті моделі, керованої даними. Це не поодиноке бажання, а загальне використання різноманітних мереж і соціальних рухів пояснюється цим. Потоки також мають низку функцій, зокрема фіксований розмір і багато повідомлень між джерелом і одержувачем. Стаття представляє сучасну методологію дослідження та застосування в Інтернет-безпеці, керовані даними соціального та Інтернет-трафіку (DDCS). Підхід до DDCS включає три елементи: збір даних для кібербезпеки, розробку кібербезпеки та моделювання кібербезпеки. Також обговорюються виклики та майбутні шляхи.

Кібератаки становлять серйозну загрозу національній безпеці країні. Сьогодні зростає кількість шкідливих інструментів, які здійснюють численні кібератаки. Знання та інструменти для стримування та пом'якшення атак були заплановані для розвідки кіберзагроз (CTI) і порталу аналізу зловмисного програмного забезпечення. Однак поточні портали CTI та аналізи зловмисного програмного забезпечення звинувачують у надто швидкому реагуванні, оскільки вони залежать від попередніх кібератак для збору даних. Онлайн-форуми хакерів надають проактивному порталу CTI та шкідливих програм нове джерело інформації. Дослідження [8] показує AZ Safe Hacker Assets Portal. Цей веб-сайт збирає та аналізує шкідливі продукти із переважно невикористаних і багатих джерел даних онлайн-груп хакерів, використовуючи найсучасніші методи машинного навчання. У цьому документі обговорюється створення та розробка порталу активів AZ Safe Hacker. Автори також пропонують основні функції порталу, включаючи пошук активів, навігацію та завантаження, перегляд вихідного коду та аналітику порівняння коду, а також інтерактивну інформаційну панель CTI.

За останні десятиліття загрози кібербезпеці зросли. Експерти вважають, що існуючих заходів безпеки скоро буде недостатньо, щоб уникнути поширення більш складних і небезпечних кібератак. Останнім часом у складності кібербезпеки дедалі більше домінують підходи, запозичені зі штучного інтелекту (ШІ) для сприяння автоматизація. У цьому документі [2] дослідники надають короткий огляд і підказки щодо байєсівських програм кібербезпеки, щоб дозволити кількісну оцінку загроз для вищого аналізу ризиків і ситуаційної обізнаності.

**Висновки та пропозиції.** Оскільки кіберзлочини стають дедалі складнішими, підходи до кібербезпеки повинні бути більш надійними та розумними. Це дозволить механізмам захисту приймати рішення в режимі реального часу, щоб ефективно реагувати на складні атаки. Однак підходи штучного інтелекту до боротьби з кіберзлочинністю досі не класифіковані, що вимагає окремого дослідження.

Тому для ефективної боротьби з кіберзлочинністю, дослідники та практики повинні знати існуючі методи кібербезпеки та застосовувати штучний інтелект.



**ЛІТЕРАТУРА**

1. X. Chen et al., «Artificial intelligence-empowered path selection: *A survey of ant colony optimization for static and mobile sensor networks*». IEEE Access. Vol. 8. Pp. 71497 – 71511. 2020.

2. AI Forum of New Zealand and AsureQuality, «Artificial Intelligence for Agriculture in New Zealand». Pp. 40. 2019.

3. Y. Raban and A. Hauptman, «Foresight of cyber security threat drivers and affecting technologies». *Foresight*. Vol. 20. No. 4. Pp. 353 – 363, 2018, doi: 10.1108/FS-02- 2018-0020.

4. X. A. Larriva-Novo, M. Vega-Barbas, V. A. Villagra, and M. Sanz Rodrigo, *«Evaluation of Cybersecurity Data Set Characteristics for Their Applicability to Neural Networks Algorithms Detecting Cybersecurity Anomalies»*. IEEE Access. Vol. 8. Pp. 9005 – 9014. 2020.

5. J. Straub et al., *«CyberSecurity considerations for an interconnected self-driving car system of systems»*. 2017 12th Syst. Syst. Eng. Conf. SoSE 2017. 2017.

6. M. Z. Alom and T. M. Taha, *«Network intrusion detection for cyber security on neuromorphic computing system»*. Proc. Int. Jt. Conf. Neural Networks. Vol. 2017- May. Pp. 3830 – 3837. 2017.

7. K. Shaukat et al., «Performance comparison and current challenges of using machine learning techniques in cybersecurity». *Energies*. Vol. 13. No 10. 2020.

8. R. Coulter, Q. L. Han, L. Pan, J. Zhang, and Y. Xiang, «*Data-Driven Cyber Security in Perspective – Intelligent Traffic Analysis*». IEEE Trans. Cybern., Vol. 50. No 7. Pp. 3081 – 3093. 2020.

9. Fostolovych, V. «Modern tools of the business management system in the field of hotel and restaurant business». *Investytsii: praktyka ta dosvid*. Vol. 11 – 12. Pp. 18 – 25. 2022.

10. V. D. Soni, «Challenges and Solution for Artificial Intelligence in Cybersecurity of the USA». SSRN Electron. J. Pp. 1–17. 2020.

11. N. Scarpato, N. D. Cilia, and M. Romano, *«Reachability Matrix Ontology: A Cybersecurity Ontology»* Appl. Artif. Intell., Vol. 33. No. 7. Pp. 643–655. 2019.

12. Y. Lu, *«Artificial intelligence: a survey on evolution, models, applications and future trends»*, J. Manag. Anal., Vol. 6, No. 1, pp. 1–29, 2019.

13. S. Mahdavifar and A. A. Ghorbani, «Application of deep learning to cybersecurity: A survey». *Neurocomputing*. Vol. 347. Pp. 149–176. 2019.

14. R. Calderon, «The Benefits of Artificial Intelligence in Cybersecurity» Econ. Crime Forensics Capstones. 36. 2021.

15. Гнатієнко Г.М., Снитюк В.Є. Експертні технології прийняття рішень. К.: Маклаут, 2008. 444 с.

16. Глибовець М.М., Олецький О.В. Штучний інтелект: Підручн. для студ. вищ. навч. закладів, що навчаються за спец. «Компютерні науки» та «Прикладна математика». К.: Вид. дім «КМ Академія». 2012. 366 с.

17. Снитюк В.Є. Прогнозування. Моделі, методи, алгоритми. К.: Маклаут, 2018. 364 с.

**REFERENCES**

1. X. Chen et al., *«Artificial intelligence-empowered path selection: A survey of ant colony optimization for static and mobile sensor networks»* IEEE Access. Vol. 8, Pp. 71497 – 71511. 2020.

2. AI Forum of New Zealand and AsureQuality, «Artificial Intelligence for Agriculture in New Zealand». P. 40. 2019.

3. Y. Raban and A. Hauptman, «Foresight of cyber security threat drivers and affecting technologies». *Foresight*. Vol. 20, No 4, Pp. 353 – 363. 2018. doi: 10.1108/FS-02- 2018-0020.

4. X. A. Larriva-Novo, M. Vega-Barbas, V. A. Villagra, and M. Sanz Rodrigo, *«Evaluation of Cybersecurity Data Set Characteristics for Their Applicability to Neural Networks Algorithms Detecting Cybersecurity Anomalies»* IEEE Access. Vol. 8. Pp. 9005 – 9014, 2020.

5. J. Straub et al., *«CyberSecurity considerations for an interconnected self-driving car system of systems»*. 2017 12th Syst. Syst. Eng. Conf. SoSE 2017. 2017.

6. M. Z. Alom and T. M. Taha, *«Network intrusion detection for cyber security on neuromorphic computing system»*. Proc. Int. Jt. Conf. Neural Networks. Vol. 2017- May. Pp. 3830 – 3837. 2017.

7. K. Shaukat et al., *«Performance comparison and current challenges of using machine learning techniques in cybersecurity»*. Energies. Vol. 13, No 10. 2020.

8. R. Coulter, Q. L. Han, L. Pan, J. Zhang, and Y. Xiang, *«Data-Driven Cyber Security in Perspective - Intelligent Traffic Analysis»* IEEE Trans. Cybern. Vol. 50. No 7. Pp. 3081–3093, 2020.

9. Fostolovych, V. «Modern tools of the business management system in the field of hotel and restaurant business». *Investytsii: praktyka ta dosvid*. Vol. 11 – 12. Pp. 18 – 25. 2022.

10. V. D. Soni, *«Challenges and Solution for Artificial Intelligence in Cybersecurity of the USA»*. SSRN Electron. J. Pp. 1 – 17. 2020.

11. N. Scarpato, N. D. Cilia, and M. Romano, «*Reachability Matrix Ontology: A Cybersecurity Ontology»* Appl. Artif. Intell. Vol. 33. No. 7, Pp. 643 – 655, 2019.

12. Y. Lu, *«Artificial intelligence: a survey on evolution, models, applications and future trends»*. J. Manag. Anal., Vol. 6. No. 1, Pp. 1 – 29. 2019.

13. S. Mahdavifar and A. A. Ghorbani, «Application of deep learning to cybersecurity: A survey». *Neurocomputing*. Vol. 347, Pp. 149 – 176. 2019.

14. R. Calderon, «The Benefits of Artificial Intelligence in Cybersecurity». Econ. Crime Forensics Capstones. 36., 2021.

15. Hnatienko H.M., Snityuk V.E. Expert decision-making technologies. K.: Maklaut. 2008. 444 p.

16. Hlybovets M.M., Oletskyi O.V. Artificial intelligence: Manual. for students higher education institutions studying for special «Computer Science» and «Applied Mathematics». - K.: Ed. House «KM Academy». 2012. 366 p.

17. Snytyuk V.E. Prognostication. Models, methods, algorithms. - K.: Maklaut, 2018. 364 p.

**СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 02.12.2023**

*УДК 004.02, 004.08, 004.09*
*DOI: https://doi.org/10.53920/ITS-2023-2-6*

**Ольга Іванівна ТКАЧЕНКО,**
кандидат фізико-математичних наук, доцент,
доцент кафедри інформаційних технологій,
Державний університет інфраструктури та технологій
ORCID ID: 0000-0003-1800-618X
**Максим Володимирович КОВАЛЬЧУК,**
магістрант кафедри інформаційних технологій,
Державний університет інфраструктури та технологій
ORCID ID: 0009-0009-6749-618X

# АВТОМАТИЗАЦІЯ РОЗГОРТАННЯ ХМАРНИХ ФУНКЦІЙ З ВИКОРИСТАННЯМ SERVERLESS ФРЕЙМВОРКУ: ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ

***В наш час проблеми моніторингу працездатності та ефективності відстеження відповідного програмного коду, виявлення помилок та відлагодження програмного коду в безсерверному середовищі є достатньо складними проблемами, актуальність вирішення якої не викликає сумнівів. Шляхи вирішення цих проблем для безсерверних функцій з використанням технології AWS є перспективними. Безсерверні функції активують оточення кожного разу при виклику, тому перший запуск безсерверної функції є досить повільним через необхідність ініціалізації середовища.***

***Розглядаються ключові аспекти розгортання та керування функціями в хмарних середовищах за допомогою Serverless фреймворку. Робиться акцент на значенні і практичних застосуваннях методів автоматизації в контексті сучасних хмарних обчислень та розробки безсерверних (Serverless) застосунків.***

***Метою роботи є аналіз та дослідження проблем, пов'язаних з використанням безсерверних функцій у хмарних середовищах, визначення можливостей використання сучасних принципів, технологій та інструментів, зокрема, Serverless фреймворку для автоматизації розгортання хмарних функцій та розробки відповідного програмного забезпечення.***

***У статті було розглянуто сутність підходу до автоматизації процесів розгортання хмарних функцій на основі використання Serverless фреймворку. Серед переваг використання Serverless фре-***

***ймворку було виділено, зокрема, такі, як зменшення завдань щодо інфраструктури, швидке та достатньо просте розгортання, еластичність, масштабованість та економія коштів, спрощені моніторинг і логування, більше часу безпосередньо на розробку.***

***Було надано загальний огляд Serverless фреймворку та наведено інструкцію (послідовність дій) щодо створення та розгортання за його допомогою безсерверних функцій в хмарному середовищі AWS. Було розглянуто приклади використання Serverless фреймворку при розв'язанні практичних задач, зокрема створення та розгортання функції для відправлення push-нотифікацій. Запропонований підхід може бути корисним при розробці застосунків з використанням хмарних технологій.***

***Ключові слова**: хмарні обчислення, хмарне середовище, фреймворк, Serverless, безсерверна функція, push-нотифікації, хмарний провайдер, Node.js, AWS.*

**Olha TKACHENKO**
PhD of physical and mathematical sciences, associate professor,
associate professor at the department
of information technologies,
State University of Infrastructure and Technology
**Maksym KOVALCHUK**
Undergraduate at the department of information technologies,
State University of Infrastructure and Technology

## AUTOMATION OF THE DEPLOYMENT OF CLOUD FUNCTIONS USING THE SERVERLESS FRAMEWORK: PROBLEMS AND PERSPECTIVES

***Nowadays, the problems of monitoring the performance and effectiveness of tracking the corresponding software code, detecting errors and debugging the software code in a serverless environment are quite complex problems, the urgency of the solution of which is beyond doubt. Ways to solve these problems for serverless functions using AWS technology are promising. Serverless functions activate the environment each time they are called, so the first run of a serverless function is quite slow due to the need to initialize the environment.***

***Key aspects of deploying and managing features in cloud environments using the Serverless framework are covered. Emphasis***

***is placed on the importance and practical applications of automation methods in the context of modern cloud computing and the development of serverless applications.***

***The purpose of the work is the analysis and research of problems related to the use of serverless functions in cloud environments, the determination of the possibilities of using modern principles, technologies and tools, in particular, the Serverless framework for automating the deployment of cloud functions and the development of appropriate software.***

***The article considered the essence of the approach to automating processes of deployment of cloud functions based on the use of the Serverless framework. Among the advantages of using the Serverless framework, such as reduction of infrastructure tasks, fast and fairly simple deployment, elasticity, scalability and cost savings, simplified monitoring and logging, more time directly for development were highlighted.***

***An overview of the Serverless framework was provided and a step-by-step guide for creating and deploying serverless features in the AWS cloud environment was provided. Examples of the use of the Serverless framework in solving practical problems were considered, in particular, the creation and deployment of a function for sending push notifications. The proposed approach can be useful in the development of applications using cloud technologies.***

***Keywords:*** *cloud computing, cloud environment, framework, Serverless, serverless function, push notifications, cloud provider, Node.js, AWS.*

**Постановка проблеми.** Serverless [1], як концепція, дає можливість розробникам концентруватися лише на функціональності їхніх додатків, не турбуючись про адміністрування серверів та внутрішньої інфраструктури. Проте, процес розгортання безсерверних функцій у хмарних середовищах залишається актуальною проблемою, яка потребує свого вирішення. Автоматизація розгортання безсерверних функцій у Serverless фреймворку [2] допомагає вирішити цю проблему, роблячи процес більш зрозумілим та ефективним.

Вирішення проблеми автоматизації розгортання безсерверних функцій передбачає, зокрема:

- аналіз можливостей та переваг автоматизації розгортання безсерверних функцій у хмарному середовищі;

- дослідження різних підходів, інструментів і методів, які можна використовувати для автоматизації цього процесу, включаючи IaC [3], Serverless фреймворк та інші.

Для демонстрації запропонованого підходу треба навести приклади практичного використання Serverless фреймворку для вирішення різноманітних задач.

**Аналіз останніх досліджень і публікацій.** В наш час безсерверні технології активно застосовуються в різних галузях, включаючи веброзробку, обробку даних, Інтернет речей (*Internet of things, IoT*) [4] та інші. Це свідчить про широкий спектр можливостей, які надає підхід з використанням Serverless.

Слід відмітити важливість розуміння того, як вирішувати проблеми, що виникають при використанні безсерверних хмарних функцій.

Дуже важливо забезпечити ефективне розгортання функцій при зростанні навантаження та автоматично масштабувати обчислювальні ресурси. Зокрема, в [5] було розглянуто можливості автоматизації розгортання та масштабування хмарних функцій.

Виявлення помилок та відлагодження програмного коду в безсерверному середовищі може бути складною проблемою. Тому проблема моніторингу працездатності та ефективності відстеження відповідного програмного коду завжди постає перед розробниками. Шляхи її вирішення для безсерверних функцій з використанням технології AWS Lambda Insights [6] були розглянуті в [7].

За замовчуванням, безсерверні функції активують оточення кожного разу при відповідному виклику, тому перший запуск безсерверної функції (Cold Start) [8] може бути досить повільним через необхідність ініціалізації спочатку середовища. Можливості оптимізації цього процесу та його прискорення розглянуті в [9].

В [10] розглядається конкретний технічний стек реалізації безсерверних функцій з використанням Node.js (серверного середовища виконання JavaScript-коду) [11], AWS Lambda (безсерверний обчислювальний сервіс від компанії Amazon) [12] та Serverless фреймворку.

**Мета статті** полягає в аналізі та дослідженні проблем, пов'язаних з використанням безсерверних функцій у хмарних середовищах, визначення можливостей використання сучасних принципів, технологій та інструментів, зокрема, Serverless фреймворку

для автоматизації (спрощення та оптимізації) розгортання хмарних функцій та розробки відповідного програмного забезпечення.

Досягнення цієї мети забезпечується вирішенням, зокрема, наступних завдань:

- визначення сутності безсерверного підходу, його основних переваг і недоліків у порівнянні зі статичним сервером, а також можливих сфер використання;
- визначення складових компонентів Serverless фреймворку (зокрема, визначення того, з яких частин складається цей фреймворк та конфігураційні yaml файли);
- дослідження проблем автоматизації розгортання хмарних функцій та шляхів їх вирішення;
- розробка покрокової інструкції для розгортання хмарних функцій за допомогою Serverless фреймворку;
- Тестування запропонованого підходу при вирішенні практичних задач за допомогою використання Serverless фреймворку.

Мета і завдання статті спрямовані на просування інноваційних підходів та технологій, які можуть підтримати стійкий розвиток хмарних технологій та забезпечити високий рівень розробленого програмного забезпечення, яке вирішує конкретні практичні проблеми.

**Виклад основного матеріалу дослідження.** Безсерверні функції — нова парадигма в хмарних обчисленнях, яка привернула значну увагу в останні роки завдяки своїм обіцянкам спростити розгортання та керування застосунками.

Функція, як послуга (FaaS) [13], – це підхід, при якому розробники можуть розгортати та запускати окремі функції (зв'язні елементи коду, так звані «одиниці» коду) без необхідності «напряму» керувати серверною інфраструктурою.

На відміну від традиційного серверного підходу, який передбачає надання та підтримку серверів для статичного розміщення застосунків, безсерверні обчислення повністю абстрагують рівень інфраструктури, дозволяючи розробникам зосередитися виключно на написанні коду.

При використанні такої моделі, обчислювальні ресурси автоматично розподіляються та масштабуються по принципу on demand [14] для виконання коду у відповідь на певні події, ситуаційні стани чи тригери.

(Під тригером будемо розуміти певну подію чи предмет, при спрацьовуванні якого, починає відбуватися дія безпосередньо або побічно пов'язана з ним, тобто це те, що стає так би мовити спусковим гачком).

Розробники пишуть програмні коди для функцій, що розгортаються та виконуються в ефемерних контейнерах без збереження стану.

Ці функції можуть тригеритись (відповідати певними діями, спонукати на реакцію) на виконання різних подій, такими речами, як HTTP-запити, модифікування в базі даних або повідомлення з черги подій.

Не менш важливою перевагою є те, що безсерверні функції дотримуються принципу pay-as-you-go, згідно з яким плата користувачам виставляється на основі фактичного часу виконання та ресурсів, спожитих їхніми функціями.

У традиційній інфраструктурі невикористані ресурси та потужності серверів можуть бути занадто дорогими. Безсерверні ж функції не потребують додаткових витрат тоді, коли вони не виконуються. Це особливо вигідно для програм із непередбачуваним робочим навантаженням.

Тим не менш, безсерверні функції мають і певні недоліки, зокрема, один з яких – автоматизація їх керування та розгортання.

Для вирішення цієї проблеми було створено Serverless фреймворк [15, 16]. Він пропонує уніфікований файл конфігурації, який об'єднує в собі всі параметри для визначення ресурсів та розгортання функцій.

Все це допомагає уніфікувати процес розгортання, а також спростити керування функціями, ресурсами і тригерами подій в рамках одного проєкту.

Фреймворк Serverless підтримує різних хмарних провайдерів, зокрема таких, як AWS [10, 12], Azure [27], Google Cloud [28] тощо.

Розробники можуть написати програмний код один раз і розгорнути його в кількох хмарних середовищах, підвищуючи портативність і зменшуючи прив'язаність (і, відповідно, залежність) до конкретного провайдера.

У випадку, коли треба динамічно розширити базовий функціонал фреймворку, використовується велика екосистема існуючих сучасних плагінів [15]. Вони підтримуються широкий спектр випадків, зокрема:

- оптимізація коду;
- підвищення безпеки;
- автоматизація розгортання;
- інтеграція з різними сторонніми службами.

Розробники можуть легко додавати та налаштовувати плагіни відповідно до потреб на проекті.

Фреймворк Serverless забезпечує інтерфейс командного рядка (*Serverless CLI*) [16], який спрощує керування проєктами. Розробники, щоб розгортати функції, викликати їх і керувати різними аспектами своїх безсерверних програм, можуть використовувати такі команди, як

- *sls deploy*;
- *sls invoke*.

Розглянемо, як за допомогою фреймворку можна створити та розгорнути безсерверну хмарну функцію. Для розробленого програмного продукту, що демонструє в статті переваги запропонованого підходу, було використано мову Node.js (JavaScript) [11] як основну мову програмування і AWS [10, 12, 22] як хмарний провайдер.

Розглянемо процес створення і розгортання безсерверної хмарної функції:

- Інсталювати Node.js і NPM [17];
- Інсталювати Serverless фреймворк як глобальний модуль. Це можна зробити за допомогою команди:

*npm i-g serverless*

- Ініціалізувати новий проєкт, виконавши команду:

*serverless create –template aws-nodejs–path <serverless-project-name>*

Налаштувати основний конфігураційний файл *serverless.yml* у каталозі розробляємого проєкту, включивши функції, тригери подій та усі необхідні ресурси. Детально властивості конфігураційного файлу описані в [18].

У конфігураційному файлі (рис. 1):

– поле *service* визначає назву проєкту;
– поле *provider* визначає тип хмарного провайдера і середу виконання коду;
– поле *functions* визначає перелік функцій.

```
service: my-serverless-project

provider:
  name: aws
  runtime: nodejs14.x

functions:
  hello:
    handler: handler.hello
```

**Рис. 1. Конфігураційний файл для хмарної функції**

Джерело: розробка авторів

- Створити файл *handler.js*;
- Експортувати асинхронну функцію *hello* (рис. 2). Функція має один вхідний параметр event, який містить інформацію про вхідну подію, а саме:
  – контекст виконання;
  – тип події;
  – властивості, передані клієнтом в функцію напряму для обробки.

```
// handler.js

module.exports.hello = async (event) => {
  // write your code here
};
```

**Рис. 2. Асинхронна обробка подій в середовищі Node.js**

Джерело: розробка авторів

- Розгорнути безсерверну службу, виконавши наступні команди:
  – *serverless print;*
  – *serverless deplo.*

Команда *serverless print* надає усі зміни без розгортання служби, що сприяє перевірці коректності конфігурації перед розгортанням безсерверної служби.

Команда *serverless deplo* ініціює виконання таких дій, як:

– упакування авторського програмного коду;
– створення необхідних ресурсів в AWS;
– розгортання відповідної хмарної функції.

- Викликати функцію: serverless *invoke -f hello*.

Безсерверні функції є універсальним засобом, що широко застосовується на практиці. Зокрема, використання зазначених функцій доцільно використовувати у таких випадках, як ті, що наведені нижче.

- Обробка логіки автентифікації та авторизації користувачів у вебзастосунках. Вони можуть:
  – створювати та верифікувати токени авторизації;
  – застосовувати політики контролю доступу;
  – інтегруватися зі сторонніми постачальниками токенів (наприклад, таких як, AWS Cognito [19]).
- Планування задач, які мають виконуватися через певні проміжки часу, зокрема, таких як:
  – регулярне резервне копіювання даних;
  – створення звітів;
  – синхронізація даних між системами.
- Надсилання користувачам сповіщень в режимі реального часу через різні канали, зокрема, такі як:
  – електронна пошта;
  – SMS;
  – push-повідомлення;
  – платформи обміну повідомленнями.
- Швидка обробка та аналіз безперервних потоків даних з пристроїв, що підтримують Інтернет речей [4], ініціюючи відповіді на основі даних, що отримуються від різноманітних приладів (датчиків, лічильників, контролерів тощо), чи подій пристрою.

Розглянемо використання *Serverless* фреймворку при написанні та подальшому розгортанні безсерверної функції для надсилання APNS-сповіщень [20] на мобільні пристрої користувачів.

Для цього насамперед потрібно, використовуючи отриману інформацію, сформувати та надіслати нотифікацію в AWS SNS-topic [21] для відповідного конкретного користувача.

На рис. 3 зображено конфігураційні файли проєкту, що розглядається.

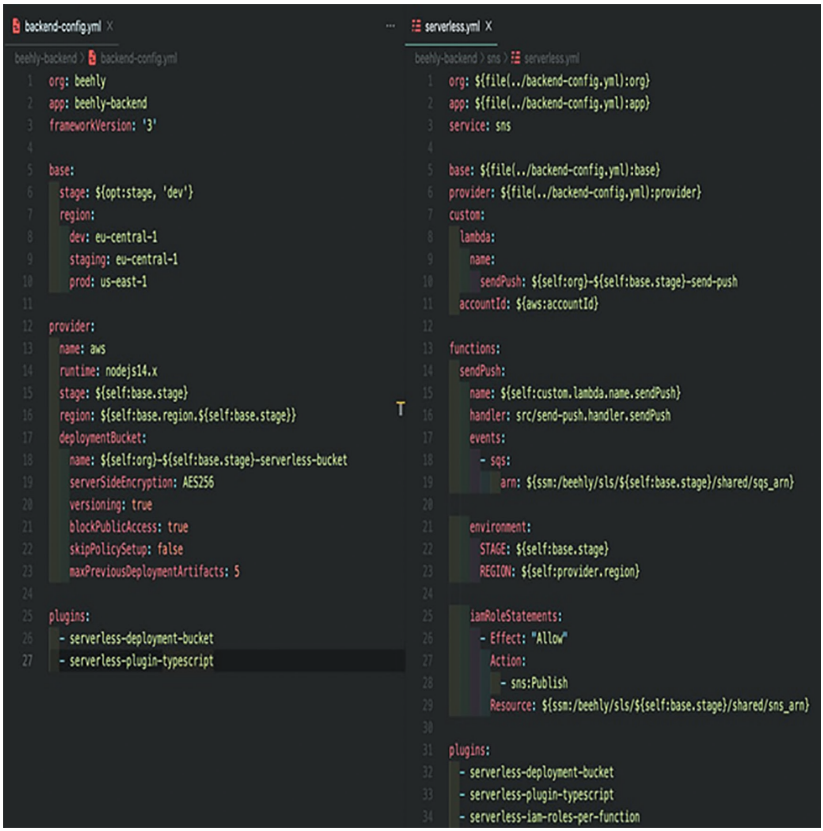


**Рис. 3. Конфігураційні файли функції для надсилання push-нотифікацій на мобільні пристрої користувачів**

Джерело: розробка авторів

Ці конфігураційні файли проєкту містять, зокрема:

- Файл *backend-config.yml*, який є загальним для усього проєкту і я якому зазначені:
  - *region* (регіони) для розгортання функцій (відповідно до кожного *environmental (оточення, оточуючого середовища)*);

  - тип провайдера;
  - використовувана мова програмування;
  - загальні плагіни;
  - інформація про deployment bucket – місце, куди буде завантажено AWS CloudFormation стек [22] кожної із заданих функцій (хмарних функцій).
- Файл *serverless.yml* є специфічним для даної функції. В ньому вказуються, зокрема:
  - назва функції;
  - посилання на відповідний обробник в TypeScript-файлі;
  - плагіни;
  - змінні *environmental*;
  - унікальний ідентифікатор AWS SQS-черги [23] (з якої надходитимуть дані про події);
  - AWS IAM-роль [24] (для того, щоб функція мала можливість публікувати нотифікації в SNS-topic).

Синтаксис Serverless фреймворку дозволяє динамічно вказувати значення в конфігураціних файлах, отримуючи їх, наприклад, зі сховища параметрів AWS Systems Manager [25].

На рис. 4 зображено асинхронний обробник подій для визначеної хмарної функції, написаний мовою програмування TypeScript.

Цей обробник здійснює *автоматичний збір даних* (вхідних повідомлень) з SQS-черги *та їхнє подальше структурування,* потім формує push-нотифікації згідно із заданим форматом і відправляє команди на публікацію нотифікацій у SNS-topic.

Для того, щоб відправити push-нотифікацію, потрібно лише створити і надіслати повідомлення зі стороннього сервісу в SQS-чергу (із заданим в конфігураційному файлі *arn*).

Це спричинить виконання хмарної функції, яка, в свою чергу, створить потрібну нотифікацію, що надійде користувачеві на мобільний пристрій.

Слід зазначити, що для того, щоб SNS мав можливість надсилати APNS сповіщення, потрібно надати ключ або сертифікат застосунку, обидва з яких можна отримати зі свого облікового запису розробника Apple.

```
const client = new SNSClient({ region: process.env.REGION });

export async function sendPush(event) {
  const records = event.Records;
  const commands: PublishCommand[] = [];

  for (const record of records) {
    const parsed = parseQueueMessage(record);
    if (parsed) {
      const { deviceEndpoint, payload, ...rest } = parsed;
      const applePushNotificationPayload: ApplePushNotificationPayload = {
        aps: { alert: payload, ...rest },
      };
      const pushNotificationPayload: PushNotification = {
        [DEFAULT_PUSH_NOTIFICATION_KEY]: DEFAULT_MESSAGE,
        [APPLE_PUSH_NOTIFICATION_BODY_KEY]: JSON.stringify(
          applePushNotificationPayload
        ),
        [APPLE_SANDBOX_PUSH_NOTIFICATION_BODY_KEY]: JSON.stringify(
          applePushNotificationPayload
        ),
      };
      commands.push(
        new PublishCommand({
          TargetArn: deviceEndpoint,
          Message: JSON.stringify(pushNotificationPayload),
          MessageStructure: 'json',
        })
      );
    }
  }
  await Promise.all(commands.map((command) => client.send(command)));
  return event;
}
```

**Рис. 4. TypeScript код функції для надсилання push-нотифікацій на мобільні пристрої користувачів**

Джерело: розробка авторів

**Висновки та пропозиції.** У статті було розглянуто сутність підходу до автоматизації процесів розгортання хмарних функцій на основі використання Serverless фреймворку. Серед переваг використання Serverless фреймворку слід виділити, зокрема, такі:

- Зменшення завдань щодо інфраструктури, завдяки чому розробники можуть уникнути необхідності управління серверами, резервування ресурсів та налаштування моніторингу; все це забезпечує хмарний постачальник, а розробники можуть займатися лише програмним кодом свого вебдодатку.

- Швидке та достатньо просте розгортання, що особливо корисно для розробників, які мають за мету швидкий випуск продукту на ринок або перевірку нових функцій продукту.
- Еластичність, масштабованість та економія коштів (Serverless автоматично масштабується відповідно до навантаження), що обумовлює те, що користувач сплачує тільки за використані ресурси (виконані функції), має високий рівень доступності до ресурсів та має можливість зменшити витрати на відповідну інфраструктуру.
- Спрощені моніторинг і логування, що спрощує відстеження розробниками ходу виконання програмного коду.
- Збільшення часу безпосередньо на розробку, бо Serverless надає розробникам можливість зосередитися на функціональності та інноваціях свого програмного продукту (вебдодатку), замість витрати часу на вирішення відповідних потрібних адміністративних завдань.

Крім того в статті надано загальний огляд Serverless фреймворку та наведено інструкцію (послідовність дій) щодо створення та розгортання за його допомогою. безсерверних функцій в хмарному середовищі AWS.

Було також розглянуто приклади використання Serverless фреймворку при розв'язанні практичних задач, зокрема створення та розгортання функції для відправлення push-нотифікацій на мобільні пристрої користувачів.



**ЛІТЕРАТУРА**

1. Serverless. URL: https://www.cloudflare.com/learning/serverless/what-is-serverless (дата звернення: 11.10.2023).

2. Serverless Framework. URL: https://www.serverless.com (дата звернення: 11.10.2023).

3. Infrastructure as Code. URL: https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code-iac (дата звернення: 12.10.2023).

4. Internet of Things. URL: https://www.ibm.com/topics/internet-of-things (дата звернення: 12.10.2023).

5. Serverless Deployment: Best Practices and Examples. URL: https://www.techmagic.co/blog/how-serverless-deployment-works-best-practices-principles-examples (дата звернення: 14.10.2023).

6. AWS Lambda Insights. URL: https://docs.aws.amazon.com/AmazonCloudWatch/ latest/monitoring/Lambda-Insights.html (дата звернення: 13.10.2023).

7. Using Amazon CloudWatch Lambda Insights to Improve Operational Visibility. URL: https://aws.amazon.com/blogs/aws/using-amazon-cloudwatch-lambda-insights-to-improve-operational-visibility (дата звернення: 13.10.2023).

8. AWS Lambda Cold Start. URL: https://blog.octo.com/cold-start-warm-start-with-aws-lambda (дата звернення: 12.10.2023).

9. How to Improve Serverless Function Cold Start Performance. URL: https://vercel.com/guides/how-can-i-improve-serverless-function-lambda-cold-start-performance-on-vercel (дата звернення: 12.10.2023).

10. Deploying a Simple Serverless Node.js Application on AWS. URL: https://hackernoon.com/deploying-a-simple-serverless-nodejs-application-on-aws-lambda-functions (дата звернення: 14.10.2023).

11. Node.js. URL: https://nodejs.org/en/docs (дата звернення: 15.10.2023).

12. AWS Lambda. URL: https://aws.amazon.com/lambda (дата звернення: 15.10.2023).

13. Function as a Service. URL: https://itglobal.com/ru-ru/company/glossary/faas (дата звернення: 16.10.2023).

14. Scaling on demand. URL: https://www.simpleservers.co.uk/scale-on-demand (дата звернення: 14.10.2023).

15. Serverless Framework Plugins. URL: https://www.serverless.com/plugins (дата звернення: 15.10.2023).

16. Serverless Framework CLI. URL: https://www.serverless.com/framework/ docs/providers/aws/cli-reference/deploy (дата звернення: 12.10.2023).

17. Node Package Manager. URL: https://www.npmjs.com. (дата звернення: 12.10.2023).

18. Serverless.yml reference. URL: https://www.serverless.com/framework/docs/ providers/aws/guide/serverless.yml (дата звернення: 18.10.2023).

19. AWS Cognito. URL: https://docs.aws.amazon.com/cognito-user-identity-pools/latest/APIReference/Welcome.html (дата звернення: 15.10.2023).

20. Apple Push Notifications. URL: https://developer.apple.com/notifications. (дата звернення: 12.10.2023).

21. AWS SNS. URL: https://aws.amazon.com/sns (дата звернення: 11.10.2023).

22. AWS CloudFormation. URL: https://aws.amazon.com/cloudformation (дата звернення: 14.10.2023).

23. AWS SQS. URL: https://www.javatpoint.com/aws-sqs (дата звернення: 17.10.2023).

24. AWS IAM. URL: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html (дата звернення: 17.10.2023).

25. AWS Systems Manager. URL: https://aws.amazon.com/systems-manager/features (дата звернення: 17.10.2023).

26. Mobile device endpoint. URL: https://heimdalsecurity.com/blog/mobile-device-management (дата звернення: 16.10.2023).

27. What is Azure? URL: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/ (дата звернення: 16.10.2023).

28. Google Cloud overview. URL: https://cloud.google.com/docs/overview (дата звернення: 16.10.2023).

**REFERENCES**

1. Serverless, available at: https://www.cloudflare.com/learning/serverless/what-is-serverless (Accessed 11 October 2023).

2. Serverless Framework, available at: https://www.serverless.com. (Accessed 11 October 2023).

3. Infrastructure as Code, available at: https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code-iac (Accessed 12 October 2023).

4. Internet of Things, available at: https://www.ibm.com/topics/internet-of-things (Accessed 12 October 2023).

5. Serverless Deployment: Best Practices and Examples, available at: https://www.techmagic.co/blog/how-serverless-deployment-works-best-practices-principles-examples (Accessed 14 October 2023).

6. AWS Lambda Insights, available at: https://docs.aws.amazon.com/AmazonCloudWatch/ latest/monitoring/Lambda-Insights.html. (Accessed 13 October 2023).

7. Using Amazon CloudWatch Lambda Insights to Improve Operational Visibility, available at: https://aws.amazon.com/blogs/aws/using-amazon-cloudwatch-lambda-insights-to-improve-operational-visibility (Accessed 13 October 2023).

8. AWS Lambda Cold Start, available at: https://blog.octo.com/cold-start-warm-start-with-aws-lambda (Accessed 12 October 2023).

9. How to Improve Serverless Function Cold Start Performance, available at: https://vercel.com/guides/how-can-i-improve-serverless-function-lambda-cold-start-performance-on-vercel (Accessed 12 October 2023).

10. Deploying a Simple Serverless Node.js Application on AWS, available at: https://hackernoon.com/deploying-a-simple-serverless-nodejs-application-on-aws-lambda-functions (Accessed 14 October 2023).

11. Node.js, available at: https://nodejs.org/en/docs (Accessed 15 October 2023).

12. AWS Lambda, available at: https://aws.amazon.com/lambda (Accessed 15 October 2023).

13. Function as a Service, available at: https://itglobal.com/ru-ru/company/glossary/faas (Accessed 16 October 2023).

14. Scaling on demand, available at: https://www.simpleservers.co.uk/scale-on-demand (Accessed 14 October 2023).

15. Serverless Framework Plugins, available at: https://www.serverless.com/plugins (Accessed 15 October 2023).

16. Serverless Framework CLI, available at: https://www.serverless.com/framework/ docs/providers/aws/cli-reference/deploy (Accessed 12 October 2023).

17. Node Package Manager, available at: https://www.npmjs.com (Accessed 12 October 2023).

18. Serverless.yml reference, available at: https://www.serverless.com/framework/docs/ providers/aws/guide/serverless.yml (Accessed 18 October 2023).

19. AWS Cognito, available at: https://docs.aws.amazon.com/cognito-user-identity-pools/latest/APIReference/Welcome.html (Accessed 15 October 2023).

20. Apple Push Notifications, available at: https://developer.apple.com/notifications (Accessed 12 October 2023).

21. AWS SNS, available at: https://aws.amazon.com/sns. (Accessed 11 October 2023).

22. AWS CloudFormation, available at: https://aws.amazon.com/cloudformation (Accessed 14 October 2023).

23. AWS SQS, available at: https://www.javatpoint.com/aws-sqs. (Accessed 17 October 2023).

24. AWS IAM, available at: https://docs.aws.amazon.com/IAM/latest/UserGuide/ id_roles.html (Accessed 17 October 2023).

25. AWS Systems Manager, available at: https://aws.amazon.com/systems-manager/features (Accessed 17 October 2023).

26. Mobile device endpoint, available at: https://heimdalsecurity.com/blog/mobile-device-management (Accessed 16 October 2023).

27. What is Azure? available at: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/ (Accessed 16 October 2023).

28. Google Cloud overview, available at: https://cloud.google.com/docs/overview (Accessed 16 October 2023).

**СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 25.09.2023**

*УДК 004.9:658*
*DOI: https://doi.org/10.53920/ITS-2023-2-7*

**Андрій Вікторович ЛЕМЕШКО,**
доктор філософії, доцент,
доцент кафедри комп'ютерної інженерії,
Державний університет інформаційно-комунікаційних технологій
ORCID ID: 0000-0001-8003-3168
**Артем Васильович АНТОНЕНКО,**
кандидат технічних наук, доцент,
доцент кафедри комп'ютерної інженерії,
Державний університет інформаційно-комунікаційних технологій
ORCID ID: 0000-0001-9397-1209
**Олександр Максимович МАТВІЙЧУК,**
магістр кафедри комп'ютерної інженерії, ДУІКТ
ORCID ID: 0009-0002-8198-9668
**Олександр Сергійович ДМИТРЕНКО,**
магістр кафедри комп'ютерної інженерії, ДУІКТ
ORCID ID: 0009-0005-2483-600X
**Вадим Юрійович БЕРЕЗДЕЦЬКИЙ,**
магістр кафедри комп'ютерної інженерії, ДУІКТ
ORCID ID: 0009-0004-1007-975X

# УПРАВЛІННЯ ТРАФІКОМ В ГІБРИДНІЙ ПРОГРАМНО-ВИЗНАЧЕНІЙ МЕРЕЖІ

***У статті досліджується питання безпеки та доступності в гібридних програмно-визначених мережах (SDN) з використанням контролера та протоколу маршрутизації EIGRP. Особлива увага приділяється методу управління навантаженням з використанням протоколу onePK. У статті розглядається проблема управління трафіком в мережах, яка є актуальною у зв'язку зі зростанням кількості підключених пристроїв та збільшенням об'єму переданих даних. Пропонується використовувати гібридну SDN мережу, яка поєднує в собі переваги традиційної та програмно-визначеної мережі. Для управління трафіком в такій мережі використовується контролер, який керує роботою мережі та маршрутизацією. Для забезпечення безпеки пропонується використовувати протокол маршрутизації EIGRP, який дозволяє забезпечити безпеку та надійність передачі даних. Окрім цього, протокол EIGRP дозволяє використовувати балансування навантаження, що забезпечує рівномірне розподілення навантаження між маршрутизаторами та покращує швидкість передачі даних. Для ефективного управління навантаженням в гібридній SDN мережі пропонується***

***використовувати метод управління трафіком з використанням протоколу onePK. Цей протокол дозволяє збільшити продуктивність мережі та забезпечити більш ефективне використання ресурсів мережі. Крім того, використання протоколу onePK дозволяє простіше та швидше налаштування та управління мережею, що забезпечує більшу доступність та надійність роботи мережі. У статті розглядаютьсяє актуальні проблеми управління трафіком в мережах та пропонує ефективні методи розв'язання цих проблем з використанням гібридної програмно-визначеної мережі, контролера, протоколу маршрутизації EIGRP та протоколу onePK. Застосування запропонованих методів дозволяє забезпечити безпеку та доступність мережі, а також покращити швидкість та ефективність передачі даних. Робота присвячена методу управління трафіком у високонавантаженій гібридній програмно-визначеній мережі. Як метод дослідження застосовується експеримент. У статті наведено універсальні програмні моделі реалізації запропонованого методу управління трафіком. Наведено формалізовану модель запропонованого методу управління трафіком, а також його універсальну програмну модель реалізації.***

***Ключові слова:*** *доступність, безпека, програмно-визначена мережа, контролер, навантаження, EIGRP, onePK.*

**Andriy LEMESHKO**
Doctor of Philosophy, Associate Professor,
associate professor of the department of computer engineering,State University of information and communication technologies
**Artem ANTONENKO**
candidate of technical sciences, associate professor,
associate professor of the department of computer engineering,
State University of information and communication technologies
**Oleksandr MATVIICHUK,**
**Oleksandr DMYTRENKO,**
**Vadym BEREZDETSKYI**
masters of computer engineering department,
State University of information and communication technologies

## MODELING OF WIRELESS NETWORKS IN OMNET ++ ENVIRONMENT INVOLVING INET FRAMEWORK

***The paper investigates security and availability in hybrid software-defined networks (SDN) using a controller and the EIGRP routing protocol. Special attention is paid to the method of load management using the onePK protocol. The article considers the problem of traffic management***

***in networks, which is relevant in connection with the growth of the number of connected devices and the increase in the volume of transmitted data. It is proposed to use a hybrid SDN network, which combines the advantages of a traditional and software-defined network. To manage traffic in such a network, a controller is used, which controls the operation of the network and routing. To ensure security, it is suggested to use the EIGRP routing protocol, which allows you to ensure the security and reliability of data transmission. In addition, the EIGRP protocol allows the use of load balancing, which ensures an even distribution of the load between routers and improves data transfer rates. For effective load management in a hybrid SDN network, it is suggested to use the traffic management method using the onePK protocol. This protocol allows you to increase network performance and ensure more efficient use of network resources. In addition, the use of the onePK protocol allows easier and faster network configuration and management, which ensures greater availability and reliability of network operation. The article examines current problems of traffic management in networks and offers effective methods for solving these problems using a hybrid software-defined network, a controller, the EIGRP routing protocol, and the OnePK protocol. The application of the proposed methods allows to ensure the security and availability of the network, as well as to improve the speed and efficiency of data transmission. The work is devoted to the method of traffic management in a highly loaded hybrid software-defined network. An experiment is used as a research method. The article provides universal software models for implementing the proposed traffic management method. A formalized model of the proposed traffic management method is given, as well as its universal software implementation model.***

***Keywords:*** *availability, security, software-defined networking, controller, EIGRP, load, onePK.*

**Постановка проблеми.** В даний час концепція SDN (Software Defined Network або програмно-визначена мережа) стрімко завойовує світ мережевих технологій. Швидке зростання обсягу трафіку призводить до зростання інфраструктури, яка дозволить його опрацювати. Це, у свою чергу, призводить до того, що управління будь-якими мережами стає громіздким, малоефективним і складним.

Програмно-визначені мережі надають можливості для більш ефективного та гнучкого управління мережею, зокрема, для управління трафіком. Однак, використання SDN вимагає вирішен-

ня ряду проблем, зокрема, забезпечення доступності та безпеки мережі. Крім того, потрібні ефективні методи управління трафіком в SDN, які б дозволяли забезпечувати ефективність мережі та запобігали перевантаженням мережі.

У зв'язку з цим, постановка проблеми полягає в необхідності розробки методу управління трафіком в гібридній програмно-визначеній мережі з використанням контролера та протоколу маршрутизації EIGRP, який би дозволяв забезпечувати ефективність та безпеку мережі, а також уникати перевантажень мережі. Для досягнення цієї мети необхідно дослідити доступні методи управління трафіком в SDN, зокрема з використанням контролерів та протоколів маршрутизації, та знайти найефективніші підходи для застосування їх у гібридній програмно-визначеній мережі.

**Аналіз останніх досліджень і публікацій.** На сьогоднішній день тема програмно-визначених мереж є досить актуальною в галузі телекомунікацій та мережевих технологій. У світі багато вчених та фахівців займаються дослідженнями та розробками у цій галузі. Розглянемо кілька останніх джерел з цієї теми, які надруковані в українських та зарубіжних виданнях.

Одним з останніх джерел з даної теми є стаття «Програмно-визначені мережі: переваги, недоліки, можливості» авторів Клименка М. та Шмарди О., яка була опублікована в журналі «Інформаційні технології та комп'ютерна інженерія» в 2017 році. У статті автори описують архітектуру програмно-визначених мереж та їх функції, а також наводять приклади застосування таких мереж у практичних ситуаціях. Стаття містить важливу інформацію про особливості програмно-визначених мереж та їх переваги в порівнянні з традиційними мережами.

Ще одним джерелом є стаття «Управління трафіком в програмно-визначених мережах» авторів Маринченко І. та Нікітіна А., яка була опублікована в журналі «Комп'ютерні системи та мережі» в 2018 році. У статті автори розглядають роль програмно-визначених мереж у підвищенні ефективності управління інформаційними потоками в організаціях. Інше джерело на дану тему – стаття «Програмно-визначені мережі: аналіз стану розробки» авторів Шевчука С. та Мелешко В., яка була опублікована в журналі «Комп'ютерні науки та інформаційні технології» в 2017 році. У статті автори досліджують питання безпеки програмно-визначених мереж та наводять різні методи її забезпечення. Вони

аналізують потенційні загрози та вразливості програмно-визначених мереж та надають пропозиції щодо їх протидії.

Другим джерелом з даної теми є стаття «Основні напрями розвитку програмно-визначених мереж» авторів Арлова О. та Селезньова Д., яка була опублікована в журналі «Науковий вісник Національного гірничого університету» в 2019 році. У статті автори проводять порівняльний аналіз двох протоколів маршрутизації OSPF та EIGRP в програмно-визначених мережах. Вони досліджують переваги та недоліки кожного з цих протоколів та надають рекомендації щодо їх використання.

Загалом, дослідження та розробки в галузі програмно-визначених мереж здійснюються як українськими вченими та фахівцями, так і дослідниками з інших країн. Це свідчить про високий інтерес до даної теми у науковому та практичному середовищі. Важливо продовжувати дослідження та розробки в галузі програмно-визначених мереж з метою покращення доступності та безпеки мереж та підвищення ефективності управління трафіком [1-20].

**Метою статті** є розгляд методу управління трафіком в гібридній програмно-визначеній мережі з використанням контролера та протоколу маршрутизації EIGRP.

**Виклад основного матеріалу дослідження.** У роботах розглянутих вище джерел запропоновано рішення щодо забезпечення механізму маршрутизації в SDN з урахуванням виконання вимог до якості обслуговування (QOS) для максимально можливої кількості потоків та в умовах змінного навантаження.

На основі висновків, розглянемо гібридну модель SDN з урахуванням протоколу маршрутизації EIGRP.

Цей протокол відноситься до типу distance-vector, однак, на відміну від RIP та IGRP, використовує як основу своєї роботи алгоритм дифузійних обчислень. Подробиці роботи даного алгоритму наведені у [6] та [7].

Маршрутизатори, що беруть участь у роботі EIGRP, обмінюються інформацією про префікси, що знаходяться в таблиці маршрутизації кожного з них. Інформація про префікси містить:

- IP-адреса мережі;
- маску підмережі;
- смугу пропускання сегмента цієї мережі;
- затримку на інтерфейсі маршрутизатора, який представляє даний сегмент мережі;

- навантаження, яке діє на інтерфейсі, що представляє даний сегмент мережі;
- надійність, розрахована на інтерфейсі цього сегмента мережі;
- розмір MTU.

В якості смуги пропускання сегмента мережі передається мінімальна смуга пропускання мережевим шляхом, через який проходить маршрут до заданої мережі. В якості затримки передається сумарне значення затримки передачі пакета по всіх каналах на маршруті до заданої мережі. Як навантаження передається максимальне значення параметра txload по всіх каналах на маршруті до заданої мережі, який автоматично розраховується на кожному інтерфейсі маршрутизатора. При цьому важливо враховувати, що передається значення навантаження, яке генерується трафіком, що передається у бік префікса-призначення. В якості надійності передається мінімальне значення відношення кількості вірно прийнятих пакетів до загальної кількості прийнятих пакетів по всім каналам на маршруті до заданої мережі. Значення навантаження та надійності обчислюється кожним маршрутизатором окремо на інтерфейсі, підключеному до каналу, який є частиною маршруту до заданої мережі. Процес передачі інформації про префікс представлений на рисунку 1.


**Рис. 1. Передача маршрутної інформації протоколом EIGRP**

На основі отриманої інформації про префікс маршрутизатор обчислює метрику [8, 9, 10]. В стандарті, що описує роботу EIGRP, визначено два варіанти метрики: стандартна та розширена. Стандартна метрика розраховується маршрутизатором за формулою (1):

$$\text{CM} = (K_1 \cdot BW_s + \text{K}_2 \cdot \frac{BW_s}{256 - Lo_{max}} + K_3 \cdot D_s) \cdot \frac{K_5}{K_4 + R_{min}}\ , \qquad (1)$$

де $BW_s = \frac{256 \cdot 10^7}{Bandwidth_{min}}$, $Lo_{max} = Load$, $D_s = 256 \cdot Delay_{summed}$, $R - Reliability$.

Як, видно, з формули (1), при розрахунку метрики для конкретного префіксу використовується параметр (навантаження), максимальне його значення по всім каналам на маршруті до заданого префіксу.

Коефіцієнти $\text{K}_1, \text{K}_2, \text{K}_3, \text{K}_4, \text{K}_5$ представляють собою вагові коефіцієнти, що змінюються від 0 до 255, вони необхідні для того, щоб при розрахунку метрики той чи інший параметр мав більший чи менший вплив на кінцевий результат розрахунку. Верхній поріг – 255 визначений необхідністю масштабування метрики до розміру в 32 біти – максимального розміру значення метрики в таблиці маршрутизації. У всіх поточних реалізаціях EIGRP реальну роль при розрахунку метрики грають затримка та пропуска спроможність.

Крім того, можна використовувати також розширену метрику, яка розраховується за формулою (2).

$$WM = (K_1 \cdot T_{min} + K_2 \cdot \frac{T_{min}}{256 - Lo_{max}} + K_3 \cdot La_{summed} + K_6 \cdot ExtM) \cdot (\frac{K_5}{K_4 - R_{min}}), \qquad (2)$$

де $T_{min} = \frac{65536 \cdot 10^7}{Bandwidth_{min}}$, $La_{summed} = \sum_1^{Hop\ Count} \frac{65536 \cdot Delay_{interface}}{10^6}$.

Використання розширеної метрики обумовлено використанням в даний час каналів з пропускною здатністю 1 Гбіт/с і вище [8, 9, 10]. З формули (1) видно, що при використанні таких каналів відмінностей у метриках не буде, тому при розрахунку розширеної метрики використовуються великі значення чисельників.

При включенні до розрахунку параметра Load, тобто при встановленні значення коефіцієнта > 1, цей параметр враховується тільки при початковому розрахунку метрики для маршруту. Після пер-

шого розрахунку та отримання значення метрики навіть при зміні параметра Load ці зміни не враховуються та перерахунок метрики не провадиться. Ця особливість пов'язана з такими складнощами. Як відомо з положень теорії масового обслуговування, навантаження (трафік), що генерується підключеними до мережі пристроями, має імовірнісну природу, тобто значення навантаження, що є в мережі не завжди і змінюється в часі випадковим чином. Розподіл ймовірностей дії навантаження трафіку при цьому різниться, залежно від типу сервісів, що надаються мережею та типу пристроїв, підключених до мережі. В результаті зміна параметра Load, який відображає зміну навантаження, що діє на інтерфейсі маршрутизатора, має нелінійну природу; зміна також відбувається відповідно до чинного в цій мережі закону розподілу ймовірностей, через що значення параметра може приймати значення, що сильно різняться за абсолютною величиною, на короткому інтервалі часу. В результаті, частий перерахунок метрики з значеннями параметра Load, що сильно розрізняються, може призводити до частої зміни маршруту для проходження трафіку до заданого префіксу, що:

- збільшує затримку при передачі трафіку;
- призводить до втрати деяких пакетів у моменти нестабільності мережі;
- викликає зміни в послідовності пакетів, що передаються.

Ілюстрація цієї ситуації наведено на рисунку 2.

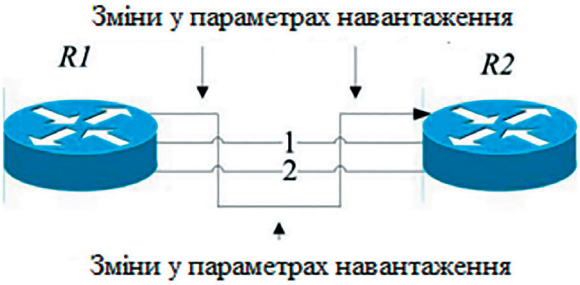


**Рис. 2. Проблема з нестабільністю маршрутів в EIGRP**

Крім того, зміна метрики маршруту викликає роботу механізмів дифузійних обчислень, закладені в EIGRP, через що інформація про часті зміни метрики, і, відповідно, маршруту, передається

до інших маршрутизаторів, які беруть участь у роботі EIGRP, які також перераховують метрику і змінюють рішення про маршрутизацію, що, в результаті, призводить до постійної частої зміни всіх маршрутів та нестабільності мережі [9, 10], в результаті яких негативні фактори, перераховані вище, посилюються і збільшується кількість відмов в обслуговуванні, що позначається на доступності інформації, що передається по мережі.

З усіх зазначених причин, як було зазначено вище, після першого розрахунку та отримання значення метрики навіть при зміні параметра Load ці зміни не враховуються та перерахунок метрики не провадиться.

Розглянемо алгоритм перерахунку навантаження. Поточні реалізації протоколу не використовують параметри Load для розрахунку метрики. Для того, щоб обійти ці обмеження, можна використовувати адаптивний алгоритм реагування на зміни навантаження, який задається різницевим рівнянням (3):

$$Load = \alpha \cdot Load + (1 - \alpha) \cdot Load_{new} \quad 0 \leq \alpha \leq 1 \tag{3}$$

Відповідно до особливостей різницевого рівняння, значення параметра Load, що обчислюється за поточний інтервал на контролері, буде залежати від значень параметра Load, обчисленого на попередньому інтервалі та значень параметра Load, отриманого за поточний інтервал від маршрутизатора. При цьому вага останнього в результаті обчислень поточного такту буде залежати від значення коефіцієнта α [11]. Зі збільшенням даного коефіцієнта зменшується чутливість даного алгоритму до змін у навантаженні, зі зменшенням даного коефіцієнта збільшується чутливість алгоритму до змін у навантаженні. Питання про те, яке значення коефіцієнта вибрати у разі конкретних топології та моделі трафіку (закону розподілу ймовірностей навантаження) є відкритим і вимагає подальшого дослідження.

На рівні контролера задається граничне значення, яке забезпечує умову реакції на зміни в навантаженні, спільно із завданням коефіцієнта α визначається загальна реакція алгоритму на зміни навантаження в мережі. Оскільки при обчисленні метрики для маршруту маршрутизатором використовуються цілі значення параметрів (пропускної спроможності, затримки, навантаження, надійності), то і обчислення на рівні контролера має сенс роби-

ти з цілими числами, тоді результат обчислень слід округлювати до найближчого цілого значення, щоб згодом передавати маршрутизатору. Обчислення здійснюється для кожного інтерфейсу маршрутизатора, параметри якого може отримати контролер.

Крім питання про вибір коефіцієнта, α також постає питання про застосування запропонованого механізму у разі конкретної мережевої топології. Він також вимагає окремого дослідження, але можна сказати про крайні випадки такого питання. Немає сенсу використовувати запропонованої механізм у мережі з топологією, зображеної на рисунку 3.

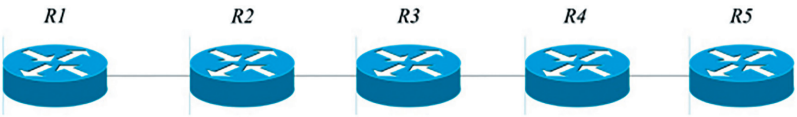


**Рис. 3. Мережева топологія**

Оскільки така топологія є послідовністю маршрутизаторів, з'єднаних між собою послідовно, існує єдиний маршрут до будь-якої з мереж-призначення. Внаслідок цього перерахунок метрики та обчислення алгоритму не мають сенсу і займають процесорний час контролера та маршрутизаторів мережі.

Після того, як внаслідок зміни значення навантаження було досягнуто певного (заданого адміністратором) граничного значення цього параметра, контролер передає на маршрутизатори (один або кілька) рішення про перерахунок маршруту. Оскільки маршрутизатору, який бере участь у роботі EIGRP, відомі поточні значення параметра навантаження, передача цих параметрів не потрібна. Після отримання рішення про перерахунок маршрутів маршрутизатор запускає обчислення метрики для всіх маршрутів, або для тих маршрутів, які дотичні значенням метрики, що змінилося, відповідно до поточних специфікацій EIGRP без будь-яких змін. Така кінцева реалізація запропонованого методу задіяє механізми та алгоритми, що вже є на мережевих пристроях, і не вимагає змін ні в апаратній, ні в програмній реалізації мережевих пристроїв. Запропонований алгоритм разом з усіма обчисленнями розгортається на контролері, у ролі якого може виступати будь-яка платформа, апаратна чи програмна з підтримкою відповідних програмних інтерфейсів. Архітектура запропонованого рішення наведено на рисунку 4.

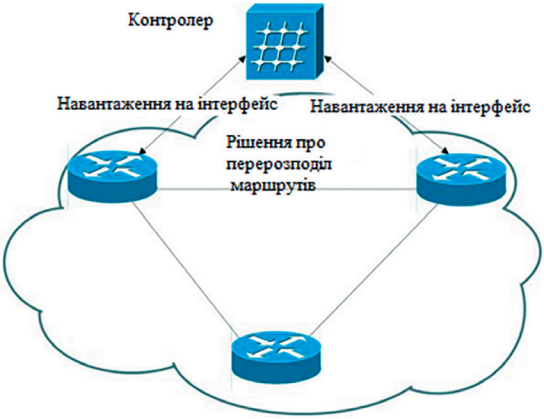


**Рис. 4. Пропоноване рішення щодо обліку навантаження в мережі ПД**

Якщо високе навантаження діє на всіх каналах у мережі, то перерахунок метрики не дозволить перенаправити частину трафіку менш завантаженим каналом, оскільки останніх у мережі немає.

Розглянемо механізм збору інформації з вузла мережі. Розглянутий алгоритм пропонується використовувати в рамках гібридної реалізації програмно-визначуваної мережі.

При реалізації запропонованого методу практично використовувалося:

- Устаткування Cisco Systems Inc.
- Програмні компоненти Cisco Systems Inc., які складаються з операційної системи для мережевого обладнання з встановленими API, а також програмних бібліотек, сумісних із зазначеними API (Cisco IOS 15.4.2, ONE Platform Kit (onePK)).

Cisco® Open Network Environment (ONE) є комплексним рішенням, яке дозволяє впровадити в існуючу мережну інфраструктуру елементи програмно-визначеної мережі, зокрема, елементи гібридної моделі програмно-визначуваної мережі. По суті, це рішення включає набір взаємопов'язаних технологій і механізмів:

- API для різноманітних платформ Cisco Systems Inc;
- контролер;
- додатки агенти для взаємодії з контролером.

Можливості onePK є реалізацією гібридної моделі SDN. Однак, за допомогою цих можливостей також може бути реалізована класична модель SDN, зокрема, за допомогою onePK на мережевих пристроях можуть бути впроваджені OpenFlow-агенти, кожен з яких є частиною архітектури класичної моделі SDN.

Пакет розробки onePK сумісний з усіма основними платформами Cisco. Гнучке середовище розробки, включене до складу onePK, надає програмний рівень представлення мережевих елементів за допомогою API, що підтримуються об'єктно-орієнтованими мовами програмування [12, 13].

Центральним елементом архітектури onePK є єдиний набір бібліотек API для всіх основних платформ Cisco.

По суті, цей рівень інфраструктури (платформи зі встановленими бібліотеками API) надає рівень абстракції для специфічних платформенних рішень. Така реалізація дозволяє розробникам додатків не концентруватися на специфічних особливостях окремих платформ чи всієї інфраструктури, що значно спрощує розробку та збільшує масштабованість додатків. Розробники можуть використовувати ті самі бібліотеки API у всій інфраструктурі, навіть якщо окремі мережеві пристрої працюють під управлінням різних операційних систем. Між рівнем уявлень та рівнем мережевої інфраструктури будується канал для забезпечення їхньої взаємодії, як правило, у рамках клієнт-серверної моделі взаємодії. Вся описана вище архітектура onePK представлена на рисунку 5.

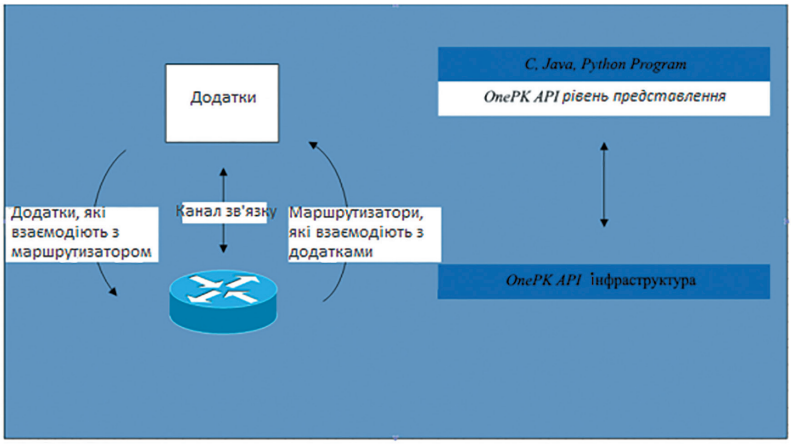


**Рис. 5. Архітектура onePK**

Існує кілька моделей розгортання onePK додатків. Додатки onePK можуть бути розгорнуті:

- на пристроях Cisco (комутаторах, маршрутизаторах);
- на інтегрованих у пристрої Cisco обчислювальних елементів;
- на зовнішніх серверах.

При реалізації механізму було обрано модель розгортання додатків на зовнішньому сервері програми onePK, оскільки запускаються на зовнішніх серверах, пов'язаних з IP з мережевими пристроями. Принцип роботи цієї моделі розгортання проілюстровано на рисунку 6.

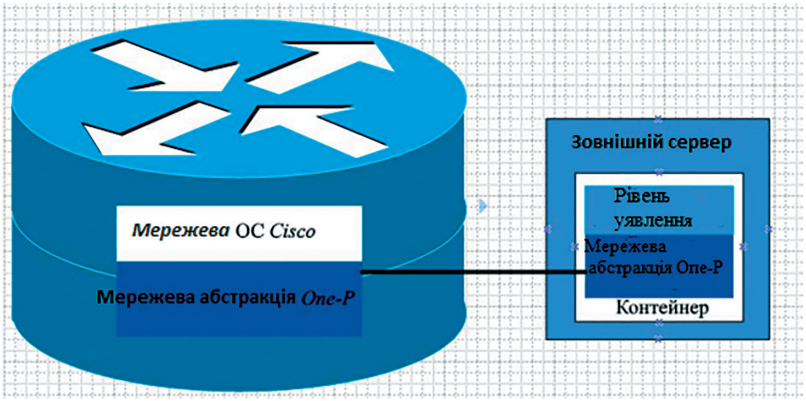


**Рис. 6. Модель розгортання додатків на зовнішньому сервері**

Ця модель дозволяє розробникам та адміністраторам мережевої інфраструктури вибирати платформу для роботи за власними критеріями. Платформи (серверні) можуть бути використані:

- сервери під управлінням операційної системи GNU/Linux;
- сервери під керуванням операційної системи Windows;
- мобільні пристрої під керуванням операційних систем Android або iOS (Apple).

Ці платформи можуть запускати виконання onePK або в окремих програмних контейнерах, або в просторі операційної системи сервера. Такий підхід забезпечує найбільший рівень ізоляції onePK додатків.

Розглянемо реалізацію механізму перерахунку навантаження. Програма, що реалізує запропонований алгоритм обліку навантаження, написана об'єктно-орієнтованою мовою програмування Java. Вибір в якості мови програмування Java обумовлений його багатоплатформністю [12], а також великою кількістю матеріалів і готових бібліотек. При цьому використані спеціальні бібліотеки API для onePK, створені спеціально для мови Java. Програма складається з трьох основних компонентів:

- DeviceSetup модуль, що власне реалізує з'єднання до API onePK пристрою;
- PinningHandler – модуль верифікації TLS сесії;

Recalculation – модуль зчитує параметри вхідного та передаючого навантаження, а так само здійснює її перерахунок за формулою (3) і передає нове значення на пристрій.

**Висновки та пропозиції.** Розглянуте рішення пропонується у рамках концепції програмно-визначуваної мережі. Ця концепція виносить площину управління мережею на новий архітектурний рівень, що дозволяє приймати рішення на основі великої кількості параметрів, що належать до мережі в цілому, що значно збільшує можливості застосування алгоритму, а також підвищує якість та доцільність його рішень. Таким чином, представлене рішення має такі характеристики:

- здатність отримувати дані від мережевих пристроїв про навантаження в мережі та її зміні;
- централізований аналіз та обробка отриманих даних на контролері;
- застосування адаптивних алгоритмів для прийняття рішень, відповідно зі змінами в навантаженні;
- концентрація процесорного навантаження на контролері.

Застосування описаного в статті методу дозволяє, по-перше, стабілізувати роботу EIGRP, і, по-друге, забезпечити більший контроль над IP-мережею передачі даних, що, в сукупності, дозволяє запобігти відмовам у обслуговуванні та забезпечити властивість доступність..

**ЛІТЕРАТУРА**

1. Боброва О., Колесник М. Програмно-визначені мережі: архітектура, технології, засоби управління // *Наукові праці Української інженерно-педагогічної академії. Серія: Технічні науки*. 2019. № 4. С. 7 – 13.

2. Клименко М., Шмарда О. Програмно-визначені мережі: переваги, недоліки, можливості // *Інформаційні технології та комп'ютерна інженерія*. 2017. Вип. 5. С. 5 – 10.

3. Маринченко І., Нікітін А. Управління трафіком в програмно-визначених мережах // *Науковий вісник Миколаївського національного університету імені В.О. Сухомлинського*. 2018. № 3(137). ч. 1. С. 117 – 120.

4. Міньков О., Курінний О., Міньков Ю. Аналіз можливостей мережі Cisco ONE в програмно-визначеній мережі // *Системні технології*. 2018. № 3(96). С. 43 – 50.

5. Шевчук С., Мелешко В. Програмно-визначені мережі: аналіз стану розробки // *Комп'ютерні науки та інформаційні технології*. 2017. Вип. 163. С. 101 – 108.

6. Арлов О. І., Селезньов Д. Є. Основні напрями розвитку програмно-визначених мереж // *Науковий вісник Національного гірничого університету*. 2019. № 4. С. 32 – 37.

7. Твердохліб А.О., Коротін Д.С. Ефективність функціонування комп'ютерних систем при використанні технології блокчейн і баз данних. *Таврійський науковий вісник. Серія: Технічні науки*. 2022 (6).

8. Цвик О.С. Аналіз і особливості програмного забезпечення для контролю трафіку. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2023. (1).

9. Новіченко Є.О. Актуальні засади створення алгоритмів обробки інформації для логістичних центрів. *Таврійський науковий вісни*к. Серія: *Технічні науки*. 2023 (1).

10. Зайцев Є.О. Smart засоби визначення аварійних станів у розподільних електричних мережах міст. *Таврійський науковий вісник. Серія: Технічні науки*. 2022 (5).

11. Karmakar N., Zhou J., Liu H. *A Survey of Traffic Engineering Techniques in Software Defined Networks* // IEEE Communications Surveys & Tutorials. 2019. Vol. 21. Issue 3. P. 2911.

12. Летенко І. Д. Нечітка керування трафіком модель динамічного у програмованих мережах // *Системи управління та інформаційні технології*. 2015. Т. 62. No 4.1. 3. 179 – 184.

13. Красов А. В., Левін М. В. Можливості управління трафіком у рамках концепції SDN // IV Міжнародна науково-технічна та науково-методична конференція «Актуаль-2015. С. 350 – 354.

14. Doyle D., Carroll J. Routing TCP/IP Vol. 1. Cisco Press. 2005. 936p.

15. Doyle D., Carroll J. Routing TCP/IP Vol. 2. Cisco Press. 2001. 976p.

16. White R., Slice D., Retana A. Optimal Routing Design. Cisco press, 2005, 504 p.

17. Pepelnjak I. EIGRP network design solutions. Cisco press. 2000. 384 p.

18. Zinin A. Cisco IP Routing: пакет forwarding and Intra-domain routing. Addison Wesley Professional. 2001. 656 p.

19. Weisfeld M. Object-Oriented thought process. Addison Wesley, 2009. 309 p.

20. Scratch S. R. Object-Oriented and classical software engineering. McGraw Hill. 2007. 654 p.

**REFERENCES**

1. Bobrova O., Kolesnyk M. Software-defined networks: architecture, technologies, management tools // *Scientific works of the Ukrainian Engineering and Pedagogical Academy. Series: Technical sciences.* 2019. No 4. P. 7 – 13 [in Ukrainian].

2. Klymenko M., Shmarda O. Software-defined networks: advantages, disadvantages, opportunities // *Information technologies and computer engineering.* 2017. Issue 5. Pp. 5 – 10 [in Ukrainian].

3. Marinchenko I., Nikitin A. Traffic management in software-defined networks // *Scientific Bulletin of V.O. Mykolaiv National University. Sukhomlynskyi.* 2018. No. 3(137), part 1. P. 117 – 120 [in Ukrainian].

4. Minkov O., Kurinnyi O., Minkova Yu. Analysis of capabilities of the Cisco ONE network in a software-defined network // *System technologies.* 2018. No. 3(96). P. 43 – 50 [in Ukrainian].

5. Shevchuk S., Meleshko V. Software-defined networks: analysis of the state of development // *Computer sciences and information technologies.* – 2017. Issue 163. Pp. 101 – 108 [in Ukrainian].

6. Arlov O.I., Seleznyov D.E. Main directions of development of software-defined networks // *Scientific Bulletin of the National Mining University.* 2019. No. 4. P. 32 – 37 [in English].

7. Tverdokhlib A.O., Korotin D.S. Efektyvnist funktsionuvannia kompiuternykh system pry vykorystanni tekhnolohii blokchein i baz dannykh. T*avriiskyi naukovyi visnyk. Seriia: Tekhnichni nauky*. 2022. (6) [in Ukrainian].

8. Tsvyk O.S. Analiz i osoblyvosti prohramnoho zabezpechennia dlia kontroliu trafiku. *Visnyk Khmelnytskoho natsionalnoho universytetu. Ceriia: Tekhnichni nauky*. 2023. (1) [in Ukrainian].

9. Novichenko Ye.O. Aktualni zasady stvorennia alhorytmiv obrobky informatsii dlia lohistychnykh tsentriv. T*avriiskyi naukovyi visnyk. Seriia: Tekhnichni nauky*. 2023 (1) [in Ukrainian].

10. Zaitsev Ye.O. Smart zasoby vyznachennia avariinykh staniv u rozpodilnykh elektrychnykh merezhakh mist. *Tavriiskyi naukovyi visnyk. Seriia: Tekhnichni nauky*. 2022. (5) [in Ukrainian].

11. Karmakar N., Zhou J., Liu H. *A Survey of Traffic Engineering Techniques in Software Defined Networks* // IEEE Communications Surveys & Tutorials. 2019. Vol. 21, Issue 3. P. 2911 [in English].

12. Letenko I. D. Fuzzy traffic management model of dynamic in programmable networks // *Management systems and information technologies*. 2015. T. 62. No. 4.1. Q. 179 – 184 [in Ukrainian].

13. Krasov A.V., Levin M.V. Possibilities of traffic management within the SDN concept // IV International scientific-technical and scientific-methodical conference Aktual-2015. C. 350-354 [in Ukrainian].

14. Doyle D., Carroll J. Routing TCP/IP Vol. 1. Cisco Press. 2005. Pp. 936 [in English].

15. Doyle D., Carroll J. Routing TCP/IP Vol. 2. Cisco Press. 2001. Pp. 976 [in English].

16. White R., Slice D., Retana A. Optimal Routing Design. Cisco press. 2005. Pp. 504 [in English].

17. Pepelnjak I. EIGRP network design solutions. Cisco press. 2000. Pp. 384 [in English].

18. Zinin A. Cisco IP Routing: forwarding and Intra-domain routing. Addison Wesley Professional. 2001. Pp. 656 [in English].

19. Weisfeld M. Object-Oriented thought process. Addison Wesley. 2009. Pp. 309 [in English].

20. Scratch S. R. Object-Oriented and classical software engineering. McGraw Hill. 2007. Pp. 654 [in English].

**СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 03.12.2023**

*УДК 004.02, 004.08, 004.09*
*DOI: https://doi.org/10.53920/ITS-2023-2-8*

**Ольга Іванівна ТКАЧЕНКО,**
кандидат фізико-математичних наук, доцент,
доцент кафедрb інформаційних технологій,
Державний університет інфраструктури та технологій
ORCID ID: 0000-0003-1800-618X
**Олексій Михайлович ТИШУРА,**
магістрант кафедри інформаційних технологій,
Державний університет інфраструктури та технологій
ORCID ID: 0009-0003-6884-6846

# ДЕЯКІ АСПЕКТИ РОЗРОБКИ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ COFFEE++

***В наш час веб-орієнтовані системи використовуються в багатьох сферах економіки, освіти, в нашому повсякденному житті. Особливе місце займають веб-орієнтовані системи закладів громадського харчування (ресторанного бізнесу), серед задач яких, зокрема, є задачі, пов'язані з обробкою замовлень клієнтів та оптимізації бізнес-процесів цих закладів. Сучасна e-commerce перетворилася на повноцінний самостійний бізнес, який пропонує багато можливостей для різних категорій користувачів (покупців, клієнтів), відрізняючись асортиментом товарів, функціональністю, дизайном відповідних вебсайтів. Веб-орієнтовані системи стають необхідним компонентом сучасних закладів громадського харчування. Підприємства, які впроваджують інноваційні технології, мають значні перевагу на ринку. Тому актуальність розробки веб-орієнтованої системи COFFEE++ не викликає сумнівів.***

***Метою роботи є аналіз та дослідження проблем щодо розробки програмного забезпечення відповідної веб-орієнтованої системи та її використання для оптимізації бізнес-процесів закладу громадського харчування (кав'ярні) та підвищення рівня задоволенності клієнтів, які роблять online-замовлення страв і напоїв, які пропонує заклад, та отримують доставку цих замовлень. Мета і завдання статті спрямовані на просування інноваційних підходів та технологій, які можуть підтримати стійкий розвиток закладу громадського харчування (кав'ярні), забезпечити високий рівень обслуговування клієнтів та отримання максимального прибутку.***

***Веб-орієнтована система COFFEE++ є інструментом для кав'ярень, які прагнуть оптимізувати роботу, підвищити рівень задоволеності клієнтів і збільшити рентабельність. Впроваджуючи цю систему, кав'ярні можуть випереджати конкурентів, адаптуватися до споживчих уподобань, надавати своїм клієнтам високий рівень обслуговування. Здатність системи генерувати звіти та аналітику дає можливість менеджерам кав'ярні приймати рішення на основі різноманітних даних (щодо клієнтів, замовлень, ситуації на ринку, тощо), визначати сфери вдосконалення та оптимізувати загальну ефективність бізнесу в закладі.***

***Ключові слова:*** *e-commerce, кав'ярня, вебсайт, вебсторінка, веб-система, вебдизайн, веб-орієнтована система.*

**Olha TKACHENKO**
PhD of physical and mathematical sciences, associate professor,
associate professor at the department
of information technologies,
State University of Infrastructure and Technology
**Oleksii TYSHURA**
undergraduate at the department of information technologies
State University of Infrastructure and Technology

## SOME ASPECTS OF DEVELOPMENT OF WEB-ORIENTED SYSTEM COFFEE++

***Nowadays, web-oriented systems are used in many areas of the economy, education, and in our everyday life. A special place is occupied by web-oriented systems of catering establishments (restaurant business), among the tasks of which, in particular, there are tasks related to the processing of customer orders and the optimization of business processes of these establishments. Modern e-commerce has turned into a full-fledged independent business that offers many opportunities for different categories of users (buyers, clients), differing in the range of products, functionality, and design of the respective websites. Web-oriented systems are becoming a necessary component of modern catering establishments. Enterprises that implement innovative technologies have significant advantages in the market. Therefore, the relevance of the development of the web-oriented system COFFEE++ is beyond doubt.***

***The purpose of the work is the analysis and research of problems related to the development of software of a suitable web-oriented system and its use to optimize the business processes of a public catering establishment (cafeteria) and increase the level of satisfaction of customers who make online orders for food and drinks offered by the establishment , and receive delivery of these orders. The purpose and objectives of the article are aimed at promoting innovative approaches and technologies that can support the sustainable development of a public catering establishment (coffee shop), ensure a high level of customer service and obtain maximum profit.***

***The web-oriented system COFFEE++ is a tool for coffee shops that seek to optimize work, increase the level of customer satisfaction and increase profitability. By implementing this system, coffee shops can stay ahead of competitors, adapt to consumer preferences, and provide their customers with a high level of service. The system's ability to generate reports and analytics enables coffee shop managers to make decisions based on a variety of data (regarding customers, orders, market situation, etc.), identify areas for improvement, and optimize overall business performance in the establishment.***

***Keywords:*** *e-commerce, coffee shop, website, web page, web system, web design, web-oriented system.*

**Постановка проблеми.** В наш час e-commerce (електронна комерція) [15] стає все більш звичайною та розвиненою системою взаємодії різних категорій користувачів з підприємствами, які надають послуги з продажу товарів в режимі online що обумовлено, зокрема, такими причинами, як:

- розвиток технологій підтримки вебсервісів та веб-орієнтованих систем [3, 8, 16];
- розширення функціональності вебсервісів та веб-орієнтованих систем електронної комерції;
- перехід багатьох потенційних покупців в online-режим відвідування магазинів, закладів харчування, аптек через, зокрема, Covid-обмеження (карантин) та воєнний стан в країні.

Зараз e-commerce перетворилася на повноцінний самостійний бізнес. Розвиток глобальних маркетплейсів [19, 20] і розширення доступу до них сприяє тому, що користувачі можуть без зайвих зусиль, економлячи час на походи по магазинах, робити покупки в будь-який час і в будь-якій точці земної кулі.

Сучасна e-commerce пропонує багато можливостей для різних категорій користувачів (покупців), відрізняючись асортиментом товарів, функціональністю, дизайном відповідних вебсайтів.

Веб-орієнтовані системи стають необхідним компонентом сучасних закладів громадського харчування. Підприємства, які впроваджують інноваційні технології, мають значні перевагу на ринку. Тому актуальність розробки веб-орієнтованої системи «Кав'ярня» не викликає сумнівів.

**Аналіз останніх досліджень і публікацій.** Ресторанний бізнес – сфера, яка найбільше постраждала під час карантину у 2020 році та бойових дій і воєнного стану. Багато ресторанів, кафе, кав'ярень перейшли на використання популярних кур'єрських служб для доставки їжі та напоїв. В наш час ресторанний бізнес зустрівся з різними складнощами, обумовленими ситуаціями, пов'язаними, наприклад, із комендантською годиною, дефіцитом та високою ціною на пальне [1].

*Веб-орієнтована система* – спеціалізований вебсайт, яким володіють виробники, продавці та інші компанії, призначені для просування товарів, збільшення продажів і залучення нових клієнтів. Однією з найбільш поширених моделей електронної торгівлі сфери B2C є веб-орієнтована система обробки замовлень.

Створення веб-орієнтованої систем передбачає використання, зокрема, таких інструментальних засобів, як:

*Мов розмітки:* HTML, XHTML, XML, CSS [11]. При створенні вебсторінок використання цих мов є необхідним. HTML відповідає за зміст та дизайн сторінки, але сучасна стратегія розробки вебсайтів та відповідних систем спрямована на використання (X)HTML і XML для передачі семантики веб-сайту, а CSS – для дизайну.

CSS – спеціалізований засіб опису сторінок, написаних мовами розмітки даних. CSS застосовується до XML-документів, але найчастіше – для візуального представлення HTML- та XHTML-сторінок. Існують різні рівні та профілі CSS (CSS1, CSS2 і CSS3) [7, 18]. Основним в CSS є розділення змісту сторінки та її візуального представлення [10].

*Вбудовування растрової графіки.* Сучасні браузери приймають зображення у форматах WEBP, JPG, GIF і PNG [6]. Верстка і дизайн більшості сторінок базується на поєднанні (X) розмітки HTML і CSS з графікою [17].

*Програми підтримки мов на боці сервера* [15, 16].

**Мета статті** полягає в аналізі та дослідженні проблем розробки програмного забезпечення веб-орієнтованої системи та її використання для оптимізації бізнес-процесів закладу громадського харчування – кав'ярні – та підвищення рівня задоволеності клієнтів, які роблять online-замовлення страв і напоїв, які пропонує заклад, та отримують доставку цих замовлень.

Досягнення цієї мети забезпечується вирішенням, зокрема, наступних завдань:

- визначення потреб закладів громадського харчування у відповідних веб-орієнтованих системах чи вебсервісах;
- визначення класів бізнес-процесів кав'ярні, здійснення яких може бути оптимізовано за допомогою відповідної веб-орієнтованої системи;
- визначення класів задач, які можуть бути вирішені більш ефективно за допомогою відповідної веб-орієнтованої системи;
- дослідження інструментарію, програмного забезпечення та технологій, адекватних потребам та можливостям сектора громадського харчування, що дозволяють збирати, аналізувати та використовувати різноманітні дані (дані про меню, цінову політику на ринку, логістику, спілкування із потенційними клієнтами, рекламні акції, маркетингову інформацію, тощо) для покращення ефективності роботи кав'ярні.

Мета і завдання статті спрямовані на просування інноваційних підходів та технологій, які можуть підтримати стійкий розвиток закладу громадського харчування (кав'ярні), забезпечити високий рівень обслуговування клієнтів та отримання максимального прибутку.

**Виклад основного матеріалу дослідження.** Класифікація веб-орієнтованих систем обробки замовлень відбувається, зокрема, за такими ознаками:

- За способом продажу товару в мережі:
  - Internet-магазини.
  - вебвітрини, торгові системи.
  - Торгові ряди.
  - Контентні проєкти.
- За бізнес-моделлю:
  - Повністю online-магазин.
  - Суміщення offline-бізнесу з online.

- За принципом взаємовідносин з постачальниками:
  - Магазини, які володіють власним складом (наявні реальні товарні запаси).
  - Магазини, що працюють за договорами з постачальниками (відсутні значні товарні запаси).
- За ступенем автоматизації серед торгових
  - Системи електронних магазинів.
  - Вебвітрини.
  - Власне Internet-магазини.
  - TIS – торгові Internet-системи [2].

Прикладами веб-орієнтованих систем, можуть бути:

- серверні модулі для запуску скриптів за певним розкладом (cron);
- програма для обробки графічних зображень, яка створює мініатюрні копії зображень за запитом скрипта.

Вибір системи управління контентом (CMS) [4] для системи електронної комерції є важливим, бо кожна CMS має свої переваги та недоліки.

Серед типів CMS для систем електронної комерції можна виділити, зокрема:

- безкоштовні CMS для інтернет-магазину (з відкритим вихідним кодом);
- комерційні CMS;
- студійні CMS – ексклюзивні CMS, розроблені багатьма вебстудіями;
- так звані «самописані» (несерійні).

У CMS основною інформаційною одиницею в базі даних є сторінка (елемент контенту). Зазвичай в базі даних CMS є спеціальна таблиця, що відображає елемент контенту і характеризує її певним набором полів. Заголовок є головним полем сторінки. Ієрархічні та структурні зв'язки між елементами контенту можуть бути реалізовані по-різному, залежно від системи. URL-адреса – унікальний ідентифікатор веб-орієнтованої системи обробки замовлень з точки зору користувача.

Будь-який сайт, Інтернет-магазин чи інший online-ресурс складається з певних елементів, одним із яких є вебсторінка. Серед вебсайтів можна виділити: односторінкові сайти (landing, лендінг) [21], online-ресурси, що складаються з тисяч сторінок, згрупованих за категоріями, розділами та підрозділами (залежно від виду та призначення Інтернет-майданчика).

Вебсторінка є однією зі складових частин вебсайту, Інтернет-магазину, порталу чи блогу. Доступ до вебсторінки здійснюється через один з браузерів, який використовується для виходу в мережу через Інтернет. У вебсторінці має бути відображено такі складові елементи, як текст; картинки; аудіо- або відеоконтен. Сторінки можуть бути представлені у різному форматі online-ресурсів.

Веб-орієнтовані системи використовують вебдодатки – допоміжні програмні засоби, призначені для автоматизованого виконання дій як на боці сервера, так і на боці користувача.

Створення веб-орієнтованих систем передбачає прходження, зокрема, таких етапів, як: визначення вимог, проєктування, реалізації, тестування. Але програмне забезпечення системи працює не на комп'ютері користувача, а на віддаленому сервері мережі. З одного боку це зручно, а з іншого обумовлює цілу низку вимог до програмного забезпечення системи, яка розробляється.

*Ефективним контентом* сторінки є інформація, для отримання якої користувач здійснює перехід на цю сторінку та за допомогою якої була додана на вебсайт ця сторінка. Ефективний контент сторінки виконує функцію логічного зв'язку в загальній інформаційній структурі вебсторінки та може містити велику кількість додаткової інформації (дизайн, навігація, реклама та ін.).

Структурна схема вебсторінки відображена на рис. 1.

Важливим завданням створення веб-орієнтованих систем обробки замовлень є моделювання та представлення відповідної інформації у базі даних вебсистеми [4]. До ефективного контенту, зокрема, входить: заголовок та основний контент, анотація, зображення – ілюстрація анотації чи основного контенту, короткий заголовок, якщо основний заголовок довгий (наприклад для меню), дата створення, дата редагування.

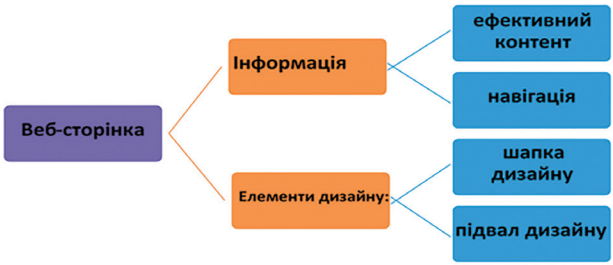


**Рис. 1. Структурна схема вебсторінки**

Джерело: [5]

CMS містить, зокрема, такі елементи як:

- *Адміністративна частина* – модуль управління контентом, який виконує функції з редагування вебсайту.
- *Структурований контент*, який містить базу даних для зберігання тексту вебсторінок у форматі HTML та набір впорядкованих медіафайлів.
- *Ядро системи* («модуль подання») забезпечує керування логіки подання контенту користувачам та навігація по контенту.

В наш час поширеними стали вебсистеми для зберігання, оцінки, аналізу та подальшого застосування даних. Ці системи можна використовувати для особистих та професійних цілей, відкриваючи багато можливостей для бізнесу.

Перш ніж почати працювати з будь-яким сайтом, необхідно чітко продумати його структуру. Це безпосередньо впливає на ранжування ресурсу в пошукових системах, а також на його сприйняття користувачами.

Структура сайту – це схема розташування його сторінок, категорій, підкатегорій і товарів. Це своєрідний план, в якому прослідковується логічний зв'язок між сторінками. З технічної точки зору навігація ресурсу являє собою набір URL, що розташовані в певній послідовності. Вона нерозривно пов'язана з семантичним ядром. А саме воно визначає, які папки та документи мають бути на сайті.

Важливо також розрізняти, що структура сайту є зовнішня і внутрішня. Під першою розуміють макет сторінки із зазначенням розташування на ній блоків. Друга відображає категорії, належність до них певних сторінок і матеріалів.

Веб-орієнтовані системи в e-commerce передбачають значно нижчі витрати на забезпечення та організацію роботи через відсутність необхідності великої матеріально-технічної бази (будівель, споруд, приміщень та обслуговуючого персоналу). Але слід вказати й на недоліки таких систем, серед яких основними, зокрема, є:

- невизначеність щодо фактичної наявності товару;
- відповідності товару основним параметрам якості;
- шахрайство при проведенні фінансових операцій (при переказі коштів за придбаний товар чи послугу);
- проблеми з логістикою та доставкою.

Система автоматизації бізнес-процесів сучасних закладів харчування, зокрема, кав'ярень – незамінний помічник в організації та оптимізації робочих процесів. Завдяки широкому функціоналу, програмне забезпечення такої системи допомагає керівникам підприємств вести бізнес на кожному етапі більш ефективно.

Обслуговування клієнтів у віртуальній кав'ярні за допомогою відповідної веб-орієнтованої системи здійснюється таким чином [5]:

1. *Відкриття вебсайту* з переліком наявних товарів та необхідними елементами інтерфейсу для здійснення вибору та оформлення покупки товарів. Для пошуку товарів у веб-орієнтованій системі можна користуватися каталогом або внутрішньою пошуковою системою. Товар можна оглянути (зазвичай за допомогою фотографій), ознайомитися з його ціною, споживчими та технічними характеристиками (за допомогою тексту і спеціальних символів).
2. *Перегляд переліку товарів та їх вибір* (занесення клієнтом товарів у свій віртуальний товарний кошик). Зробивши остаточний вибір товарів, клієнт підтверджує замовлення.
3. *Реєстрація покупця* вимагає заповнення спеціальної форми, яка містить інформацію про покупця, поштову та/чи електронну адресу, особистий пароль та ряд додаткових відомостей. Під час реєстрації особиста інформація покупця захищається спеціальними методами безпеки. Такими засобами можуть бути протоколи SET або SSL [2].
4. *Вибір способу доставки та оплати товару.*
5. *Підтвердження інформації про замовлення.*
6. *Оплата замовленого товару* одним з поширених способів:
   - Оплата готівкою (при доставці товару кур'єром за вказаною адресою).
   - Оплата в системі онлайн-платежів.
   - Оплата поштовим чи телеграфним переказом.
   - Накладений платіж (для покупців, які віддають перевагу отриманню товару поштою).
   - Оплата за безготівковим розрахунком.
7. *Доставка покупки клієнту* одним з наведених нижче способів:
   - Доставка кур'єрською службою.

- Міжнародна доставка.
- Доставка поштою, коли придбаний товар передається клієнтові у поштовому відділенні.
- Доставка магістральними (повітряними, залізничними, автомобільними, водними) шляхами. Зазвичай він використовується для доставки великих або громіздких товарів. Узгоджені умови угоди застосовуються до цін і термінів доставки.

Як правило, розробники і дизайнери співпрацюють при створенні вебдодатків. Файл XAML [4] складається з конструкцій, які детально описують кожен елемент, присутній на даній сторінці, з кожним описом з набором властивостей.

Розробка веб-орієнтованих систем передбачає використання технологій front-end [5] та back-end [7]. Front-end технологія підтримується HTML5, CSS3 та JavaScript.

Структура веб-орієнтованої системи обробки додатку (рис.2) містить, зокрема [6]:

- *Ядро системи* забезпечує взаємодію всіх інших складових як єдиної системи.
- *Модуль бази даних* забезпечує управління ресурсами бази даних.
- *Модуль реєстрації користувачів* забезпечує роботу з персональними даними користувача (клієнта кав'ярні).
- *Модуль роботи із замовником* забезпечує роботу з формування замовлень (на основі даних, отриманих від замовника у відповідних формах) та подальші транзакції.
- *Модуль звітів* відповідає за автоматизацію звітного процесу.
- *Користувацький модуль* надає користувачеві доступну та комфортну форму інтерфейсу, що базується на використанні технологій підтримки front-end.
- *Модуль моніторингу процесів* забезпечує, зокрема, відслідковування статусів поставлених адміністратором завдань та завантаженості системи.

8. *Модуль безпеки* відповідає за процеси аутентифікації та авторизації, а також за безпеку системи в цілому.

9. *Адміністративний модуль* забезпечує роботу адміністратора щодо управління системи в цілому та окремих її частин.


**Рис. 2. Модель веб-орієнтованої системи**

Джерело: [6]

*Підсистема управління* – вебсервіс зі зручним графічним користувацьким інтерфейсом, який дає, зокрема, такі можливості:

- аутентифікація в системі та переадресації до підключених до неї підсистем;
- адміністрування у вебсистемі інших підключених підсистем;
- налаштування на кожного об'єкта системи.
- обмін інформацією між клієнтом та базою даних.

Для створення єдиного інформаційного простору веб-орієнтованих систем обробки замовлень використовується PDM-система (зовнішня система документообігу), яка виконує роль підсистеми спеціального призначення (рис. 3) [3, 8].

*Підсистеми* – вебдодатки, призначені для вирішення технологічних завдань та інтегровані в більшу систему. Запуск підсистем залежить від прав користувача, запиту вебдодатку на доступ та необхідного доступу вебдодатку до відповідних баз даних (знань).

*Агенти* — підсистеми в межах веб-орієнтованої системи , які взаємодіють між собою. Для розширення можливостей агентів у адміністратора системи можна запросити абстрактний клас Агент (для обміну інформацією та взаємодії через вебсервіс MultiAgentSystem [8]).


**Рис. 3. Ієрархічна модель веб-орієнтованої системи**
Джерело: [3]

Розглянемо сучасні вітчизняні вебсистеми кав'ярень.

Пекарня-кав'ярня Bakery – кав'ярня (м. Черкаси та м. Сміла), яка реалізує широкий асортимент продукції власного виробництва [10].

Пекарня-кав'ярня надає пропозиції з реалізації наступних категорій продуктів:

- Напої.
- Випічка.
- Сніданки.
- Піца.
- Десерти.
- Морозиво.

Кав'ярня здійснює доставку замовлень по м. Черкаси та м. Сміла. Оплата приймається за допомогою таких платіжних сервісів, як: applePay, googlePay, payPal [9]. Веб-орієнтована система пекарні-кав'ярні Bakery (рис. 4) має інтуїтивний UI/UX [11], сприяючи легкому процесу реєстрації для роботи із замовленням, трекінгом та відгуком. Інтерфейс зрозумілий і, якщо користувач відвідує сайт цієї системи вперше, то не зустріне труднощів при роботі з ним.

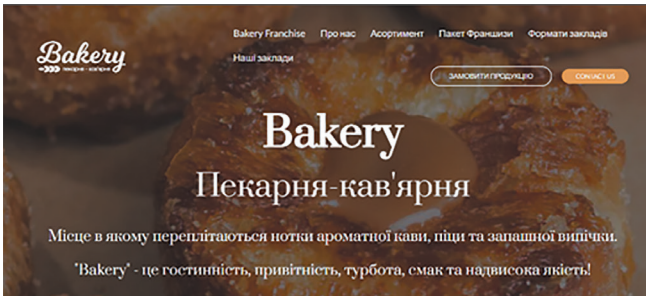


**Рис. 4. Вебсайт пекарні-кав'ярні Bakery**

Джерело: [12]

Кав'ярня The blue cup coffee shop [13] розташована у м. Києві і має широкий асортимент продукції, зокрема:

- Сніданки.
- Сендвічі.
- Салати.
- Основні страви.
- Супи.
- Напої (кава, холодні напої, раф кава, какао, чай, пиво, вино).
- Десерти.

Цей заклад не доставляє свою продукцію, можливо замовити «із собою» або «на виніс». Оплата здійснюється за допомогою: applePay, googlePay, payPal. На рис. 5 представлено вебсайт кав'ярні The blue cup coffee shop.

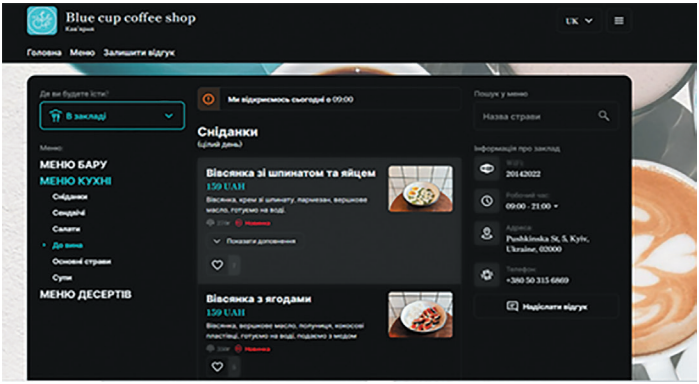


**Рис. 5. Вебсайт кав'ярні The blue cup coffee shop**

Джерело: [13]

AROMA KAVA – мережа кав'ярень на території України [14]. Цей заклад в наш час займає передову позицію на ринку. На сайті закладу пропонуються наступні позиції (рис. 6):

- Класична кава.
- Авторські кавові напої.
- Десерти.

AROMA KAVA доставку замовлень не здійснює. Відрізняється від більшості своїм дизайном та стилізованою рекламою.

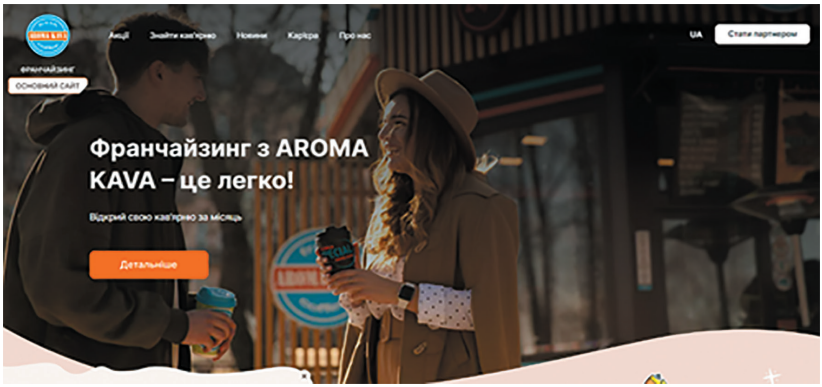


**Рис. 6. Вебсайт кав'ярні Aroma Kava**

Джерело: [14]

Авторська веб-орієнтована система COFFEE++ була розроблена для вирішення проблем, які постають перед власниками та менеджерами кав'ярень, з метою підвищення результативності, оптимізації процесів і покращення результативності ресторанного бізнесу.

Авторська веб-орієнтована система COFFEE++ (головне вікно вибору напоїв в якій представлено на рис. 7) пропонує можливість управління запасами, автоматизуючи процеси відстеження, замовлення та поповнення запасів, допомагає мінімізувати розбіжності в запасах, запобігти дефіциту та зменшити втрати.

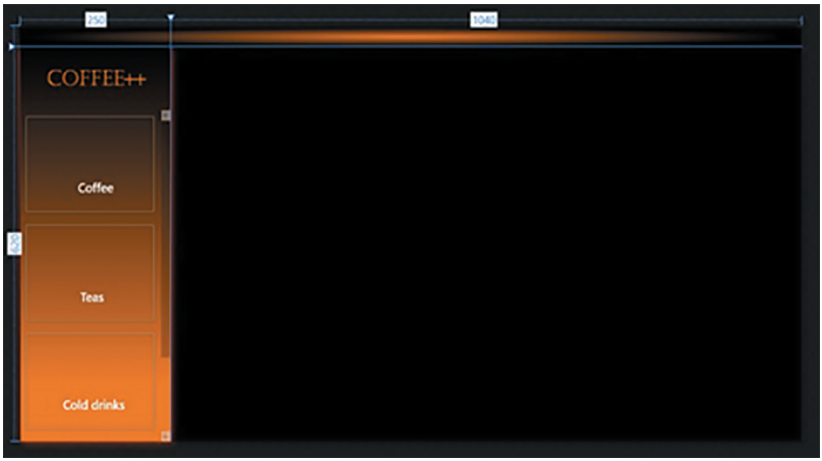


**Рис. 7. Головне вікно вибору напою в системі COFFEE++**
Джерело: авторська система

Авторська веб-орієнтована система COFFEE++ [22] передбачає вирішення, зокрема, наступних проблем:

- *Автоматизація обміну даними між модулями системи.* Для вирішення цієї проблеми в системі було використано сучасні методи проєктування баз даних та відповідне програмне забезпечення, що обумовило зменшення навантаження на персонал кав'ярні (наприклад, адміністрацію), зберігаючи при цьому можливість прямого управління обміном даними.
- *Зручна обробка записів (додавання, видалення, перегляд, редагування).* У системі було розроблено зручний та комфортний інтерфейс, який дозволяє в інтуїтивно зрозумілій формі виконувати всі необхідні операції над відповідними даними користувачів, сформованими замовленнями та переліками товарів і послуг.
- *Імпорт даних про товари та послуги з файлу CSV (отримуючи зображення з вебресурсу та візуалізація його у відповідному форматі).* Для цього в системі було розроблено спеціальні допоміжні сервіси.

Однією з переваг системи COFFEE++ є її спроможність спрощувати та оптимізувати управління замовленнями. Використовуючи веб-орієнтовану систему, працівники кав'ярні можуть швидко

обробляти отримані замовлення, контролювати їхній статус та забезпечувати швидку і точну доставку.

Інтуїтивно зрозумілий користувацький інтерфейс системи та оновлення в режимі реального часу дозволяють координувати роботу різних відділів, усуваючи потенційні "вузькі місця" та покращуючи якість обслуговування.

**Висновки та пропозиції.** Веб-орієнтована система COFFEE++ є цінним інструментом для власників та керівників кав'ярень, які прагнуть оптимізувати роботу, підвищити рівень задоволеності клієнтів та збільшити рентабельність закладу.

Впроваджуючи цю систему та використовуючи її переваги, кав'ярні можуть випереджати конкурентів, адаптуватися до мінливих споживчих уподобань і надавати своїм клієнтам винятковий досвід.

Наявність веб-орієнтованої системи значно зменшує початкові витрати та поточні витрати на обслуговування, що робить систему доступним і масштабованим рішенням для бізнесу будь-якого розміру.

Крім того, здатність системи генерувати детальні звіти та аналітику дає можливість власникам та менеджерам приймати рішення на основі даних, визначати сфери для вдосконалення та оптимізувати загальну ефективність бізнесу.

Веб-орієнтована система COFFEE++ суттєво впливає на бізнес-процеси таких закладів громадського харчування, як кав'ярні, сприяючи ефективності, зручності та задоволенню клієнтів, що зрештою призводить до утримання клієнтів в цих закладах, і, решта-решт, збільшення доходу закладу.



**ЛІТЕРАТУРА**

1. Тріль М. Як ресторанний бізнес пережив 2022 рік: дослідження. AIN.Business. URL: https://ain.business/2022/12/30/yak-restorannyj-biznes-perezhyv-2022-rik-doslidzhennya/ (дата звернення: 11.10.2023).

2. Лавріщева К. Програмна інженерія: Підручник. Київ: Національна академія наук України, 2008. 319 с.

3. Титенко С. Web-орієнтовані інформаційні системи. Київ: НТУ «Київський Політехнічний Інститут», 2015. 51 с.

4. Стадник Ю. Технології створення програмних та інтелектуальних систем. Львів: Львівський національний університет імені Івана Франка, 2021. 46 с.

5. Структура сайту: основні види та правила їх розробки. URL: https://webtune.com.ua/statti/web-rozrobka/struktura-sajtu/ (дата звернення: 16.10.2023).

6. Gilberto D., Jesslyn A., Afrianto Y. Analysis and Design of Web-based Information System for Coffeeshop Management using Design Thinking Methodology: Case of Kopi KurangLebih. *Journal of information systems and informatics*, 2023. Vol. 5. № 1. P. 217-231. DOI: https://doi.org/10.51519/journalisi.v5i1.455.

7. Fruhlinger J. What is JavaScript? The full stack programming language. InfoWorld. URL: https://www.infoworld.com/article/3441178/what-is-javascript-the-full-stack-programming-language.html (дата звернення: 14.10.2023).

8. Yao J. Web-based support systems. Springer, 2010. 464 p.

9. Khan B. H. Web-Based training. Educational Technology Pubns, 2003. 599 p.

10. Yulianto H. D., Fauzi R. Design of Web-based Online Sales Information System. IOP Conf. Series: Materials Science and Engineering, 879 (2020) 012007. IOP Publishing. DOI: http://dx.doi.org/10.1088/1757-899X/879/1/012007.

11. Ali A.I. Kosba E., El-sonbaty Ya. A web-based system to enhance lecture interaction: Case Study at Sultan Qaboos University (SQU). *International Journal of Computer Applications*, 2016. 135(9). P. 36-43. DOI: http://dx.doi.org/10.5120/ijca2016908508.

12. Bakery. URL: https://bakery-ck.com.ua/ (дата звернення: 18.10.2023).

13. Blue cup coffee shop. URL: https://bluecupcoffeeshop.choiceqr.com/menu/section:menyu-kuhni/do-vina (дата звернення: 15.10.2023).

14. Aroma kava – мережа кав'ярень по всій Україні. URL: https://aromakava.ua/ (дата звернення: 17.10.2023).

15. Що таке e-commerce. URL: https://hub.kyivstar.ua/news/shho-take-e-commerce/ (дата звернення: 18.10.2023).

16. Regant F., Jansen W. Developing Web-Based Point of Sales Application Encryption on DBMS for Culinary Industry. *Journal of information systems and informatics,* 2023. Vol. 5. №. 3. P. 1020 – 1032. DOI: http://dx.doi.org/10.51519/journalisi.v5i3.544.

17. Singgalen Ye.A., Sutresno S.A. Digital Innovation Design of Tourism Destination Marketing Website Using Design Thinking Method. *Journal of information systems and informatics*, 2023. Vol. 5, No. 2. P. 428 – 444. Vol. 5. №. 3. DOI: http://dx.doi.org/10.51519/journalisi.v5i2.464.

18. About CSS. URL: https://dbpedia.org/page/CSS (дата звернення: 11.10.2023).

19. Що таке маркетплейс та чим він відрізняється від звичайного інтернет магазина. URL: https://kovel.tv/commerce/52-csho_take_marketplejs_ta_chim_vin_vidriznyaetsya_vid_zvichajnogo_internet_magazina.html. (дата звернення: 17.10.2023).

20. What Is A Marketplace? URL: https://www.shopery.com/insights/what-is-a-marketplace. (дата звернення: 19.10.2023).

21. Shwake E. What is a landing page? Here's everything you need to know. URL: https://www.wix.com/blog/what-is-a-landing-page (дата звернення: 17.10.2023).

22. Ткаченко О.А., Тишура О.М. Система «Кав'ярня» – програмне забезпечення керування касою в сфері послуг. Матеріали IV Міжнар. наук.-практ. конф. «Інформаційні технології та цифрова економіка» (Київ, 04–05 травня 2023 р.). С. 150 – 152.

**REFERENCES**

1. Tril M. How the restaurant business survived the year 2022: research.. AIN.Business, available at: https://ain.business/2022/12/30/yak-restorannyj-biznes-perezhyv-2022-rik-doslidzhennya/ (Accessed 11 October 2023).

2. Lavrishcheva K. Software engineering: Textbook. Kyiv: National Academy of Sciences of Ukraine, 2008. 319 p.

3. Tytenko S. Web-oriented information systems. Kyiv: NTU «Kyiv Polytechnic Institute», 2015. 51 p.

4. Stadnyk Yu. Technologies for creating software and intelligent systems. Lviv: Ivan Franko Lviv National University, 2021. 46 p.

5. Site structure: main types and rules for their development, available at: https://webtune.com.ua/statti/web-rozrobka/struktura-sajtu/ (Accessed 16 October 2023).

6. Gilberto D., Jesslyn A., Afrianto Y. Analysis and Design of Web-based Information System for Coffeeshop Management using Design Thinking Methodology: Case of Kopi KurangLebih. *Journal of information systems and informatics*, 2023. Vol. 5. № 1. P. 217 – 231. DOI: https://doi.org/10.51519/journalisi.v5i1.455.

7. Fruhlinger J. What is JavaScript? The full stack programming language. InfoWorld, available at: https://www.infoworld.com/article/3441178/what-is-javascript-the-full-stack-programming-language.html (Accessed 14 October 2023).

8. Yao J. Web-based support systems. Springer, 2010. 464 p.

9. Khan B. H. Web-Based training. Educational Technology Pubns, 2003. 599 p.

10. Yulianto H. D., Fauzi R. Design of Web-based Online Sales Information System. *IOP Conf. Series: Materials Science and Engineering*, 879 (2020) 012007. IOP Publishing. DOI: http://dx.doi.org/10.1088/1757-899X/879/1/012007.

11. Ali A.I. Kosba E., El-sonbaty Ya. A web-based system to enhance lecture interaction: Case Study at Sultan Qaboos University (SQU). *International Journal of Computer Applications*, 2016. 135(9). P. 36 – 43. DOI: http://dx.doi.org/10.5120/ijca2016908508.

12. Bakery, available at: https://bakery-ck.com.ua/ (Accessed 18 October 2023).

13. Blue cup coffee shop, available at: https://bluecupcoffeeshop.choiceqr.com/ menu/section:menyu-kuhni/do-vina (Accessed 15 October 2023).

14. Aroma kava – network of coffee shops throughout Ukraine, available at: https://aromakava.ua/ (Accessed 17 October 2023).

15. What is e-commerce, available at: https://hub.kyivstar.ua/news/shho-take-e-commerce/ (Accessed 18 October 2023).

16. Regant F., Jansen W. Developing Web-Based Point of Sales Application Encryption on DBMS for Culinary Industry. *Journal of information systems and informatics*, 2023. Vol. 5. №. 3. P. 1020 – 1032. DOI: http://dx.doi.org/10.51519/journalisi.v5i3.544.

17. Singgalen Ye.A., Sutresno S.A. Digital Innovation Design of Tourism Destination Marketing Website Using Design Thinking Method. *Journal of information systems and informatics*, 2023. Vol. 5, No 2. P. 428 – 444. Vol. 5. № 3. DOI: http://dx.doi.org/10.51519/journalisi.v5i2.464.

18. About CSS, available at: https://dbpedia.org/page/CSS (Accessed 11 October 2023).

19. What is a marketplace and how does it differ from a regular online store, available at: https://kovel.tv/commerce/52-csho_take_marketplejs_ta_chim_vin_vidriznyaetsya_vid_ zvichajnogo_internet_magazina.html (Accessed 17 October 2023).

20. What Is A Marketplace?, available at: https://www.shopery.com/insights/what-is-a-marketplace (Accessed 19 October 2023).

21. Shwake E. What is a landing page? Here's everything you need to know, available at: https://www.wix.com/blog/what-is-a-landing-page (Accessed 17 October 2023).

22. Tkachenko O.A., Tyshura O.M. The «Cafeteria» system is cash register management software in the service sector. Materials of the 4th International science and practice conf. *«Information technologies and digital economy»* (Kyiv, May 4–5, 2023). P. 150 – 152.

**СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 25.09.2023**

*УДК 004.056*
*DOI: https://doi.org/10.53920/ITS-2023-2-9*

**Юрій Валентинович ДЕМЧЕНКО,**
старший викладач,
Державний університет інфраструктури та технологій,
Київський інститут залізничного транспорту,
факультет Інфраструктури та рухомого складу залізниць,
кафедра вагонів та вагонного господарства
ORCID ID: 0009-0007-6058-1264

# БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ ВАГОННОГО ГОСПОДАРСТВА

***Стаття присвячена опрацюванню питань забезпечення інформаційної безпеки вагоноремонтних підприємств, захисту систем зберігання і обробки комп'ютерних даних, при яких зберігається повна конфіденційність, доступність і цілісність всієї наявної інформації. Головною умовою нормальної роботи повинні бути максимальні знання про всі структурні складові та критерії інформаційної безпеки комп'ютерних даних. Так із кожним наступним роком зростає нагальна роль інформаційної безпеки виробничих процесів, оскільки наше суспільство вступило в нову епоху інформаційних війн, де цінність первинної інформації безперечно має пріоритет. Хоча навіть проста інформація – це не тільки товар, а й значний інструмент для маніпуляції, порушення технологічних процесів, створенню конфліктів, отримання грошової винагороди. Отже, інформаційна безпека – це комплекс захищеності всієї інформації та пов'язаної з нею виробничої інфраструктури, від всіх, як випадкових, так і навмисних дій, внаслідок чого проходить збій у наявній інформації, а також в інфраструктурі підтримки. Метою стороннього проникнення в комп'ютерні мережі підприємства є нанесення шкоди або знищення наявних даних, заволодіння конфіденційною інформацію з можливим подальшим використанням у противоправних цілях.***

***Ключові слова:*** *інформаційні системи, інформаційна безпека, сфера обробки інформації, захист, шкідливі програми, конфіденційність, вагонне господарство, антивірусні технології, збитки, заподіяння шкоди, високі технології.*

**Yurij ДЕМЧЕНКО**
Senior teacher,
State university of infrastructure and technologies
Kyiv institute of railway transport,
faculty of Infrastructure and movable to composition of railways,
department of carriages and carriage economy

# SECURITY OF INFORMATION SYSTEMS OF THE CARRIAGE ECONOMY

***The article is devoted to the study of the issues of ensuring the information security of railcar repair enterprises, protection of computer data storage and processing systems, which preserve complete confidentiality, availability and integrity of all available information. The main condition for normal operation should be maximum knowledge of all structural components and criteria for information security of computer data. Thus, the role of information security in production processes is growing every year, as our society has entered a new era of information wars, where the value of primary information is undoubtedly of paramount importance. However, even simple information is not only a commodity, but also a significant tool for manipulation, disruption of technological processes, creation of conflicts, and obtaining monetary rewards. Therefore, information security is a security complex of all information and related production infrastructure from all, both accidental and intentional actions, resulting in a failure in the available information and support infrastructure. The purpose of an unauthorized intrusion into an enterprise's computer networks is to damage or destroy existing data, to obtain confidential information with the possible subsequent use for illegal purposes.***

***Keywords:*** *information systems, information security, information processing, protection, malware, privacy, railcar industry, anti-virus technologies, losses, damage, high technology.*

**Постановка проблеми.** У сучасному світі інформаційні системи удосконалюються високими темпами, а інформаційна безпека та політика захисту набувають все більш масштабного характеру, висуваючи цю проблему на перший план. Глобалізація посилюється себе з кожним роком, але окрім позитивних моментів, виникають і дуже серйозні негативні процеси, до яких цивілізований

світ виявився неготовим. Суттєво зросла роль інформаційної безпеки виробничих процесів, оскільки цифрова інформація стала як продуктом, так і сировиною, яку обробляють, виробляють, готують, продають і дуже часто крадуть для отримання прибутку. Нині в основному інформаційну безпеку цифрових технологій визначають через комп'ютерну безпеку. Захист інформації, що знаходиться всередині комп'ютера, є у рази складнішим ніж убезпечення таємниці стандартного листування. Існуючі проблеми інформаційної безпеки актуальні й вимагають більш системного, поглибленого, постійного вивчення, аналізу та удосконаленню. Система інформаційної безпеки транспортної галузі виступає одним із основних елементів у системі національної безпеки. На сьогоднішній день, коли продовжується збройна агресія зі сторони росії, логістична складова нашої держави в основному опирається на залізничний транспорт. У даній статті розглянемо методи забезпечення інформаційної безпеки підприємств вагонного господарства, як одного елементу функціонування транспортної галузі.

**Аналіз останніх досліджень і публікацій.** У сучасній науковій літературі відсутній єдиний погляд на зміст поняття «інформаційна безпека». Існує необхідність уточнення поняття «інформаційна безпека», що допоможе осмисленню нових її аспектів, дозволить ширше розкрити суть і надати поняттю системного характеру. Провідні спеціалісти з цього питання ведуть публічну дискусію, особливо щодо характеристик існуючих небезпек, їхньої внутрішньої структури. Принципи побудови системи забезпечення національної безпеки висвітлюється в працях В. Богуша та О. Юдина «Інформаційна безпека держави» [1]. Автори висвітлюють загальні принципи побудови системи інформаційної безпеки без детальних вказівок на її впровадження на підприємствах, насамперед транспортної галузі. Система інформаційної безпеки підприємств вагонного господарства повинна в повній мірі відображати захищеність будь-яких корпоративних інтересів в існуючій інформаційній сфері від внутрішніх і зовнішніх загроз. Цікаві напрацювання стосовно інформаційної безпеки підприємств вагонного господарства викладені в роботах провідних вчених, а саме Черняка Г.Ю., Щербини Ю.В [2, 3], Обуховського В.В., Іщенко В.М., Щербини Ю.В. [4].

**Мета статті.** Аналіз методів забезпечення інформаційної безпеки підприємств вагонного господарства.

**Виклад основного матеріалу дослідження.** Інформаційна безпека будь-якої організації – це стан захищеності інформаційного середовища цієї організації, що забезпечує її формування, використання і розвиток. У сучасному соціумі інформаційна сфера має дві складові: інформаційно-технічну (штучно створений людиною світ техніки, технологій тощо) та інформаційно-психологічну (природний світ живої природи, що включає і саму людину). Інформаційну безпеку зазвичай можна розділити на дві складові частини: інформаційно-технічну та інформаційно-психологічну (психофізичну) безпеки.

Типова модель інформаційної безпеки складається з трьох категорій: конфіденційність (це стан будь-якої інформації, при якому доступ до неї здійснюють суб'єкти, які мають на це право); цілісність (недопущення недозволеної модифікації наявної інформації); доступність (недопущення тимчасового чи постійного приховування цієї інформації від працівників, які мали на це право доступу).

Існують й інші необов'язкові категорії безпеки: неспростовність (неможливість відмови від авторства); підзвітність (отримання повного співпадіння суб'єкта доступу та індикації всіх його дій); достовірність (відповідності передбаченому результату); автентичність або справжність (показує, що суб'єкт аналогічний заявленому).

Виділяється декілька категорій дій, що завдають шкоди інформаційній безпеці підприємства.

Перше, це дії, здійснювані авторизованими користувачами. Під цю категорію потрапляють цілеспрямована крадіжка або знищення даних на сервері, або робочій станції.

Друге, це методи впливу, що здійснюються різними хакерами, тобто людьми, які скоюють комп'ютерні злочини, у тому числі під час будь-якої конкурентної боротьби між компаніями. До цих методів відносять проникнення без дозволу в чужі комп'ютерні мережі або DoS-атаки. DoS-атака (від англ. Denial of Service – відмова в обслуговуванні) і DDoS-атака (від англ. Distributed Denial of Service – розподілена атака типу «відмова в обслуговуванні») – атака на обчислювальну систему з метою довести її до відмови, тобто створення таких умов, при яких легітимні (правомірні) користувачі системи не можуть одержати доступ до надаваємих системою ресурсів, або цей доступ стає ускладнений [5]. Іншими

словами це зовнішня атака на комп'ютерні мережі, що відповідають за ефективну роботу підприємства. Злочинці організовують лавинну, цілеспрямовану відправку даних на вузли інформаційних систем для їх перевантаження і виведення на якийсь час із ладу. У результаті це тягне за собою порушення або припинення роботи в бізнес-процесах компанії, на яку організована атака, та втрату її клієнтів, репутації, фінансових збитків.

Третє, це комп'ютерні віруси, категорія електронних методів впливу та інші шкідливі програми. Це реальна небезпека для всіх сучасних інформаційних систем, що часто використовують комп'ютерні мережі, електронну пошту та інтернет. Проникнення вірусу на вузли корпоративної мережі призводить до цілковитого порушення їх функціонування, значної втрати робочого часу, повної втрати наявних даних, зникнення усіх особистих даних, що були в комп'ютерній мережі та навіть фінансових засобів підприємства. Вірусна програма, яка потрапила в будь-яку корпоративну мережу підприємства, може дати злочинцям частковий або повний контроль над діяльністю організації [7, 10].

Четверте, це спам, що за кілька років перетворився на значну загрозу безпеці. Електронна пошта стала одним із головних каналів поширення різних шкідливих програм. Багато часу витрачається на перегляд різного роду спаму та його подальше видалення, що викликає у співробітників підприємства почуття типічного психологічного дискомфорту. Приватні особи й організації стають основними жертвами шахрайських схем. Разом зі спамом можливо нерідко видалення важливої кореспонденції, що призводить до втрати всієї корисної інформації. Небезпека втрати будь-якої кореспонденції збільшується при використанні різноманітних простих та складних методів для фільтрації спаму.

П'яте, це так звані «побутові» загрози. Можливий вплив на інформаційну безпеку вагоноремонтних підприємств різноманітних зовнішніх факторів таких, як: неправильна експлуатація або зберігання, що може стати причиною втрати даних; крадіжка як комп'ютерів, так і потрібних носіїв інформації; типові та не типові форс-мажорні обставини (затоплення, коротке замикання, механічне пошкодження) тощо.

Наявність різноманітної системи інформаційної безпеки підприємства є найважливішою умовою високої конкурентоспроможності та ефективної працеспроможності.

Всесвітня інформаційна мережа (павутина) зростає великими кроками, і як показує статистика, користувачів з плином часу стає тільки більше. Доступна інформація у всесвітній мережі займає всі сторони життєдіяльності як людини, так і нашого суспільства. Користувачі мережі довіряють цій формі своє особисте життя і свою трудову діяльність.

Фахівці галузі звертають увагу на те, що головна причина несанкціонованого проникнення в комп'ютерні мережі підприємства – це елементарна непідготовленість користувачів і їх порівняно невеликий досвід у сфері інформаційних технологій. Це зумовлено стрімким розвитком мережі інтернет та мережевих технологій.

Слід зазначити, що близько дев'яноста відсотків від всіх проникнень у комп'ютери шкідливих програм, припадає на інтернет (через електронну пошту та постійний перегляд Web-сторінок). Особливо серед таких програм виділяється цілий клас паразитів, таких як інтернет-черв'яки [6]. Вони можуть поширюватися автономно від механізму роботи комп'ютера й виконувати свої основні завдання щодо зміни програмних налаштувань комп'ютера-жертви, знищують адресну книгу або іншу цінну інформацію, дезорієнтують самого користувача, запускають розсилку з робочого комп'ютера за адресами, які були взяті із адресної книжки, можуть зробити комп'ютер підприємства стороннім ресурсом або навіть забирати значну частину робочих ресурсів для своїх цілей, а також самоліквідуватися, знищуючи при цьому на дисках всі файли.

Усі викладені проблеми, потрібно вирішувати за допомогою опрацьованого на вагоноремонтному підприємстві документа, що повністю відображає політику інформаційної безпеки. В цьому документі мають бути чітко виписані такі положення: як організована робота з інформацією вагоноремонтного підприємства; хто до неї має вільний доступ; як проходить копіювання і зберігання даних; який режим роботи на персональних комп’ютерах; наявність реєстраційних документів на обладнання персональних комп’ютерів та їх програмне забезпечення; вимоги до приміщення, де розташовується персональні комп’ютери і робочі місця користувачів; наявність робочих інструкцій і їх технічної документації.

Необхідно обов’язково відслідковувати новинки в технічних та інформаційних системах, які публікуються у фахових виданнях, і стежити за подіями, що обговорюються на відповідних семінарах та в соціальних мережах.

За нагальної потреби підключення інформаційних систем, інформаційно-телекомунікаційних мереж та всіх засобів обчислювальної техніки підприємств вагонного господарства до інформаційно-телекомунікаційних мереж міжнародного інформаційного обміну може проводиться тільки з використанням спеціальних, сертифікованих засобів захисту інформації, в тому числі шифрувальних (криптографічних) засобів, які пройшли порядок сертифікації, установлений законодавством.

Забезпечення інформаційної безпеки підприємств вагонного господарства повинно вирішуватися системно. Мають застосовуватися різні засоби захисту. Це фізичні, апаратні, програмні, організаційні та інші. Вони можуть застосовуватися як одночасно, так і під централізованим управлінням. Але при цьому всі компоненти системи мають взаємодіяти між собою і забезпечувати захист від зовнішніх та внутрішніх загроз.

У даний час ми маємо великий вибір методів забезпечення інформаційної безпеки підприємства це: засоби ідентифікації і автентифікації користувачів комп'ютерів; засоби шифрування інформації, що зберігається в комп'ютерах; міжмережеві екрани комп'ютерів; віртуальні приватні мережі; засоби тематичної фільтрації; інструменти перевірки цілісності вмісту дисків комп'ютерів; засоби антивірусного захисту програм; системи виявлення слабких місць мереж і аналізатори мережевих атак.

Всі перелічені засоби можуть бути використані самостійно, а також разом з іншими, що робить можливим створення систем інформаційного захисту в мережі різної складності і конфігурації, незалежно від платформ, які використовуються.

Основними елементами безпеки інформації називають авторизацію та ідентифікацію. Функція ідентифікації при доступу до інформаційних активів дисків комп'ютерів відповідає на запитання: «хто ви є?», «де ви є?», «ви користувач мережі?». А функція авторизації дозволяє доступ до ресурсів конкретного користувача. Роль функції адміністрування є в наданні користувачеві певних ідентифікаційних особливостей у межах мережі, розгляду та наданні кількості дій, допустимих для нього.

Шифрування даних дає змогу значно зменшити втрати при доступу стороннього користувача до даних, що зберігаються на стаціонарному носії, а також зчитування інформації, яка пересилається електронною поштою. Даний засіб ефективно забезпечує

захист конфіденційності наявної інформації. Високий показник криптостійкості і законне використання – основні вимоги до систем шифрування.

Принцип, на якому побудована дія міжмережевих екранів, ґрунтується на перевірці кожного масиву даних на ідентичність, вихідної та вхідної IP-адреси, дозволених наявних баз адресатів. Як наслідок значно розширюються можливості міжмережевих екранів інформаційних мереж за контролем переміщення даних.

Аналізуючи криптографію та діючі міжмережеві екрани, звертаємо увагу на інші захищені мережі приватних власників таких, як віртуальна приватна мережа (Virtual Private Network – VPN). При їх використанні вирішуються всі проблеми цілісності й конфіденційності даних при пересилці відкритими або напіввідкритими комунікаційними каналами. Впровадження VPN зводиться до наступних завдань, що можна розділити: на захист усіх інформаційних потоків у структурі підприємства вагонного господарства (шифрування всієї інформації на виході з внутрішньої мережі); повністю захищений доступ користувачів мережі до наявних інформаційних масивів підприємства, що використовує інтернет; захист наявних інформаційних потоків всередині корпоративної мережі підприємства.

Фільтрація кількості вхідної та вихідної інформації через електронну пошту – один із найефективних способів захисту конфіденційної інформації від її втрати. Постійна перевірка всіх поштових повідомлень і включень у них на основі правил, які встановлені на підприємстві, забезпечує ефективний захист підприємства вагонного господарства від відповідальності по судовим позовам і дає можливість захистити своїх співробітників від наявного спаму. Функція тематичної фільтрації може переглядати і перевіряти файли всіх наявних форматів, включаючи стислі і графічні. Під час цієї операції пропускна спроможність робочої мережі не змінюється.

Усі наявні зміни на працюючому сервері відслідковуються адміністратором цієї мережі або авторизованим виконавцем, використовуючи елементи технології перевірки незмінності вмісту на жорсткому диску. В результаті це допомагає виявляти різні дії з файлами, такі як зміна, видалення, відкриття, та ефективно ідентифікувати появу вірусів, виявити несанкціонований доступ до мережі або крадіжку даних користувачами, що мали до нього доступ.

Новітні антивірусні технології виявляють майже всі відомі користувачам вірусні програми, використовуючи порівняння наявного коду підозрілого робочого файлу із зразками, що знаходяться в антивірусній базі комп'ютера. Розроблено свіжі моделюючі програми технології поведінки, які ефективно виявляють нові вірусні програми.

Щоб протидіяти техногенним і природним загрозам інформаційній безпеці на підприємстві вагонного господарства, повинен бути спроєктований і впроваджений цілий набір різних процедур для протидії надзвичайним ситуаціям. Це може бути фізичний захист усієї робочої апаратури від пожежі та її наслідків, що може мінімізувати збитки, якщо виникне така ситуація. Один із надійних методів захисту інформації від втрати даних – постійне резервне копіювання матеріалу при чіткому виконанні встановлених правил.

На підприємствах вагонного господарства на сьогоднішній день конструкторська, службова, робоча, технологічна документація відпрацьовується на комп'ютерній техніці. Вся робоча інформація обробляється, архівується виключно на електронних носіях інформації. На підприємствах існує можливість для скритого, не дозволеного копіювання, передачі, видалення, зміни, знищення електронних даних на носіях інформації. Безпечне, надійне, безперервне функціонування баз даних в інформаційних комп'ютерних системах підприємства, є основою його стабільної роботи.

Причини, які можуть викликати зміну або знищення інформації, що знаходиться в комп'ютерах автоматизованих робочих місць, мають наступне походження, а саме ненавмисна помилка працівників, свідоме завдання шкоди зловмисниками, шкідливим програмним продуктом, неполадками в самому комп'ютері, схованими елементами недоробленого програмного забезпечення, виведення з ладу технічних пристроїв автоматизованих робочих місць, аваріями на електричних та інженерних мережах, стихійними лихами.

Статистика показує, що від п’ятдесяти до вісімдесяти відсотків випадків втрат даних комп'ютера становлять помилкові або не зовсім професійні дії працюючого персоналу. Втрата даних з технічних причин – від п'ятнадцяти до двадцяти п'яти відсотків.

Помилки в діях працівників можна пояснити низьким рівнем дисципліни, слабкою професійною підготовкою, безвідповідальністю, недостатніми знаннями програмного забезпечення і самої комп'ютерної техніки. Незважаючи на причини втручання чи втрати будь-якої із необхідних інформацій для підприємства, од-

нозначно буде завдана шкода бізнесовому іміджу підприємства та його фінансовій репутації.

При розробці проєкту автоматизованих робочих місць на підприємствах вагонного господарства необхідно в обов'язковому порядку враховувати ці загрози і розробляти заходи по мінімізації її наслідків. Комплекс заходів для досягнення позитивних цілей включає в себе організаційні, програмні, інженерно-технічні методи захисту інформації та її збереження.

Найвідоміші методи захисту комп'ютерної інформації – захист за допомогою спеціальних програм або програмні методи захисту, що являє собою комплекс програм різного призначення, алгоритмів для забезпечення безперебійної роботи комп'ютерної техніки. Вони виконують контроль та розмежування доступу до наявної інформації з видаленням непідконтрольних дій по відношенню до неї. Програмні методи захисту передбачають простоту, реалізацію, універсальність, гнучкість, пристосованість, варіативність у налаштуванні.

Щоб захистити комп'ютерну інформацію спеціалісти використовують антивірусні програми разом із профілактикою і постійною діагностикою, це ускладнює проникнення комп'ютерного вірусу в інформаційну систему, очищає уже заражені робочі файли та диски і не допускає нового зараження.

Ідеальних антивірусних програм, які вчасно виявляли б і знищували будь-який вірус, спеціалісти ще не зробили. Найбільш надійний антивірусний захист має багаторівнева система, що має наступні складові: використання тільки ліцензійного програмного продукту, періодичне резервне копіювання інформаційних матеріалів і програм, постійна перевірка уже отриманих даних на наявність вірусних програм, використання нових антивірусних продуктів під час перевірки своїх носіїв інформації при перенесенні на них інших масивів інформації, а також при переформатуванні.

Існує думка, що найефективнішим способом захисту інформації під час передачі її через комп'ютерні лінії є шифрування. Його називають криптографічним методом захисту інформації.

Технічний захисту інформації – це пристрої та способи, що попереджають і убезпечують від крадіжок, нецільового використання, приведення в непридатний стан мережевого устаткування та автоматизованих робочих місць. Для цього встановлюють на вікна та двері металеві решітки, огорожі, турнікети, кодові замки на двері, різні системи відеоспостереження і сигналізації. Також всі робочі підрозділи забезпечуються системами зв'язку, управління

контролю доступом, телекомукаційними, пожежними, охоронними системами та іншими інженерними пристроями [8].

Важливим для інформаційної безпеки є досягнення стану її захищеності, тобто створення і підтримка відповідних інженерно-технічних потужностей та інформаційної організації, що відповідають реальним і потенційним загрозам [9].

Організаційні методи засоби захисту інформації включають у себе регламентацію виробничої праці та на нормативно-правовій базі взаємовідносини між виконавцями, що приводить до зменшення чи суттєво ускладнення незаконного привласнення інформації конфіденційного характеру. Вони ще організовують роботу з персоналом, документацією, синхронізацією різних технічних засобів, а також проведення аналізу різних загроз інформаційній безпеці підприємства, організовують режим охорони.

Для забезпечення захисту зберігання та обробки інформації в автоматизованих робочих місцях при проникненні в їх робочу структуру розділяють доступ до об'єктів усіх суб'єктів.

Організаційні методи захисту інформації включають в себе аналітичну роботу з кадрами, вибірковий режим доступу, цілісність документообігу, постійне використання технічних засобів забезпечення безпеки, аналітичний аналіз по недопущенню зовнішніх та внутрішніх загроз інформаційній безпеці, призначення відповідальних за наявне обладнання, проведення періодичних та цільових інструктажів працюючого персоналу.

Організаційні методи захисту інформації мають три складові: контроль доступу персоналу, обмеження доступу персоналу, розмежування доступу персоналу.

Обмеження доступу персоналу до комп'ютерних машин – такий принцип роботи, щоб виключити будь-який доступ до цієї системи сторонніх осіб. Один із варіантів, це розмістити автоматизовані робочі місця в штучно ізольованому приміщенні. При тривалому простої системний блок користувача і всі носії інформації рекомендується розміщати в сейфі.

Розмежування доступу персоналу полягає в розподілі всієї інформації на якісні частини й дозволу доступу до неї згідно із затвердженими функціональними та посадовими інструкціями персоналу. Суть завдання розмежування доступу – це захист інформації від зловмисника, що має допуск до роботи в даній комп'ютерній мережі.

Розділення інформації може проходить відповідно до наступних критеріїв, таких як: функціональне призначення, ступінь важ-

ливості та інші. При проведенні розмежування доступу виконують наступні правила: технічне обслуговування всього обладнання під час експлуатації повинне проводитись спеціалістами, які не мають доступу до захищеної інформації; будь-яку заміну програмного забезпечення комп'ютерної мережі покладають на спеціально, призначеного для цієї цілі, спеціаліста.

Контроль доступу – впізнання автентичності особи та цілковита фіксація моменту початку роботи. При цьому, на початку усім, хто допускається до закритої інформації, надається унікальний образ, число, ім'я – ідентифікацію. Процес перевірки або впізнання дійсності особи, чи дійсно особа є та за кого себе подає називається автентифікацією [11].

Суб'єктами автентифікації та ідентифікації в комп'ютерній мережі можуть бути технічний засіб, документ, працівник тощо. Працівником може виступати оператор комп'ютерної мережі, користувач, адміністратор.

Автентифікація може проходити відносно технічних засобів, персоналу, документів. Найвідоміший метод – надання паролю та запиту при новому вході в комп'ютерну мережу. Приклад автентифікації технічних засобів – поставити автентифікаційний термінал для входу в комп'ютерну мережу працівника. Описана процедура також проходить за допомогою паролю.

**Висновки та пропозиції.** Проведений аналіз методів забезпечення інформаційної безпеки підприємств вагонного господарства дає чітке розуміння в необхідності системного та комплексного підходу до зазначеної проблеми. Одним з необхідних кроків до забезпечення інформаційної безпеки відносяться організаційні заходи захисту інформації, що являють собою сукупність заходів щодо підбору, перевірки и навчання персоналу, який бере участь у всіх стадіях інформаційного процесу підприємств вагонного господарства. Також на вищезазначених підприємствах необхідно дотримуватися постійного контролю за джерелами виникнення потенційних загроз та у відповідь сучасним викликам здійснювати пошук й застосовувати найкращі рішення у здійсненні захисту інформації різними формами та способами. Форми і способи забезпечення інформаційної безпеки утворюють власне інструмент, за допомогою якого засоби інформаційної безпеки вирішують весь комплекс завдань із захисту життєво важливих інтересів підприємств вагонного господарства.

**ЛІТЕРАТУРА**

1. Юдін О. К., Богуш В. М. Інформаційна безпека держави: навч. посіб. Харків: Консум. 576 с.

2. Черняк Г. Ю., Щербина Ю. В. Розробка моделі пасажирського вагона для досліджень динаміки в програмному комплексі «Универсальный механизм». *Збірник наукових праць Київського університету економіки і технологій транспорту. Серія «Транспортні системи і технології»*. Київ, 2007. Т 12. С. 75–82.

3. Черняк Г. Ю., Щербина Ю. В. Базова комп'ютерна модель просторової динаміки пасажирського вагона для швидкісного руху. *Залізничний транспорт України*. 2012. № 6. С. 55–58.

4. Обуховський В. В., Іщенко В. М., Щербина Ю. В. Аналіз автоматизованих систем керування вагонів метрополітену (пасажирських поїздів). *The 3 rd International scientific and practical conference «Topical aspects of modern scientific research»(November 23–25, 2023)*. CPN Publishing Group, Tokyo, Japan. 2023. 725 p.

5. Краснобрижий І. В. Види та методики реалізації dos та ddos атак на державні автоматизовані системи, а також можливі шляхи боротьби з ними. *Економічна та інформаційна безпека: проблеми та перспективи:* матеріали Всеукр. наук.-практ. конф., м. Дніпро, 14 квіт. 2017 р. Дніпро, 2017. С. 89-94.

6. Авраменко В. С., Авраменко А. С. Основи операційних систем : навч. посіб. Черкаси, 2018. 524 с.

7. Буров Є. В. Комп'ютерні мережі : підручник. Львів: «Магнолія 2006», 2010. 262 с.

8. Нормативний документ системи технічного захисту інформації. Захист інформації на об'єктах інформаційної діяльності. Положення про категоріювання об'єктів, де циркулює інформація з обмеженим доступом, що не становить державної таємниці НД ТЗІ 1.6-005-2013 : наказ Адміністрації Державної служби спеціального зв'язку та захисту інформації України від 15.04.2013 № 215.

9. Крюков О. І. Інформаційна безпека держави в умовах глобалізації. *Державне будівництво*. 2007. № 2.

10. Петрик В. М., Галамба М. В. Інформаційна безпека України: поняття, сутність та загрози. *Юридичний журнал*. 2006. № 11. С. 49–52.

11. Критерії оцінювання захищеності інформації в комп'ютерних системах від несанкціонованого доступу : нормативний документ НД ТЗІ 2.5-004-99 від 28 квіт. 1999 р. № 22.

**REFERENCES**

1. Yudin O. K., Bohush V. M. Informatsiina bezpeka derzhavy: navch. posib. Kharkiv: Konsum. 576 s.

2. Cherniak H. Yu., Shcherbyna Yu. V. Rozrobka modeli pasazhyrskoho vahona dlia doslidzhen dynamiky v prohramnomu kompleksi «Unyversalnыi mekhanyzm». Zbirnyk naukovykh prats Kyivskoho universytetu ekonomiky i tekhnolohii transportu. Seriia «Transportni systemy i tekhnolohii». Kyiv, 2007. T 12. S. 75–82.

3. Cherniak H. Yu., Shcherbyna Yu. V. Bazova kompiuterna model prostorovoi dynamiky pasazhyrskoho vahona dlia shvydkisnoho rukhu. Zaliznychnyi transport Ukrainy. 2012. № 6. S. 55–58.

4. Obukhovskyi V. V., Ishchenko V. M., Shcherbyna Yu. V. Analiz avtomatyzovanykh system keruvannia vahoniv metropolitenu (pasazhyrskykh poizdiv). The 3 rd International scientific and practical conference "Topical aspects of modern scientific research"(November 23–25, 2023). CPN Publishing Group, Tokyo, Japan. 2023. 725 p.

5. Krasnobryzhyi I. V. Vydy ta metodyky realizatsii dos ta ddos atak na derzhavni avtomatyzovani systemy, a takozh mozhlyvi shliakhy borotby z nymy. Ekonomichna ta informatsiina bezpeka: problemy ta perspektyvy: materialy Vseukr. nauk.-prakt. konf., m. Dnipro, 14 kvit. 2017 r. Dnipro, 2017. S. 89 94.

6. Avramenko V. S., Avramenko A. S. Osnovy operatsiinykh system : navch. posib. Cherkasy, 2018. 524 s.

7. Burov Ye. V. Kompiuterni merezhi : pidruchnyk. Lviv: «Mahnoliia 2006», 2010. 262 s.

8. Normatyvnyi dokument systemy tekhnichnoho zakhystu informatsii. Zakhyst informatsii na obiektakh informatsiinoi diialnosti. Polozhennia pro katehoriiuvannia obiektiv, de tsyrkuliuie informatsiia z obmezhenym dostupom, shcho ne stanovyt derzhavnoi taiemnytsi ND TZI 1.6-005-2013 : nakaz Administratsii Derzhavnoi sluzhby spetsialnoho zviazku ta zakhystu informatsii Ukrainy vid 15.04.2013 № 215.

9. Kriukov O. I. Informatsiina bezpeka derzhavy v umovakh hlobalizatsii. Derzhavne budivnytstvo. 2007. № 2.

10. Petryk V. M., Halamba M. V. Informatsiina bezpeka Ukrainy: poniattia, sutnist ta zahrozy. Yurydychnyi zhurnal. 2006. № 11. S. 49–52.

11. Kryterii otsiniuvannia zakhyshchenosti informatsii v kompiuternykh systemakh vid nesanktsionovanoho dostupu : normatyvnyi dokument ND TZI 2.5-004-99 vid 28 kvit. 1999 r. № 22.

**СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 02.12.2023**

Науковий журнал

# «IT SYNERGY»

Заклад вищої освіти
«Міжнародний науково-технічний університет
імені академіка Юрія Бугая»
Вип. 2 (5)

Київ, 2023

Відповідальний за випуск: О. І. Бражнікова
Дизайн та верстка: А. В. Дученко



Підписано до друку 29.12.2023.
Формат 60x90/16. Ум. друк. арк. – 8.51. 148 с. Тираж 100. Зам. № 601.

Друк: Видавництво ТОВ «А-ЦЕНТР».
Свідоцтво про реєстрацію Серія ДК № 599 від 14.01.2001 р.
04112, м. Київ, вул. Івана Гонти, 3А.