

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний авіаційний університет  
Факультет кібербезпеки, комп'ютерної та програмної  
інженерії



СУЧАСНІ ТЕНДЕНЦІЇ РОЗВИТКУ  
СИСТЕМНОГО ПРОГРАМУВАННЯ

Тези доповідей науково-практичної конференції

25-26 листопада 2021 року

Київ 2021



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний авіаційний університет  
Факультет кібербезпеки, комп'ютерної та програмної  
інженерії

СУЧАСНІ ТЕНДЕНЦІЇ РОЗВИТКУ  
СИСТЕМНОГО ПРОГРАМУВАННЯ

Тези доповідей науково-практичної конференції

25-26 листопада 2021 року

Київ 2021

Сучасні тенденції розвитку системного програмування: тези доповідей науково-практичної конференції, м. Київ. – 25-26 листопада 2021 р., Національний авіаційний університет, – К.: НАУ, 2021. – 56 с.

Збірник містить тези доповідей, які були представлені на конференції «Сучасні тенденції розвитку системного програмування».

У доповідях розглянуті наукові, практичні та методичні питання системного програмування: розробка компонентних систем, систем штучного інтелекту, методи та алгоритми розподілу задач за рівнями реалізації та обчислювальними ресурсами, підвищення ефективності обміну інформацією та захисту даних в обчислювальних системах.

Для фахівців з програмування, електроніки та схемотехніки.

#### **Редакційна колегія:**

**Литвиненко О.Є.** – д.т.н., професор, завідувач кафедри комп'ютеризованих систем управління факультету кібербезпеки, комп'ютерної та програмної інженерії НАУ (Україна, Київ);

**Кучеров Д.П.** – д.т.н., с.н.с., професор кафедри комп'ютеризованих систем управління факультету кібербезпеки, комп'ютерної та програмної інженерії НАУ (Україна, Київ);

**Артамонов Є.Б.** – к.т.н., доцент кафедри комп'ютеризованих систем управління факультету кібербезпеки, комп'ютерної та програмної інженерії НАУ (Україна, Київ).

*Затверджено вченою радою факультету кібербезпеки, комп'ютерної та програмної інженерії Національного авіаційного університету (протокол № 10 від 13.12.2021 р.)*

## ЗМІСТ

<b>СЕКЦІЯ 1. РОБОТОТЕХНІКА, ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ, СИСТЕМИ ОБРОБКИ ЗОВНІШНІХ ДАНИХ</b> .....	6
Коба О.В. ОПТИЧНІ БУФЕРИ ТА МАТЕМАТИЧНІ МОДЕЛІ ЇХ ФУНКЦІОНУВАННЯ .....	6
Крант Д.В. МЕТОДИ ОРГАНІЗАЦІЇ ДОСТУПУ ДО SAN-ШИН АВТОМОБІЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ З ANDROID- ДОДАТКІВ .....	8
Волнейкін М.О. ІДЕНТИФІКАЦІЯ ОСОБИ ЧЕРЕЗ СИСТЕМУ ВІДЕОСПОСТЕРЕЖЕННЯ.....	9
Горбатенков О.В. МЕТОД ЗАСТОСУВАННЯ ДАТЧИКІВ ДЛЯ ОПТИМІЗАЦІЇ ВИКОРИСТАННЯ РЕСУРСІВ .....	10
Лазебний А.С. МЕТОД РЕАЛІЗАЦІЇ СИСТЕМИ СПОСТЕРЕЖЕННЯ ЗА РОСЛИНАМИ У ТЕПЛИЦЯХ .....	11
Масловський Б.Г. ПІДСИСТЕМА ЕНЕРГОЗБЕРЕЖЕННЯ У СИСТЕМІ «ІНТЕЛЕКТУАЛЬНИЙ ДІМ».....	12
Симонов Д.М. САМОРОБНИЙ ПОРТАТИВНИЙ ПОДАВЛЮВАЧ РАДІОСИГНАЛІВ.....	14
Давидова А.С., Кашкевич С.О. МЕТОД РЕАЛІЗАЦІЇ КВАНТОВИХ ОБЧИСЛЕНЬ В МОВІ ПРОГРАМУВАННЯ Q# .....	15
Жванський А.І. АВТОМОБІЛЬНА КОМУНІКАЦІЙНА СИСТЕМА VEHICLE TO EVERYTHING (V2X) .....	16
Хлищиборщ П.О. КОДУВАННЯ ПОВІДОМЛЕНЬ У RFID СИСТЕМАХ .....	18
Мартінова О.П., Станко С.М. МЕТОДИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНО-РОЗВАЖАЛЬНИХ ПРОГРАМ ПРИ КЕРУВАННІ АВТОМОБІЛЕМ .....	19
<b>СЕКЦІЯ 2. ПАРАЛЕЛЬНІ І РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ.....</b>	<b>20</b>
Ткаченко В.Г., Гришко Н.О. АНАЛІЗ ВИКОРИСТАННЯ REDIS В РОЗПОДІЛЕНІЙ СИСТЕМІ ОБРОБКИ ЗАМОВЛЕНЬ .....	20
Марченко А.А. ПРИНЦИПИ ПОБУДОВИ РОЗПОДІЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ РОБОТИ МЕРЕЖІ РЕСТОРАНІВ .....	22
Артамонов Є.Б., Голего Н.М. МІКРОСЕРВІСНА АРХІТЕКТУРА E- LEARNING СИСТЕМ.....	23

Кравець М.О., Халімон Н.Ф. НЕРЕЛЯЦІЙНІ МОДЕЛІ БАЗ ДАНИХ ТА ВИМОГИ ACID .....	25
Артамонова К.Є., Яковенко Л.В. ОСОБЛИВОСТІ ВИКОРИСТАННЯ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ .....	26
<b>СЕКЦІЯ 3. МУЛЬТИПЛАТФОРМЕННІ ПРОГРАМНІ ЗАСОБИ, МОБІЛЬНІ ТЕХНОЛОГІЇ .....</b>	<b>27</b>
Бесараб І.В. АКТУАЛЬНІ ТЕХНОЛОГІЇ У РОЗРОБЦІ МОБІЛЬНОГО ПЗ.....	27
Бондаренко Б.В., Халімон Н.Ф. ВЕБ-ДОДАТОК ДЛЯ СЛУЖБ КРОВІ НА ПЛАТФОРМІ SPRING .....	28
Спілок В.Ю. МОДИФІКАЦІЯ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ АВТОМОБІЛЬНОГО ПОРТАЛУ .....	29
Стенякін І.А. ПОВЕДІНКОВІ ПАТТЕРНИ АУТЕНТИФІКАЦІЇ КОРИСТУ-ВАЧІВ В ОНЛАЙНОВИХ ЕЛЕКТРОННИХ СИСТЕМАХ НАВ-ЧАННЯ .....	31
Халімон Н.Ф. ВЕРТИКАЛЬНЕ ТА ГОРИЗОНТАЛЬНЕ МАСШТАБУВАННЯ БАЗ ДАНИХ.....	33
Остапенко В.О. ОСОБЛИВОСТІ 3D-ВІЗУАЛІЗАЦІЇ В НАВЧАЛЬНИХ СИСТЕМАХ .....	35
<b>СЕКЦІЯ 4. СПЕЦІАЛІЗОВАНІ ПРОГРАМНІ ЗАСОБИ.....</b>	<b>36</b>
Жарова О.В. МОДИФІКАЦІЯ СИСТЕМИ ФОРМУВАННЯ ІНДИВІДУАЛЬНОЇ ОСВІТНЬОЇ ТРАЄКТОРІЇ СТУДЕНТА НАУ ..	36
Кокот Д.Ю., Халімон Н.Ф. СПЕЦІАЛІЗОВАНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРИ ВІМ-ПРОЄКТУВАННІ .....	37
Приходько Р.В. ПРИНЦИПИ ВИКОРИСТАННЯ ЛІЦЕНЗІЙНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	39
Сябрук І.М., Яковенко Л.В. СУЧАСНИЙ СТАН СЕРВЕРНОЇ ПЛАТФОРМИ NODE.JS.....	40
Головач Ю.Ю. МЕТОДИ МУЛЬТІСЦЕНАРНОГО ПІДХОДУ У СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ ЛІКУВАННІ РАКУ ЛЕГЕНЬ .....	41
Юсенко А.С., Смокович М.В. МЕТОД ВИДАЛЕННЯ ШУМІВ З ЗОБРАЖЕННЯ .....	43

Дудинець С.В. FLUTTER ЯК ЗАСІБ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ .....	44
Швиндя А.М. ГЕЙМІФІКАЦІЯ ПРОЦЕСІВ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ .....	45
Росінська Г.П., Криворучко І.О. ОСОБЛИВОСТІ ЗМІНИ БІЗНЕС-ПРОЦЕСІВ ПІД ЧАС ВПРОВАДЖЕННЯ ERP-СИСТЕМ .....	46
Легкий С.Ю. ВИКОРИСТАННЯ REST АРІ СЕРВЕРА У ОБРОБЦІ ЗАМОВЛЕНЬ РЕСТОРАНУ .....	47
Глазок О.М., Станіщук К.А. ЗАСОБИ ОБРОБКИ ТА ВІЗУАЛІЗАЦІЇ ДАНИХ У МОВІ PYTHON .....	48
Нечипорук О.П., Брановицька І.В. РОЗРОБКА МОБІЛЬНИХ ДОДАТКІВ ДЛЯ ОПЕРАЦІЙНОЇ СИСТЕМИ IOS .....	49
Шмалюк Д.В., Граф М.С. АНАЛІЗ ECOMMERCE ПЛАТФОРМ ДЛЯ ПРОДУКТОВИХ КОМПАНІЙ ЯКІ ПРАЦЮЮТЬ У СЕГМЕНТІ B2B .....	51
Мартинов М.С. ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ІГРОВІЙ ІНДУСТРІІ.....	53

## **СЕКЦІЯ 1. РОБОТОТЕХНІКА, ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ, СИСТЕМИ ОБРОБКИ ЗОВНІШНІХ ДАНИХ**

УДК 519.872 (043.2)

*Коба О.В., д.ф.-м.н.*

*Інститут кібернетики ім. В.М.Глушкова НАНУ*

### **ОПТИЧНІ БУФЕРИ ТА МАТЕМАТИЧНІ МОДЕЛІ ЇХ ФУНКЦІОНУВАННЯ**

На сьогодні досить багато комп'ютерних мереж складаються з оптоволоконних ланок, що з'єднані між собою електричними вузлами. Передача даних в магістралі здійснюється у вигляді світла, зазвичай від світлодіоду або лазера. Технології DWDM забезпечують швидкість передачі даних значно вище 1 Тбит / с. Проте у вузлах це світло повинне бути перетворено в електричний домен, щоб передати всі дані в їх окремі пункти призначення. Із-за швидкого збільшення пропускної здатності каналів комутаційна здатність стає вузьким місцем системи. У теперішній час дослідницька діяльність зосереджена на оптичних комутаційних технологіях, які припускають меншу кількість перетворень з оптичної в електронну або зовсім без них. Проте і досі важливою проблемою в таких мережах є буферизація.

В телекомунікаціях оптичний буфер – це пристрій, який може тимчасово зберігати інформацію. Як і у випадку зі звичайним буфером, це носій, який дозволяє компенсувати різницю за часом виникнення подій. Більш конкретно, оптичний буфер слугує для збереження даних, які були передані оптично (тобто у форматі світла), без перетворення їх в електричну область.

У теперішній час дослідження оптичних буферів проводяться у двох окремих областях. Одна з них – вивчати технологічну реалізацію цього буфера і намагатися зменшити його розмір за допомогою повільних пристроїв. Друга – покращити загальну продуктивність за рахунок використання аналітичних методів.

Оскільки оптичний сигнал передається у вигляді світла, його не можна зберігати у класичних буферах (як, наприклад, в оперативній пам'яті), його можна лише затримати, збільшивши шлях проходження світлового променя у каналі зв'язку. Для цього можна скерувати його у багаточисельні кільця оптичного волокна, тобто затримати світло у циклічному оптоволоконному буфері [1].

Вперше модель з циклічним очікуванням, що описувала деякі аспекти посадки літаків, дослідив угорський математик Л.Лакатош [2]. Він дослідив систему масового обслуговування  $M/M/1$  з  $T$ - поверненням і



*Робототехніка, програмування мікроконтролерів, системи обробки зовнішніх даних*  
*FCFS* дисципліною обслуговування. Бельгійські математики розширили сферу застосування моделі Л.Лакатоша до моделі оптичного буфера з загальним часом обслуговування [3]. Ідею застосування своєї моделі відносно оптичних буферів Л.Лакатош розвинув в роботах [4, 5].

Авторка узагальнила модель функціонування оптичного буфера до системи масового обслуговування типу Лакатоша  $GI/G/1$  з  $T$ -поверненням (система типу  $GI/G/1$  з дисципліною обслуговування *FCFS* і  $T$ -поверненням), де  $T$  час перебування на циклі орбіти (за термінологією теорії СМО).

Для такої системи віднайдено умову існування ергодичного розподілу вкладеного ланцюга Маркова, ергодичну середню кількість заявок на орбіті та стаціонарне середнє числа циклів заявки на орбіті [6, 7].

#### Використані джерела

1. *Серебрякова С.В.* Застосування циклічних систем обслуговування // Доп. НАН України, 2016, №3 с.32-34.
2. *Lakatos L.* On a simple discrete cyclic-waiting queueing problem // J. Math. Sci. – 1998. – No 4(92). –P. 4031–4034.
3. *Rogiest W., Laevens K., Fiems D., Bruneel H.* Analysis of a Lakatos type queueing system with general service times // Proc. of ORBEL 20. Quantitative Methods for Decision Making, Ghent, Jan. 19–20, 2006. – P. 95–97.
4. *Lakatos L., Efrrosinin D.* A Discrete Waiting Time Model for Optical Signal. Distributed computer and communication networks. 17<sup>th</sup> international conference, DCCN 2013, Moscow, Russia, October 7-10, 2013, pp.114-123.
5. *Lakatos L., Szeidl L., Telek M.* Introduction to Queueing Systems with Telecommunication, Springer, Boston, 2013, 385 p.
6. *Koba E.V.* On a  $GI/G/1$  retrial queueing system with a FIFO queueing discipline // Theory of Stochastic Processes. – 2002. – 24, №8. – P.201-207.
7. *Коба О.В.* Стаціонарні характеристики системи масового обслуговування  $GI/G/1$  із  $T$ -поверненням при обслуговуванні у порядку черги // Вісник Національного авіаційного ун-ту. – 2003.- №1. –С.122-125.

**Крант Д.В.**

*Національний авіаційний університет*

## **МЕТОДИ ОРГАНІЗАЦІЇ ДОСТУПУ ДО CAN-ШИН АВТОМОБІЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ З ANDROID-ДОДАТКІВ**

CAN шина є послідовною шиною передачі даних з підтримкою одночасної роботи багатьох ведучих пристроїв. Відносно проста реалізація фізичного шару цієї шини зробила її дуже розповсюдженою у якості шини передачі даних між блоками рухомих об'єктів. CAN шина є обов'язковою для використання у автомобілях починаючи з 2004 року.

Структура команд, що передаються шиною, та її архітектура дозволяє отримувати повідомлення будь-якому блоку. Тобто шина працює за принципом broadcast, коли повідомлення розповсюджуються між усіма блоками, та кожен з них приймає індивідуальне рішення про обробку тієї чи іншої команди.

Для реалізації підключення будемо використовувати наступні компоненти: плата розробки з мікроконтролером AVR Atmega 328P, CAN-SPI конвертор MCP2515 та блютуз передавач HC-06. Пропоную під'єднатися до CAN шини за допомогою двох дротів CAN-H та CAN-L. Так як ми підключаємося у середину шини, немає необхідності використовувати термінуючий резистор 120 Ом.

CAN шину необхідно під'єднати до конвертора MCP2515, що дозволить отримати доступ до даних за допомогою інтерфейсу SPI. Після цього необхідно сконфігурувати модуль бездротової передачі даних HC-06 на дублювання вводу та вивода serial порту мікроконтролера. За допомогою отриманих в процесі реверс-інжинірингу таблиць із співвідношення значень CAN ID команди до автовиробника та інформації що в них міститься ми маємо можливість розпарсити отриманий потік даних та отримати необхідну нам інформацію.

Цей процес є досить складним з точки зору бізнес логіки та навантаження на обчислювальні ядра, та є можливість його спростити. У якості альтернативи пропонується використання CAN Bus декодерів. Це промислові рішення що мають можливість автоматично визначати модель і марку авто та перетворювати CAN команди у чітко визначену та уніфіковану структуру даних, яка надалі може бути використана на будь-якому рівні архітектури.

**Волнєйкін М.О.**

*Відокремлений структурний підрозділ «Фаховий коледж інженерії та управління Національного авіаційного університету»*

## **ІДЕНТИФІКАЦІЯ ОСОБИ ЧЕРЕЗ СИСТЕМУ ВІДЕОСПОСТЕРЕЖЕННЯ**

При плануванні системи відеоспостереження часто потрібно розпізнавати всіх відвідувачів, що входять до будівлі. Розпізнавання – це можливість визначити відомого оператора людини. Ідентифікація – отримання зображення, яке дає можливість однозначно дізнатися про незнайому людину. Це актуально як для об'єктів торгівлі, так і для офісних будівель, ТРЦ, шкіл, дитячих садків, житлових комплексів.

Також для ідентифікації особи необхідно мати базу даних з якою буде можливо порівняти особу. Це дасть можливість надалі використовувати отримані відомості при розслідуванні інцидентів, що відбулися, і зборі доказової бази. Щоб вирішувати завдання ідентифікації, потрібно вибрати критерії оцінки. Існує ціла низка рекомендацій – як вітчизняних, так і зарубіжних – яка щільність пікселів є достатньою для ідентифікації. Однак не можна обійтися лише критерієм дозволу картинки.

Крім того, вибираючи місце установки для системи спостереження ідентифікації, потрібно враховувати типову поведінку відвідувача, щоб кут нахилу осіб у кадрі не перевищував би  $\pm 10^{\circ}$ - $20^{\circ}$ . Наприклад, при виході та вході на ескалатор людина швидше за все дивитиметься собі під ноги, а при відкритті дверей – на ручку дверей.

Також треба враховувати, що при вході з вулиці людина може нести в руках парасольку. Для ідентифікації людини та перевірки обраного ракурсу встановлення камери, у програмі проектування відеоспостереження необхідно використати спеціальні тривимірні моделі людей. Це тривимірні моделі з похиленою вниз головою, людей у капюшоні та з розкритою парасолькою.

При виборі фокусної відстані об'єктива слід враховувати, що кут ясного зору людини по горизонталі становить приблизно  $36^{\circ}$ , що відповідає фокусній відстані приблизно 6,9 мм. Тому відеокамери з фокусною відстанню об'єктива менше ніж 6, 9 мм візуально віддалятимуть зображення, більше 6,9 мм – відповідно наблизатимуть.

**Горбатенков О.В.**

*Фаховий коледж інженерії та управління*

## **МЕТОД ЗАСТОСУВАННЯ ДАТЧИКІВ ДЛЯ ОПТИМІЗАЦІЇ ВИКОРИСТАННЯ РЕСУРСІВ**

На сьогодні в системах внутрішнього освітлення будівель знайшло широке застосування автоматичне керування з використанням датчиків. З метою економії електроенергії у Німеччині прийнято закон про обов'язкову установку в будинках спеціальних датчиків (присутності та руху). Для проектування або реконструкції будівлі необхідно реалізувати ці енергоощадні умови.

Принцип роботи датчиків базується на автоматичному вмиканні/вимиканні освітлення у приміщення залежно від інтенсивності потоку світла та/або присутності людей. Це можливо завдяки пасивній технології інфрачервоного випромінювання: вбудовані ІР датчики записують теплове випромінювання та перетворюють його на електричний сигнал, який вимірюється. Спектр випромінюваної людьми теплової енергії знаходиться в інфрачервоному діапазоні та не видим для людського ока

На оптичній лінзі збирається теплове випромінювання та проєктується на ІР датчики. Датчики реєструють зміни теплового випромінювання, викликані рухом, та перетворюються на електричний сигнал. Вбудована електроніка обробляє отриманий сигнал та виконує відповідно до сигналу заздалегідь визначену дію (ввімкнення/вимкнення освітлення).

Оптична система лінз фіксує теплову радіацію та проєктує дані на ІР датчик. Область виявлення датчика має активні та пасивні зони. На ІР датчик проєктуються лише активні зони. Внаслідок змін показань від однієї активної зони до іншої надсилається сигнал

Основною, та водночас, головною перевагою датчиків є просте встановлення та їх налаштування. Для подальшої роботи не потрібно прокладання спеціальних мереж, або застосування додаткового обладнання. Датчики встановлюються в розрив електричного ланцюга та одразу готові до використання. Використання даного обладнання дозволяє оптимізувати витрати ресурсів на 75%.

*Лазебний А.С.*  
*Національна академія СБУ*

## **МЕТОД РЕАЛІЗАЦІЇ СИСТЕМИ СПОСТЕРЕЖЕННЯ ЗА РОСЛИНАМИ У ТЕПЛИЦЯХ**

Ринок готових рішень для спостереження за рослинами у теплицях доволі обмежений і в основному пропонує професійні системи або унікальні проекти під замовлення (ціна подібних систем у базовій комплектації починається з 65 000 грн.). Аналіз компонентної бази систем та спрощення системи курування дозволив знизити витрати таку ж саме базову комплектацію до 10 000 грн.

В побудованій схемі було використано плату Arduino UNO та датчик вологості (рис. 1). Одне з спрощень було досягнуто за рахунок використання базових компонентів мови програмування C (рис. 2) для реалізації алгоритмів автоматичного поливу теплиці.

Даний проект направлено на спрощення системи моніторингу стану тепличних рослин та на автоматизацію системи поливу. Дані процеси передбачають віддалений контроль та регулювання основних параметрів в теплиці (температура, вологість повітря та ґрунту).

Дана система побудована за концепцією IoT та передбачає управління через телефон.

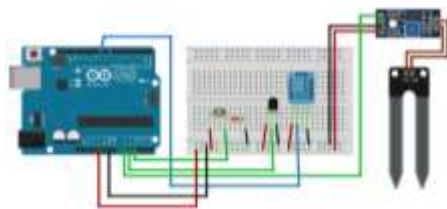


Рис. 1. Приклад схема підключення датчику вологості до плати керування

```
void setup() {
  // pin 13 as digital output with LED, as an example
  pinMode(13, OUTPUT);
  // pinMode(10) as analog input
  // pinMode(11) as analog input
  // pinMode(12) as analog input
  // pinMode(14) as analog input
}

void loop() {
  // read the sensor value
  float sensorValue = analogRead(A0);
  // convert the sensor value to voltage
  float voltage = sensorValue * (5.0 / 1023);
  // print the voltage to the serial monitor
  Serial.println(voltage);
  // turn the LED on (HIGH) if the voltage is greater than 1.0V
  if (voltage > 1.0) {
    digitalWrite(13, HIGH);
  } else {
    digitalWrite(13, LOW);
  }
  // wait for 100 milliseconds before the LED goes on
  delay(100);
  // wait for 100 milliseconds before the LED goes off
  delay(100);
}
```

Рис. 2. Частина коду реалізації системи

*Масловський Б.Г., к.т.н., доцент  
Національний авіаційний університет*

## **ПІДСИСТЕМА ЕНЕРГОЗБЕРЕЖЕННЯ У СИСТЕМІ «ІНТЕЛЕКТУАЛЬНИЙ ДІМ»**

даний час концепція «Системи інтелектуального керування будинком» набуває нового підходу в організації життєзабезпечення, при якому за рахунок комплексу програмно-апаратних засобів значно зростає ефективність функціонування й надійність керування.

Основною особливістю інтелектуального дому є об'єднання окремих підсистем різних виробників у єдиний керований комплекс. Інтегрована система керування інтелектуальним будинком займається рішенням завдань забезпечення комплексної роботи всіх інженерних систем домі: освітлення, опалення, вентиляції, водопостачання, контролю доступу й багатьох інших.

Так однією з важливих підсистем системи «інтелектуальний дім» є підсистема (кластер) енергозбереження, яке здійснюється крім установки ефективних опалювальних приладів по наступним напрямкам: розбивка зони впливу підсистеми на частини й повне автоматичне їх керування, а також ефективна система управління системою та її виконавчих пристроїв в цілому.

В основі ІД може лежати мережа на основі технології Ethernet. Передача даних у ній здійснюється за допомогою спеціальних протоколів: EI (European Installation Bus), Clipsal C-Bus, LonWorks, HART та ін.

У підсистемі, що розроблено така мережа реалізована на основі двох сучасних мікропроцесорних систем: Arduino для зчитування значень датчиків від різних приладів, та пристрою Raspberry Pi має MySQL сервер і WEB-сервер, що працює на ньому. Це два основних і важливих застосування домашньої автоматизації де через сервер MySQL, бази даних створюється інформація від Arduino Wi-Fi для управління через що користувач може підключитися до PHP-файлу і відкрити в браузері, на планшеті або ПК, а потім моніторити і управляти потрібними технічними засобами. Для цього передбачено використання програм PuTTY і Xming / Xlaunch які формують доступ до віртуального робочого столу Raspberry Pi.

Описане структурне і алгоритмічне рішення побудови однієї з головних підсистем системи «інтелектуальний дім» можна як

Робототехніка, програмування мікроконтролерів, системи обробки зовнішніх даних  
остову для проектування також інших підсистем, як у навчальному процесі (курсове та дипломне проектування), так і для розробок проектів для створення систем в цілому.

### **Використані джерела**

1. Роберт К. Элсенпитер, Тоби Дж. Велт. Умный Дом строим сами. 2006. – 256 с.
2. Mann William C. The state of the science//Smart technology for aging, disability and independence. – John Wiley and Sons, 7 July 2005. –ISBN 0-471-69694-3.
3. Home Automation: From the Basement to the Cloud – Scout Blog, 7 дек 2016. – 482 с. – ISBN 978-5-97060-315-4.

*Симонов Д.М.*

*Національна академія СБУ*

## **САМОРОБНИЙ ПОРТАТИВНИЙ ПОДАВЛЮВАЧ РАДІОСИГНАЛІВ**

Подавлювач сигналів – це прилад, який подавлює сигнали супутників. Подавлювач сигналів створює перешкоди на тих частотах, на яких він працює, роблячи об’єкт глушення “невидимим”. Спектр застосування даних пристроїв досить широкий: їх використовують спецслужби, водії/кур’єри і, звичайно, злочинці.

На наведеній схемі побудови саморобного подавлювача (рис. 3) було використано наступне обладнання: високовольтний підвищуючий модуль напруги (рис. 1), шість алюмінієвих кутків розміром 15 на 15 мм. і товщиною 2-3 мм. (рис.2), елемент живлення у вигляді двох батарейок АА та платформа на яку встановлюються кутки.

Шість алюмінієвих кутків необхідно розташувати як на схемі із мінімальним проміжком між ними. Таким чином вони формують високочастотну дугу, на яку подається напруга від модуля. Дроти модуля треба поєднати із першим та останнім кутком у ряду. Дроти живлення модуля поєднуються із елементом живлення. Елементом живлення може виступати акумулятор, або дві батарейки АА як на схемі (рис. 3).

Шляхом досліду, я визначив, що даний подавлювач сигналів спроможний пригнічувати сигнали GPS, Bluetooth, WiFi, радіосигнал, телевізійний сигнал і створювати перешкоди у стільниковому зв’язку. Радіус дії подавлювача – до 7 метрів. Якість подавлення сигналу залежить від відстані подавлювача до об’єкту глушення. Тобто, чим ближче подавлювач, тим сильніша перешкода.

Недоліками даного подавлювача сигналів є неможливість вибірково пригнічувати окремі види сигналів та відносно маленький радіус дії.

Основними перевагами є простота наведеної схеми та доступність приладу, оскільки його комплектуючі є у вільному доступі. Станом на 22.11.2021, собівартість приладу приблизно 146 грн.



*Давидова А.С.<sup>1</sup>, Кашкевич С.О.<sup>2</sup>*

*<sup>1</sup>Фаховий коледж інженерії та управління НАУ*

*<sup>2</sup>Національний авіаційний університет*

## **МЕТОД РЕАЛІЗАЦІЇ КВАНТОВИХ ОБЧИСЛЕНЬ В МОВІ ПРОГРАМУВАННЯ Q#**

При роботі програм з звичайними змінними по суті відбувається маніпулювання нулями і одиницями, які зберігаються в бітах, що поєднуються у байти. Цим нулям та одиницям відповідають дискретні бінарні стани фізичних систем. Але у випадку використання квантових комп'ютерів, які працюють із безперервними діапазонами станів, необхідно реалізовувати нові підходи до збереження та обробки значень змінних (які називаються «кубіти»).

Класичний біт може приймати тільки один із двох станів - "включено" або "вимкнено", як звичайна лампа розжарювання. Кубіти, основа квантових комп'ютерів, більше схожі на світильник із регулюванням яскравості.

Для роботи з кубітами можемо використати стандартні інструменти мови Q# та представити визначення подібної змінної наступним чином:

```
namespace Quantum.Bell
{
    ...
    operation Set (desired: Res_q, qub_1: Qubit) : ()
    {
        body
        {
            let curr_zn = M(qub_1);
            if (desired != curr_zn)
                X(qub_1);
        }
    }
}
```

Ця операція переводить наш кубіт у вибраний стан – 0 або 1. Спочатку вимірюється кубіт (ця операція позначається буквою M), і він колапсує в стан 0 або 1. Якщо виміряний стан відповідає бажаному, тоді змінюється його стан за допомогою вентиля NOT, X.

Для роботи з кубітами необхідно використовувати стандартні і вже прописані в мові інструменти: стан Белла, вентиль Адамара, тощо.

**Жванський А.І.**

*Фаховий коледж інженерії та управління*

## **АВТОМОБІЛЬНА КОМУНІКАЦІЙНА СИСТЕМА VEHICLE TO EVERYTHING (V2X)**

Vehicle to Everything (V2X) — це автомобільна комунікаційна система, яка підтримує передачу інформації від транспортного засобу до рухомих частин системи дорожнього руху, які можуть вплинути на транспортний засіб. Основною метою технології V2X є підвищення безпеки дорожнього руху, економії енергії та ефективності руху на дорогах.

У системі зв'язку V2X інформація передається від датчиків транспортних засобів та інших джерел через високо пропускісні надійні канали, що дозволяє їй спілкуватися з іншими автомобілями, інфраструктурою, такою як місця для паркування та світлофори, а також пішоходами, які використовують смартфони.

Завдяки обміну інформацією, наприклад про швидкість, з іншими об'єктами навколо автомобіля, технологія покращує усвідомлення водієм потенційних небезпек і допомагає зменшити тяжкість травм, смертельних випадків у ДТП та зіткнення з іншими транспортними засобами.

Технологія також підвищує ефективність руху, попереджаючи водіїв про майбутній трафік, пропонуючи альтернативні маршрути, щоб уникнути дорожні затори, і виявляє доступні місця для паркування.

Ключові компоненти технології V2X включають V2V (vehicle-to-vehicle) і V2I (vehicle-to-infrastructure). V2V дозволяє транспортним засобам спілкуватися з іншими транспортними засобами на дорозі, тоді як V2I дозволяє транспортним засобам спілкуватися із зовнішніми об'єктами, такими як світлофори, місця для паркування, велосипедисти та пішоходи. Технології допомагають підвищити безпеку дорожнього руху, зменшити споживання палива та покращити взаємодію між водіями та іншими учасниками дорожнього руху.

Коли системи V2X інтегровані в традиційні транспортні засоби, водії можуть отримувати важливу інформацію про погодні умови, найближчі аварії, дорожні умови, попередження про проведення дорожніх робіт, наближення до аварій та дії інших водіїв, які їздять на тій же дорозі.

Автономні транспортні засоби, оснащені системами V2X,

*Робототехніка, програмування мікроконтролерів, системи обробки зовнішніх даних*  
можуть надавати більше інформації наявній навігаційній системі транспортного засобу. Системи також дозволяють автономним транспортним засобам сканувати навколишнє середовище та приймати негайні рішення на основі отриманої інформації.

Хоча ринок V2X все ще знаходиться на ранніх стадіях, більшість виробників почали впроваджувати цю технологію, і транспортні засоби все більше спілкуються між собою та інфраструктурою навколо них. Транспортні засоби також стають розумними, і вони оснащені системами, які вимагають меншої участі людини. Завдяки адаптивному круїз-контролю та датчикам користувачі отримують переваги від безпечніших та екологічно чистих поїздок зі зниженими викидами вуглецю .

Однак для реалізації повних переваг систем V2X потрібен час, оскільки для того, щоб транспортний засіб зв'язувався з об'єктом він повинен бути обладнаний технологією V2X. Більшість об'єктів, таких як місця для паркування, світлофори та традиційні транспортні засоби, не мають систем V2X, а це означає, що вони не можуть спілкуватися з транспортними засобами, які вже використовують цю систему.

*Хлищборщ П.О.*  
*Національний авіаційний університет*

## **КОДУВАННЯ ПОВІДОМЛЕНЬ У RFID СИСТЕМАХ**

У каналі зв'язку RFID систем присутня цікава особливість – в ній відсутні фізичне з'єднання за допомогою провідників, а інфор-мація передається за допомогою магнітного поля, яке створене за допомогою вбудованих в пристрої котушок індуктивності. Так як в такій системі передачі інформації неможливо чітко відокремити послідовні високі або низькі рівні сигналу із-за особливостей тех-нології, доводиться використовувати в якості носія інформації саме зміну логічних сигналів.

Найпопулярнішим методом кодування, що базується на зміні логічних сигналів є манчестерський код. Сигнал, закодований манчестерським кодом, є самосинхронізуючим, тобто для передачі даних не потрібна додаткова лінія передачі тактових імпульсів, за рахунок того, що за час передачі одного біта даних, незалежно від того, це 1 або 0 забезпечується один перехід з одного логічного рівня на інший, що дозволяє приймачеві синхронізувати свій внутрішній тактовий генератор або таймер із тактовим генератором передавача.

Сигнал, закодований відповідно до манчестерського коду, не має постійної складової навіть у разі передачі довгих послідовностей з нулів і одиниць, тому у котушках індуктивностей RFID систем завжди буде змінний електричний струм, що і потрібно для передачі повідомлень за допомогою змінного магнітного поля.

Кодування манчестерського коду відносно просте - проводиться логічною операцією «виключне або» (додавання по модулю два, XOR) над поточним бітом даних, що кодується, і бітом тактового генератора. Для декодування спочатку проводиться бітова синхронізація - визначається середина періоду за часом перепаду сигналу двох періодів, проте синхронізації не достатньо, потрібно ще визначити початок інформаційного блоку, для цього в початку кожного блоку вставляється двійкове слово синхронізації.

Манчестерське кодування також використовуються і в інших технологіях передачі інформації, наприклад в популярних Ethernet (IEEE 802.3) та 10BASE-T. Даний спосіб кодування також можна використовувати для запису/зчитування даних на фізичних носіях, наприклад на магнітних стрічках.

*Мартінова О.П., к.т.н., доцент, Станко С.М.*  
*Національний авіаційний університет*

## **МЕТОДИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНО- РОЗВАЖАЛЬНИХ ПРОГРАМ ПРИ КЕРУВАННІ АВТОМОБІЛЕМ**

На сьогодні автомобільне обладнання вже може керувати програмами на смарт-телефонах, такими як вибір дзвінків, перегляд відео чи прослуховування музики. Перебуваючи в дорозі, навігатор надає звіт про стан дорожнього руху в режимі реального часу та інструкції та контролює стан транспортного засобу для управління ризиками. Автомобіль шукає найближчу (чи навіть найдешевшу) заправку, коли бензин або акумулятор закінчуються. На сьогодні впроваджені практично всі функції автомобіля, які раніше були лише у проектах, тепер час за навчанням розумних систем і налагодженням програмного забезпечення автомобілів.

Інформаційно-розважальна система в машині (IVI) – це термін з автомобільної промисловості, який відноситься до транспортних систем, що поєднують доставку інформації з доставкою розваг пасажиром та водієм. Для надання та управління такими послугами системи IVI обладнані інтерфейсом людини-машини (НМІ), що складається з аудіо / відео інтерфейсів, клавіатур, сенсорних екранів тощо.

Додаткові інструменти та функції, характерні для цих систем, включають відтворення аудіо / відео та двосторонні засоби зв'язку (CD / радіо, навігація, голосові команди, розваги на задніх сидіннях тощо). Крім того, зв'язок мобільних пристроїв в останні роки став головним компонентом систем IVI, намагаючись задовольнити зростаючі потреби у доступі до Інтернету та загального вмісту смартфонів, таких як потокове передавання музики, інформація про дорожній рух та прогнози погоди.

Система IVI складається з безлічі взаємопов'язаних апаратних та програмних компонентів, а загальну архітектуру або структуру довільної системи можна описати набором шарів, починаючи від апаратного рівня внизу і закінчуючи НМІ вгорі. Між кожним рівнем чітко визначені інтерфейси, і кожен шар включає ряд ключових технологічних блоків у програмному та / або апаратному забезпеченні для забезпечення бажаних функціональних можливостей системи

## СЕКЦІЯ 2. ПАРАЛЕЛЬНІ І РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ

УДК 004.5 (043.2)

*Ткаченко В.Г.<sup>1</sup>, к.т.н., доцент, Гришко Н.О.<sup>2</sup>*

*<sup>1</sup>Національний авіаційний університет*

*<sup>2</sup>Фаховий коледж інженерії та управління НАУ*

### **АНАЛІЗ ВИКОРИСТАННЯ REDIS В РОЗПОДІЛЕНІЙ СИСТЕМІ ОБРОБКИ ЗАМОВЛЕНЬ**

При реалізації розподіленої системи обробки замовлень в мережі ресторанів, яка налічує більше 20 закладів, а навантаження на мережні сервіси коливаються від 15 до 20 тисяч користувачів на день було використано систему Redis (від компанії “Redis Labs”). Redis організовує зберігання подій з використанням імутабельного механізму. Імутабельний механізм – це аналог журналу транзакцій, що підтримує тільки приєднання нових даних до вже наявних, але не зміну існуючих.

В реалізованій системі обробки замовлень використовується 9 розподілених сервісів, робота яких регламентується засобами Redis. Управління реалізоване на фронтенді у вигляді бібліотеки, яка робить запити за ключами відразу на потрібні сервери, які працюють тільки з локальними даними. Для роботи з базами даних було використано метод захищеного доступу.

При впровадженні оновленої системи на стадії тестування дане рішення показало підвищення швидкодії роботи системи під великим навантаженням (від 50 транзакцій на секунду), але час відповіді системи в окремих випадках її функціонування був більшим, ніж за попереднім рішенням.

Це може бути обумовлено збільшенням часу на відповідь сисеми, тому що додаткові ресурси витрачаються на маршрутування (marshalling) та дамаршрутування. Під маршалінгом розуміється процес перетворення об'єкта у спеціальний формат даних для зберігання або передачі. Дане дослідження направлене на аналіз роботи системи під час дослідної експлуатації під реальним навантаженням. Передбачається порівняти роботу налаштованої мережі з включеною і виключеною системою Redis. Основним показником будуть час відповіді системи та максимальний час виконання запиту.

Коли клієнт, товар або замовлення створюється/видаляється, подія повинна передаватися асинхронно службі CRM (Customer

relationship management) за допомогою RESP (Redis Serialization Protocol) для управління взаємодією із поточними та потенційними клієнтами. В реалізованій структурі службу CRM можна запускати та зупиняти під час виконання без будь-якого впливу на інші мікросервіси. Це передбачає, щоб усі повідомлення, надіслані CRM під час простою, були закріплені для обробки.

До основних переваг використання Redis, які обумовили його вибір, можна віднести:

1) можливість затримки до мілісекунд. Redis підтримує час відгуку в мілісекундах. Зберігаючи дані в пам'яті, може зчитувати дані швидше, ніж бази даних на основі дисків;

2) простота використання розробниками. Redis синтаксично простий у використанні та вимагає мінімальної кількості коду для інтеграції у програму;

3) розбиття даних між вузлами. Це дозволяє масштабувати систему та обробляти більше даних, коли кількість запитів зростає;

4) підтримка широкого набору мов програмування. Має багато клієнтів із відкритим кодом, доступних для розробників. Підтримувані мови включають Java, Python, PHP, C, C++, C#, JavaScript, Node.js, Ruby, Go та багато інших;

5) розширені структури даних. Redis підтримує рядки, списки, набори, відсортовані набори, хеші, бітові масиви та гіперлогічні журнали;

6) багатопоточна архітектура. Оскільки кеш багатопоточний, він може використовувати кілька ядер обробки. Це означає, що можна обробляти більше операцій, збільшуючи обчислювальну здатність;

7) можливість створення знімків системи. За допомогою Redis можна зберігати дані на диску з моментальним знімком, який можна використовувати для архівування або відновлення;

8) реплікація. Redis дозволяє створювати кілька копій основного Redis. Це дозволяє масштабувати читання бази даних і створювати високодоступні кластери;

9) геопросторова підтримка. Redis має спеціально створені команди для роботи з геопросторовими даними у реальному часі. Можна виконувати такі операції, як пошук відстані між двома елементами (наприклад, людьми чи місцями) та пошук усіх елементів на певній відстані від точки.

*Марченко А.А.*

*Міжнародний науково-навчальний центр інформаційних технологій та систем НАН України і МОН України*

## **ПРИНЦИПИ ПОБУДОВИ РОЗПОДІЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ РОБОТИ МЕРЕЖІ РЕСТОРАНІВ**

Тривалий період карантинних обмежень, викликаних пандемією коронавірусу, вплинув на діяльність і розвиток ресторанного бізнесу. Приміщення ресторанів було зачинено або обмежено на вхід для відвідувачів, що спричинило значні збитки. Дозвіл на приготування їжі та видачу або доставку її замовнику надав поштовх багатьом закладам громадського харчування змінити формат надавання послуг клієнтам.

Популяризація серед населення он-лайн замовлення продуктів харчування і готової їжі призвело до появи великої кількості сервісів доставки. Вартість відправлення клієнтам їжі сторонніми сервісами змушує власників великих мереж ресторанів замислитись щодо запуску власного сервісу доставки. Але такий серйозний крок не варто робити без комплексного аналізу продажів усіх ресторанів мережі.

Розроблено інформаційну систему (ІС) моніторингу роботи мережі ресторанів для збору й аналізу даних продажів залежно від формату замовлення. ІС має багаторівневу клієнт-серверну архітектуру. На головному сервері знаходиться інтегрована база даних, де зберігається інформація щодо усіх замовлень кожного закладу мережі. На допоміжному сервері розміщено програмне забезпечення для формування бази даних звітів, у яку з заданою періодичністю записуються зведені показники за кожним форматом обслуговування: у залі ресторану, самовивіз і доставка різними службами. Такий підхід дає змогу оперативно отримати статистичну інформацію, не навантажуючи основний сервер.

Клієнтську частину реалізовано як мобільний застосунок і як веб-застосунок. Застосунок отримує статистичну інформацію з бази даних звітів. Користувачу (керівнику мережі) надається порівняльна динаміка різних показників роботи ресторанів за вибраний період часу, що дає змогу оцінити ефективність управління мережею закладів і прийняти правильне рішення.



**Артамонов Є.Б., Голего Н.М.**  
*Національний авіаційний університет*

## **МІКРОСЕРВІСНА АРХІТЕКТУРА E-LEARNING СИСТЕМ**

Стрімке зростання попиту на електронні системи навчання (ЕСН) надало новий поштовх у напрямку розробки онлайн-ових ЕСН та навіть відновленню робіт над проектами, від яких розробники відмовились у зв'язку з відсутністю монетизації. Особливо дане зростання попиту стало відчутним з переходом на дистанційне навчання в університетах і школах всього світу під час карантину.

Досвід роботи з різноманітними готовими рішеннями онлайн-ових ЕСН показав у більшості випадків неможливість їх використання у сучасних реаліях, коли до системи одночасно можуть звертатись десятки, а іноді і сотні тисяч користувачів.

До основних проблем в роботі готових рішень онлайн-ових ЕСН можна віднести:

- довготривалі відповіді системи;
- помилки обробки коду у зв'язку з відсутністю процесорного часу на сервері;
- обмеження доступності бази даних (БД) у зв'язку з переповненням черги запитів.

У більшості випадків дані проблеми зазвичай вирішуються за рахунок експоненціального збільшення потужності апаратного забезпечення. Але з урахуванням того, що більшість онлайн-ових ЕСН побудовано за монолітною архітектурою, дане рішення дозволяє тільки відкласти проблеми в роботі системи на деякий час, а не вирішити їх.

На основі результатів, що було представлено в роботах та власних досліджень було запропоновано перехід на мікросервісну архітектуру побудови онлайн-ових ЕСН. Для цього було проведено аналіз основних показників роботи онлайн-ових ЕСН у порівнянні з розробленими рішеннями та представлено аналіз можливих недоліків даного рішення.

Під час розробки системи було визначено основні властивості мікросервісів, які необхідно забезпечити для функціонування всього комплексу онлайн-ової ЕСН:

- невеликий – один сервіс може розвивати одну команду не більше, ніж з 6 співробітників, одна команда може розвивати 6–10

сервісів, один сервіс може бути повністю переписаний однією командою за одну Agile-ітерацію;

- незалежний – втілення патернів High Cohesion і Low Coupling, коли кожен мікросервіс працює в своєму процесі та тому повинен явно позначити свій API;

- будується навколо бізнес-потреби та використовує обмежений контекст – зони відповідальності мікросервіса, сформульована навколо певної бізнес-потреби, чим вона компактніше, тим простіше створити новий мікросервіс;

- використання підходу Design for failure (страхування від збоїв) – одне з найбільш критичних місць в мікросервісній архітектурі – необхідність розробляти код для розподіленої системи, складові елементи якої взаємодіють через мережу. Мікросервіси повинні правильно обробляти різні варіанти відмови, вміти чекати, вміти повернутися до нормальної роботи при відновленні контрагента;

- обмежена централізація (багато баз, багато контекстів, що передбачає неможливість синхронної взаємодії кількох мікросервісів). Дана проблема вирішується відмовою від постійної узгодженості даних, де можливі помилки або обробляються за рахунок більш складної архітектури, або завдяки даним моніторингу;

- ітераційний розвиток (поступове впровадження мікросервісів з тестуванням їх взаємодії).

Для переходу на розподілену онлайн-овочу ECH на базі MSA мовою PHP з використанням СУБД MySQL було вирішено виділити наступні сервери для реалізації мікросервісів:

- сервер навчальних курсів (НК);
- сервер віртуальних лабораторій (ВЛ);
- сервер особистих кабінетів (ОК);
- сервер системи тестування знань (СТЗ);
- сервер оцінки роботи користувачів (ОРК);
- сервер аналітики;
- сервер центру обробки даних (ЦОД);
- сервери імпорту даних для організації доступу до сервісів.

*Кравець М.О., Халімон Н.Ф., к.т.н., доцент  
Національний авіаційний університет*

## **НЕРЕЛЯЦІЙНІ МОДЕЛІ БАЗ ДАНИХ ТА ВИМОГИ ACID**

NoSQL або постреляційні бази даних (БД) – це термін, що об'єднує в собі нереляційні сховища даних, які не підпорядковуються нормальним правилам зберігання даних, так званим вимогам ACID. Вони отримали назву NoSQL (Not Only SQL), як сховища нереляційного типу. Зазвичай такі системи не мають жорсткої структури, не використовують об'єднання таблиць (операція join), у яких є надмірність даних, а самі дані зберігаються в структурах, відмінних від таблиць.

Виконання вимог ACID гарантує достовірність даних у разі помилок та збоїв:

- неподільність (Atomicity). Транзакція або виконується повністю або не виконується.

- узгодженість (Consistency). Транзакція переводить базу даних із одного узгодженого стану до іншого.

- ізолюваність (Isolation). Результати транзакції стають доступними інших транзакцій лише після її фіксації.

- тривалість (Durability). Після фіксації транзакції зміни стають постійними.

Традиційні транзакційні бази даних мають класичні властивості атомарності, узгодженості, ізолюваності, довговічності, тобто "класичні" SQL БД задовольняють характеристики ACID. Про NoSQL бази говорять, що підтримки ACID у них немає. Оригінальне визначення ACID відноситься до операцій із базою даних, тобто до транзакцій. Під "операцією" у визначеннях ACID слід розуміти безліч дрібних операцій CRUD (Create, Read, Update, Delete) над даними. Операції, що задовольняють вимоги ACID, називають транзакціями (логічна одиниця роботи з даними).

У NoSQL БД зазвичай "великі транзакції" не підтримуються (з використанням операції join), але окремі операції модифікації даних можуть задовольняти деяким властивостям ACID. Окремим винятком є NewSQL та графові бази даних. Часто вони декларують підтримку транзакцій, які повністю аналогічні таким у SQL базах.

*Артамонова К.Є., Яковенко Л.В.*  
*Національний авіаційний університет*

## **ОСОБЛИВОСТІ ВИКОРИСТАННЯ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ**

В розробленій системі обробки замовлень користувачів було використано мікросервісну архітектуру, яка передбачала розподіл системи на 10 окремих мікросервісів.

Дослідження проводилось на двох варіантах виділених серверів:

- 1) Intel Xeon E3-1230 v5/6, з частотою 3,2 ГГц, 6 ядрами та 6 ГБ.
- 2) Intel Xeon 2x E5-2690 v2, з частотою 3,0 ГГц, можливістю багатопоточності – 40 потоків, 20 ядрами та 256 ГБ.

Слід відзначити, що реалізація MSA на сервері за варіантом 1 ускладнена малою кількістю ядер процесорів, тому що дана архітектура передбачає розгортання 10 віртуальних серверів. Цю проблему було вирішено за рахунок розподілу потоків користувачів за сервісами, тому середній час відповіді сторінок був у межах допустимого. При оцінюванні показників середнього часу доступу до системи, що побудована за MSA, використано середній час доступу до всіх мікросервісів без урахування «простоювання» того або іншого мікросервісу в одиницю часу.

Отримані дані показали:

- для навантажень до 1000 користувачів відсутня потреба у переході на MSA;
- при очікуваному навантаженні від 10000 користувачів монолітна архітектура використовує ресурси достатньо потужного серверу, але все одно не може забезпечити безпомилкову роботу системи. Найчастіше в системі викликалась помилка перевищення часу очікування відповіді);
- MSA за рахунок можливого використання балансування навантаження на рівнях L4 та L7 може впоратись і з більшою, ніж 50000, кількістю користувачів. Це відбувається за рахунок розподіленої обробки запитів і кешування даних окремих сервісів.

### **СЕКЦІЯ 3. МУЛЬТИПЛАТФОРМЕННІ ПРОГРАМНІ ЗАСОБИ, МОБІЛЬНІ ТЕХНОЛОГІЇ**

УДК 004.624 (043.2)

*Бесараб І.В.  
Innit. Inc*

#### **АКТУАЛЬНІ ТЕХНОЛОГІЇ У РОЗРОБЦІ МОБІЛЬНОГО ПЗ**

В розробці додатків для сучасних мобільних пристроїв ми постійно стикаємося з труднощами які призводять до негативних наслідків для продукту. Ці труднощі можуть бути абсолютно різними по суті свого походження тож для їх вирішення застосовуються різні підходи та інструменти що полегшують роботу для розробників ПЗ.

Аналіз даних та вміле користування сучасних інструментів та сервісів може значно прискорити розробку сучасних мобільних додатків, а також заощадити час на встановлення причини виникнення помилки та її місця в програмному коді.

На доповіді розглянуто чотири різні випадки вирішення проблем пов'язаних з розробкою мобільних застосунків для системи iOS, а саме:

- підгонка контенту на пристроях з малим екраном;
- відстеження стабільності додатків;
- зворотна сумісність;
- сторонні бібліотеки.

В результаті дослідження кожного з випадків був сформований алгоритм рішення типових задач. А також встановлені та розібрані позитивні та негативні сторони деяких сторонніх бібліотек та фреймворків при додаванні них до загального коду проекту.

Як висновок можна сказати, що кожний програмний продукт є унікальним тож, набутий досвід має допомогти підібрати технологію котра буде задовольняти вимоги поставлені менеджером проекту та досягти поставлених цілей за відносно короткий проміжок часу.

*Бондаренко Б.В., Халімон Н.Ф., к.т.н., доцент  
Національний авіаційний університет*

## **ВЕБ-ДОДАТОК ДЛЯ СЛУЖБ КРОВІ НА ПЛАТФОРМІ SPRING**

Вже сьогодні забезпечення постійної наявності компонентів крові є серйозною проблемою охорони здоров'я незалежно від країни. Не менш важливими завданнями, які стоять перед медичними організаціями, є підвищення безпеки реципієнтів і донорів, а також зниження фінансових витрат на збір, обробку та доставку донорського продукту. Тільки впровадження нових підходів в управлінні запасами крові, грамотна логістика, раціональне використання ресурсів служби переливання крові та високотехнологічні інтелектуальні рішення для ідентифікації, відстеження та управління чутливими продуктами здоров'я (еритроцитами, плазмою, тромбоцитами) можуть стати вирішенням зростаючої проблеми.

Розроблений веб-додаток реалізує ведення реєстру донорів та реципієнтів. Було створено модулі, які надають можливість: авторизації та аутентифікації через JWT (JSON Web Tokens); реєстрації нового користувача; створення заявок про донорів та реципієнтів; збереження даних про донорів та реципієнтів; вирішення асинхронних задач (видалення прострочених користувачів з бази даних, видалення прострочених заявок).

Веб-додаток розроблено на базі модульного фреймворку Spring з підтримкою інверсії управління (впровадження залежностей) для платформи Java. Використано наступні засоби. Spring Boot – фреймворк для створення та запуску додатків; Spring Security – середовище аутентифікації, авторизації та контролю доступу, яке використовується для захисту програм на основі Spring; технологію Java Persistence API для організації зв'язку з базою даних; систему управління збірками Maven; систему управління базами даних Postgres; Hibernate – засіб для організації взаємодії між об'єктами та реляційними структурами (object-relational mapping, ORM) для платформи Java.

**Спілок В.Ю.**

*Фаховий коледж інженерії та управління НАУ*

## **МОДИФІКАЦІЯ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ АВТОМОБІЛЬНОГО ПОРТАЛУ**

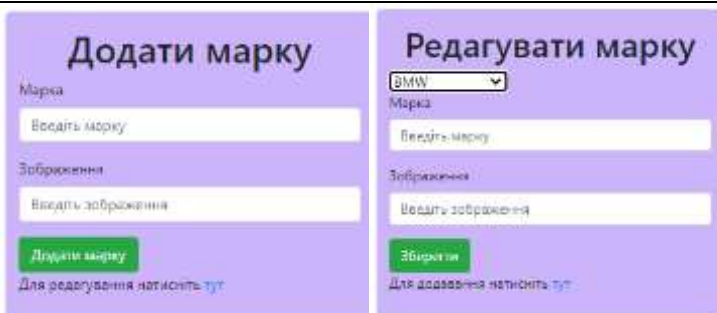
Логіка традиційних CMS поєднує в собі бекенд- та фронтенд-частини однієї системи. Контент виявляється пов'язаним з конкретними технологіями, архітектурою і шаблонами клієнт-серверного додатку.

Headless CMS – це принципово новий підхід до реалізації системи управління контентом. Логіка Headless CMS така, що до «тіла» за потреби можна приставляти різні «голови». Це дозволяє використовувати один бекенд для керування сайтом (або сайтами) та мобільним додатком, а також автоматизувати розповсюдження контенту по всіх доступних майданчиках та пристроях.

Розроблена модифікація передбачає управління лише контентом незалежно від інтерфейсу сайту, у якому він використовуватиметься (представлятися кінцевому користувачеві). Система управління будується з нуля і використовується в першу чергу як сховище контенту та набору інструментів. Вона забезпечує адміністративний інтерфейс для розробників контенту, їхньої спільної роботи над вмістом. Якщо передбачена можливість залишати коментарі, заявки, створювати анкети або задавати налаштування облікових записів, ці дані також можуть зберігатися в системі, модеруватися і редагуватися персоналом. Введення, редагування та зберігання даних зображено на рис.

Для користувачів адміністративної панелі важлива зручність роботи в системі. Централізоване управління полегшує взаємодію з різними платформами. Можна додавати та редагувати контент, керувати налаштуваннями в одному звичному адміністративному інтерфейсі. Статично згенерований контент CMS легко піддається масштабуванню через CDN.

Розроблена модифікація CMS дозволила мінімізувати витрачені ресурси при створенні автомобільного порталу, а керування різними платформами здійснюється централізовано з одного інтерфейсу. Це дозволяє гнучко налаштувати контент для кожного окремого каналу.



+ Параметри

				id	name	image
<input type="checkbox"/>				1	Lexus	lexus.png
<input type="checkbox"/>				4	Audi	audi.png
<input type="checkbox"/>				5	Subaru	subaru.png
<input type="checkbox"/>				6	BMW	bmw.png
<input type="checkbox"/>				7	Lada	lada.png
<input type="checkbox"/>				8	Jeep	jeep.png

Рис. Приклад введення та зберігання даних.

Але недоліком підходу Headless для розробки CMS є те, що він забезпечує лише бекенд, тому фронтенд-архітектуру необхідно реалізовувати окремо самому або за допомогою додаткових ресурсів.



Стенякін І.А.

Національний авіаційний університет

## ПОВЕДІНКОВІ ПАТТЕРНИ АУТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ В ОНЛАЙНОВИХ ЕЛЕКТРОННИХ СИСТЕМАХ НАВЧАННЯ

У системах електронного навчання виникає необхідність зберігання все більшої кількості різноманітної інформації, що стосується конкретного користувача: графік занять, завдання, відповіді, оцінювання робіт і т.д.[1] З переходом навчання у віддалений формат, необхідність аутентифікації стала актуальною не лише для користувача, а й для контролюючих органів освіти – щоб уникнути випадків шахрайства, у т.ч. спроб виконання роботи іншою людиною.

Поведінкова біометрична технологія – це високоточна технологія аутентифікації, яка може ідентифікувати користувачів на основі їх моделей поведінки. Вона ідентифікує унікальні індивідуальні характеристики того, як користувачі взаємодіють зі своїм мобільним пристроєм або комп'ютером замість інших поширених технологій, які ідентифікують користувачів на основі їх фізичних ознак. Цифрові ідентифікатори, засновані на поведінковій біометрії, також унікальні для людини, як відбитки пальців. Вони можуть швидко та точно перевіряти особу користувача від одного сеансу до іншого та постійно перевіряти особу протягом одного сеансу.

Таким чином, система повинна звертати увагу на характер взаємодії користувача з пристроєм. Розглядаються такі підвиди взаємодії: здійснення зв'язків, активність у мережі, переміщення протягом доби, періодичність набору тексту.

Незвична поведінка користувача за будь-яким із параметрів фіксується і зв'язуються показники з інших видів активностей[2]. Сформувавши ймовірності злому по кожному з них, можна розрахувати підсумкову ймовірність таким чином:

$$H = \operatorname{argmax}_{i \in \{0,1\}} \prod_{x_t \in \Omega} P(x_t | H_i),$$

$$\text{де } \Omega = \{x_t | T_{\text{current}} - T(x_t) \leq \omega\},$$

де  $\omega$  – замір часу події в секундах,  $T(x_t)$  – це позначка часу події  $x_t$ , а  $T_{\text{current}}$  – поточна позначка часу[3].

Отже, аналізуючи різноманітні паттерни поведінки користувача, є можливість визначити характер взаємодії юзерів з

пристроями.

Виходячи з таких міркувань, бажано використовувати сукупну вірогідність втручання для прийняття рішення про розлогування користувача в в онлайнних електронних системах навчання. Зафіксувавши шаблони поведінки користувача та поєднавши їх з його профілем, можна забезпечити неперевну(активну) аутентифікацію у цільовій системі.

### **Використані джерела**

1. Артамонов Є. Б. Порівняльний аналіз підходів до визначення зацікавленості користувача навчальними матеріалами в адаптивних навчальних системах / Є. Б. Артамонов // Наукоємні технології. - 2016. - № 4. - С. 383-388. - Режим доступу: [http://nbuv.gov.ua/UJRN/Nt\\_2016\\_4\\_5](http://nbuv.gov.ua/UJRN/Nt_2016_4_5).

2. Артамонов Є. Б. Електронні сховища даних із захищеним доступом / Є. Б. Артамонов, О. О. Беляков // Наукоємні технології. - 2013. - № 4. - С. 402-405. - Режим доступу: [http://nbuv.gov.ua/UJRN/Nt\\_2013\\_4\\_10](http://nbuv.gov.ua/UJRN/Nt_2013_4_10).

3. [https://www.researchgate.net/publication/225237361\\_Implicit\\_Authentication\\_through\\_Learning\\_User\\_Behavior](https://www.researchgate.net/publication/225237361_Implicit_Authentication_through_Learning_User_Behavior)

*Халімон Н.Ф., к.т.н., доцент  
Національний авіаційний університет*

## **ВЕРТИКАЛЬНЕ ТА ГОРИЗОНТАЛЬНЕ МАСШТАБУВАННЯ БАЗ ДАНИХ**

Масштабування (database scalability) – здатність системи реагувати на збільшення навантаження шляхом нарощування її ресурсів. Існує два основні підходи до масштабування:

1). Вертикальне масштабування (scale-up) – збільшення продуктивності системи виконується за допомогою більш потужного обладнання;

2). Горизонтальне масштабування (scale-out) – необхідний рівень продуктивності системи досягається за допомогою розбиття системи на структурні компоненти та розподілення їх за окремими фізичними комп'ютерами мережі.

Класичним способом масштабування реляційних баз даних є вертикальне масштабування. При необхідності обробляти та зберігати величезні обсяги даних доводиться розширювати обчислювальну потужність комп'ютера, додавати пам'ять, переходити на більш об'ємні та швидкодіючі диски, застосовувати методи, які підвищують швидкість виконання запитів [1]. Звичайно, такий підхід має свої межі і є досить дорогим. Вертикальне масштабування передбачає нарощування потужностей сервера.

Альтернативою вертикальному масштабуванню є горизонтальне. Його основна ідея – це можливість додавати додаткові обчислювальні вузли у так звані кластери, агрегована потужність та обсяг яких дозволяють зберігати та обробляти значні обсяги даних [2]. У контексті баз даних цей підхід полягає у розподілі даних між кількома базами даних, розміщеними на різних вузлах кластера. Але для підтримки горизонтального масштабування необхідно було відмовитися від критично важливих особливостей реляційних СУБД:

- виконання транзакцій;
- операції JOIN;
- незмінної моделі даних (Static Schema).

Відмова від цих можливостей та підтримка горизонтального масштабування призвели до використання нереляційних моделей,

нових способів зберігання даних та, як наслідок, до організації нових типів сховищ [3]. До основних способів горизонтального масштабування баз даних відносять: партиціонування (partitioning) або секціонування, реплікацію (replication) та шардинг (sharding).

На разі більшість NoSQL СУБД (MongoDB, Redis, Cassandra) включають шардинг як основну з ключових можливостей за замовчуванням. Деякі з них підтримують автоматичне розбиття (partitioning methods, партицирование) і балансування навантаження на вузли кластера. На даний час партиціонування також реалізовано у багатьох реляційних СУБД, серед таких – Oracle, MySQL, Microsoft SQL Server, PostgreSQL, IBM DB2, Informix, Teradata, Adaptive Server Enterprise.

### **Використані джерела:**

1. Халімон, Н.Ф. Статистическая динамика процесса запросов с переключениями в сетях хранения данных / Н.Ф. Халімон // Державний університет інформаційно-комунікаційних технологій «Вісник ДУІКТ». – 2013. – Т.10, №4. □ С.68.

2. Халімон, Н.Ф. Використання кластерів великих даних / Н.Ф.Халімон, П.О.Купріянов // «Сучасні тенденції розвитку системного програмування»: науково-практична конференція, 25-26 жовтня 2019 р.: тези доп. / □К., 2019. □ С.22.

3. Халімон Н.Ф. Системні засоби підтримки темпоральних баз даних / Н.Ф.Халімон, В. С. Власенко // «Сучасні тенденції розвитку системного програмування»: науково-практична конференція, 21-22 листопада 2017 р.: тези доп. / □К., 2017. □ С.28.

***Остапенко В.О.***

*Київський національний університет ім. Т.Г.Шевченка*

## **ОСОБЛИВОСТІ 3D-ВІЗУАЛІЗАЦІЇ В НАВЧАЛЬНИХ СИСТЕМАХ**

Підвищення кваліфікації операторів складних промислових технічних систем на реальних установках і в реальних умовах на даний момент в більшості випадків небезпечно і дороге, а іноді взагалі не може бути реалізованим. Альтернативою є комп'ютерні імітаційно-тренажерні комплекси, які за рахунок використання віртуальних об'єктів в максимально можливій мірі моделюють реальні установки і дозволяють під час підвищення кваліфікації отримувати правильні і стійкі навички.

В дистанційній освіті більшого розповсюдження набули системи 3D-візуалізації з елементами інтерактивності. Інтерактивність передбачає як можливість нелінійного вивчення матеріалу (за власним сценарієм учня), так і використання стандартних сценаріїв, які закладені в систему розробниками.

Але для впровадження віртуальних лабораторій необхідний комплексний підхід, а не реалізація тренажерів у окремих галузях. На даний момент проблематика побудови віртуальних 3D-тренажерів практично не розглянута у роботах вітчизняних та закордонних вчених, і у більшості випадків носить формат описання окремих рішень, а не загальної концепції реалізації та впровадження елементів 3D-візуалізації до освітнього процесу.

Віртуальні тренажери дозволяють реалізувати основні принципи адаптивних навчальних систем, в яких використовується побудова динамічних сценаріїв. Так формування адаптивної навчальної системи можна будувати за функціональністю подібно комп'ютерним іграм з використанням нелінійних сюжетів, різноманітних рішень одних і тих самих задач, незакріпленого порядку розв'язків, багатоваріантних способів проходження курсу.

## СЕКЦІЯ 4. СПЕЦІАЛІЗОВАНІ ПРОГРАМНІ ЗАСОБИ

УДК 004.624 (043.2)

*Жарова О.В., к.пед.н.*

*Національний авіаційний університет*

### МОДИФІКАЦІЯ СИСТЕМИ ФОРМУВАННЯ ІНДИВІДУАЛЬНОЇ ОСВІТНЬОЇ ТРАЄКТОРІЇ СТУДЕНТА НАУ

В НАУ вже більше року працює система формування індивідуальної освітньої траєкторії студента. За час роботи система зарекомендувала себе, як зручний засіб з мінімалістичним інтерфейсом для опитування студентів щодо їх індивідуальної освітньої траєкторії.

Але особливість опитування студентів на майбутній рік викликала іноді незрозуміння у опитуваних, тому що для них відображалась група з номером, якою вона буде наступного року, а не поточного (особливо це проявилось при опитуванні бакалаврів).

Тому було вирішено внести зміни в логіку виводу назви груп при авторизації.

Тепер- буде відображатись той номер групи, який є на момент голосування (виводиться для студентів `group_infos.name` за правилом:

1) if (`educational_degree=1`), тоді в умові на рік прописати `year+1`;

2) if (`educational_degree=2 or educational_degree=3`), тоді рік незмінний.

Для цього необхідно відстежувати дату голосування та її попадання в період: для кожного навчального року.

Але дане коригування змінило логіку вибору дисциплін дл відповідного курсу:

1) дисципліни показувати для `educational_degree=1` брати зв'язку за `course +1` і `year+1`,

2) для решти `educational_degree` вказані `course` і `year`

Приклади виконання з точки зору адміністрування:

1) для бакалаврів:

- вибрали курс 1, бакалавр, вибрали поточний рік 2021 року, написали номер групи - КС-101;

- додали групу на наступний рік (2022, 2 курс, КС-201),

- при авторизації показується група поточна КС-101, 1-й курс, а дисципліни на 2022 рік.

2) для магістрів і післядипломної освіти:

- вибрали курс 1, МАГІСТР, вибрали поточний рік 2022, написали номер групи - КС-101М,

- при авторизації показується група поточна НА ту дату, яка потрапила в період навчального року КС-101М, 1й курс, та дисципліни на 2022 рік.

*Кокот Д.Ю., Халімон Н.Ф., к.т.н., доцент  
Національний авіаційний університет*

## **СПЕЦІАЛІЗОВАНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРИ ВІМ-ПРОЄКТУВАННІ**

Технологія ВІМ (Building Information Modeling) виникла порівняно недавно, але за останні роки активно стає домінуючою у світовій проектно-будівельній практиці, замінюючи методи проектування, які застосовувалися раніше. ВІМ – технологія інформаційного моделювання об'єктів, є розвитком досить популярної сьогодні системи автоматизованого проектування (САПР). Основною відмінністю від останньої, окрім тривимірного креслення, є наявність у моделі бази даних, що містить детальну інформацію про технологічні, технічні, архітектурні, інженерні та економічні характеристики об'єкта із можливістю одночасного створення та використання бази кількох проектувальниками.

Головною перевагою ВІМ-проектування є скорочення термінів проектування шляхом об'єднання всієї проектною інформації та, за необхідності, її трансформації без втрати даних. На ринку в даний час є програми, що підтримують концепцію ВІМ. Серед них можна виділити такі: Autodesk Revit, Autodesk Civil 3D, Graphisoft ArchiCAD, Nemetschek Allplan, SketchUp. Найбільш потужною у реалізації концепції ВІМ-проектування є програми ArchiCAD та програми сімейства Autodesk.

Кожна з перерахованих програм має свої переваги та слабкі сторони. ArchiCAD, як інструмент ВІМ-проектування, орієнтована на роботу в архітектурному бюро, але занадто трудомістка для виконання робіт з інших напрямків. Програмний продукт Autodesk Revit дозволяє користувачеві виконувати роботи з архітектурного, конструктивного розділів, а також проектувати внутрішні інженерні мережі. Програма Autodesk Civil 3D дозволяє займатися проектуванням генпланів та зовнішніх інженерних комунікацій. Поздовжні профілі генеруються автоматично відповідно до нормативних документів. Однак ці продукти мають більш слабкий порівняно з ArchiCAD інструментарій для архітектурного проектування, у них відсутні деякі важливі функції, необхідні при проектуванні, як наприклад, автоматичний підбір кольорів.

Більшість наявних на ринку програм проектування, що

реалізують BIM-проектування не мають вбудованої можливості обирати кольори промислових стандартів для створених об'єктів, а наявні реалізації (наприклад, в AutoCAD) мають обмежені можливості підбору.

Робота архітектора-проектувальника часто є пов'язаною з виїздами на реальний об'єкт та отриманням цифрових даних. Було створено мобільний додаток, який надає можливість автоматичного підбору кольорів згідно стандартів. Розроблене спеціалізоване програмне забезпечення реалізує алгоритм підбору кольору, який засновано на суб'єктивному хроматичному сприйнятті кольору людиною, обираючи із бази кольорів RAL найбільш подібний колір. Додаток розроблено на базі .NET Framework 4 та Xamarin на мові програмування C# у середовищі розробки Visual Studio.



**Приходько Р.В.**

*Навчально-науковий інститут інформаційної безпеки та стратегічних комунікацій*

## **ПРИНЦИПИ ВИКОРИСТАННЯ ЛІЦЕНЗІЙНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Використання ліцензійного програмного забезпечення - означає дотримання законодавства, відповідність стандартам і вимогам при сертифікації, повагу праці розробників і прав інтелектуальної власності, повага до інтересів партнерів і замовників, а в кінцевому підсумку - самоповага.

Легалізація програмного забезпечення спрямована на встановлення ефективної системи правової охорони програмного забезпечення як об'єкта авторського права та відповідає курсу країни на європейську інтеграцію.

Щоб зрозуміти, для чого потрібно використовувати ліцензійне програмне забезпечення, користувач повинен знати всі його переваги, а саме такі як:

1. Юридичний захист.
2. Технічна підтримка виробника програмного забезпечення.
3. Гарантія та повна впевненість в працездатності ліцензійної програми.
4. Постійні оновлення і нові версії програм.
5. Захист вашого комп'ютера від вірусів.

Використовування неліцензійного програмного забезпечення може понести за собою кражу даних користувача або можливість зараження комп'ютерними вірусами. У сучасному світі проблема втрат від дій зловмисного програмного забезпечення набирає обороти та використовується майже всюди.

Визначення наявності ліцензій на програмне забезпечення, встановлене на підприємстві, здійснюється на підставі надання компанією документів, які можуть це підтвердити. До них відносяться бухгалтерські документи, що підтверджують придбання ПЗ, підписана ліцензійна угода з правовласником, голографічні наклейки, ліцензійні диски, з яких здійснювалося встановлення.

По суті, ліцензія виступає гарантією того, що користувач ПЗ зможе без багатьох зусиль користуватися програмами і не боятися покарань від суду, та зможе безпечно користуватися нею.

*Сябрук І.М., Яковенко Л.В.*  
*Національний авіаційний університет*

## **СУЧАСНИЙ СТАН СЕРВЕРНОЇ ПЛАТФОРМИ NODE.JS**

Node JS – це середовище виконання JavaScript, побудоване на базі JS-движка V8, яке розроблено в Google та застосовуване в Google Chrome.

Дослідження системи питань та відповідей про програмування Stack Overflow показує, що Node.js - це, на сьогодні, найпопулярніша серед розробників технологія. Node.js і AngularJS залишаться найчастіше використовуваними технологіями.

П'ятий рік поспіль JavaScript є найпоширенішою мовою програмування в порівнянні з іншими мовами програмування.

Node.js не тільки здійснив революцію в серверній розробці, але, завдяки йому, зроблено внесок і в продуктивність клієнтських додатків. Крім того, він відіграє помітну роль у розширенні всієї екосистеми JavaScript та у вдосконаленні сучасних JS-фреймворків, таких як Angular, React.js або Vue.js.

Для налагодження серверних JavaScript - додатків можна використовувати ті ж методи, які застосовуються і для налагодження клієнтського коду, застосовуючи node-inspector, де зібрані засоби інструментів розробника Chrome.

На базі Node.js можна створювати корпоративні системи. Незважаючи на те, що в ньому є не дуже багато вбудованих засобів, що спрощують створення подібних систем, відомі компанії на IT-ринку використовують Node як корпоративну веб-платформу.

Використовуючи JS на клієнті та на сервері, можна створювати веб-програми, які можуть рендеруватись як в браузері, та і на сервері в реальному часі. Прикладами таких програм можуть бути програми для спільної роботи, інтерактивні чати, системи миттєвого обміну повідомленнями та онлайн-ігри.

Використовуючи Node.js можна створювати односторінкові програми. Більшість навантаження під час роботи таких додатків лягає на клієнтську частину системи, написану JavaScript.

Сервери на Node мають ще одну властивість – масштабування. Серверну частину програм можна масштабувати залежно від потреб проекту використовуючи безліч невеликих розподілених обчислювальних вузлів, які можуть обмінюватися даними.

**Головач Ю.Ю.**  
Компанія «Squad»

## **МЕТОДИ МУЛЬТИСЦЕНАРНОГО ПІДХОДУ У СИСТЕМІ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ ЛІКУВАННІ РАКУ ЛЕГЕНЬ**

За умови введення карантину, у зв'язку з пандемією, практично всі сфери діяльності людини зазнали змін, а саме перенесення більшої частини діяльності у онлайн формат (навіть консультації лікарів за можливості проводяться в режимі онлайн-конференцій). Даний перехід призвів до стрімкого розвитку різноманітних систем автоматизації, які направлені на зменшення безпосереднього контакту замовника та виконавця послуги.

Саме в цей момент розробники автоматизованих систем зіштовхнулись з проблемою адаптації інтерфейсів для осіб, які не мають жодного досвіду в роботі з програмними системами. До цього моменту в більшості випадків автоматизованими системами користувались безпосередньо зацікавлені в цих системах користувачі та/або фахівці в галузі, діяльність якої було автоматизовано. До вимог користувачів у вподобаннях, поглядах, цілях використання автоматизованих систем додалися задачі врахування різного рівня технічної підготовки та різноманітність технічного оснащення.

Запропонований підхід мультисценарного інтерфейсу для кожної ролі користувачів був використаний для побудови системи підтримки прийняття рішень при лікуванні раку легенів. У дослідженні прийняли участь 500 учасників протягом 30 днів, контролювався тип обраного сценарію відображення даних на початку та в кінці дня. У середньому користувачі працювали з системою 25–30 хвилин на день.

У системі можна відзначити поступовий перехід до сценаріїв вищої складності, які передбачають аналіз всієї доступної інформації. Дана тенденція зменшується з часом і за місяць приходиться до сталого стану (рис. 1).

У підсумку після стрімких і стрибкоподібних змін у 10 перших днів, після введення нового механізму налаштування інтерфейсу, було відмічено повільну змін кількості користувачів на кожному зі сценаріїв (рис. 2).



Рис. 1. Графіки кількості змінених сценаріїв за днями роботи

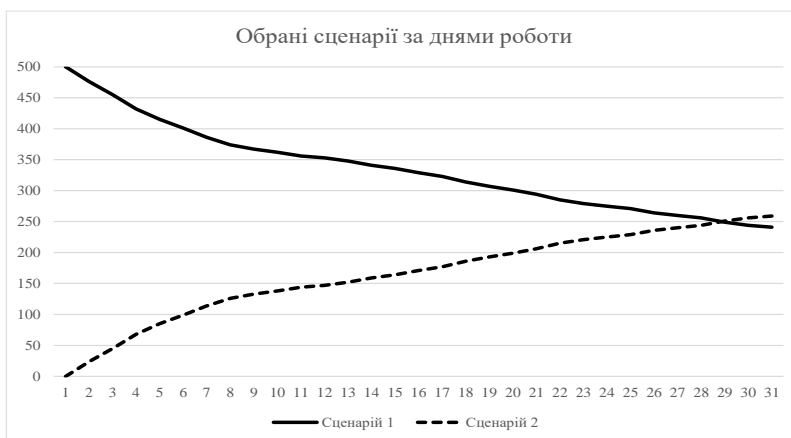


Рис. 2. Графіки кількості користувачів на кожному з сценаріїв за днями роботи

Кількість користувачів за спрощеним сценарієм повільно зменшувалась, а за повним сценарієм – повільно збільшувалась.

Ці дані доводять правильність обраного рішення, яке дозволяю без ручних налаштувань у незнайомому ПЗ розпочати повноцінну роботу з ним. В системі закладений механізм ручного налаштування, але після введення механізму автоматичного налаштування інтерфейсу тільки 7 користувачів з 500 (1,4 %) скористались даною опцією.

**Юсенко А.С., Смокович М.В.**  
*Національний авіаційний університет*

## **МЕТОД ВИДАЛЕННЯ ШУМІВ З ЗОБРАЖЕННЯ**

Матриця фотоапарата, як і будь-який приймач випромінювання, має власні шуми, тобто на корисний сигнал завжди накладається паразитний шум. Чим менша площа чутливого елемента, тим більше відносний рівень шуму.

Щороку проблема обробки цифрових фотознімків стає більш актуальною. У подоланні викликів цієї проблеми мають допомогти нейронні мережі, що використовують алгоритми видалення шуму з фотографій. Перспективним є видалення шумів за допомогою багатокандрового (multi-frame) підходу, що об'єднують в собі переваги оптимізаційних і навчальних методів. Нейронна мережа чергує кроки оцінки щільності руху (dense motion) з кроками реконструкції шару з обструкцією та цільовим зображенням за допомогою підходу «від не точного до точного». Під підходом «від не точного до точного» (coarse-to-fine) в контексті візуального розпізнавання зазвичай розуміється застосування методу до версії зображення з низькою роздільною здатністю, та поступове збільшення здатності і застосування попередніх результатів до «більш точного» зображення. Моделювання щільності руху дозволяє поступово відновлювати деталізований вміст у відповідних шарах зображення. Замість того, щоб покладатися на методи по відокремленню шарів зображення, використовується метод машинного навчання для злиття викривлених потоком зображень, щоб виявити потенційні порушення постійної яскравості і помилки в оцінці потоку. Навчання мережі злиття відбувається з використанням синтетично згенерованого набору даних, який добре себе показує на реальних послідовностях зображень, які раніше не використовувалися.

Цей метод повністю спирається лише на дані, а також вчиться на різноманітних тренінгових даних. Використовуючи глибокі згорткові нейронні мережі (CNN) для оцінки оптичного потоку і реконструкції фонового зображення, цей метод є прийнятним для видалення не лише якось певного типу оклюзій, а загалом може бути використаний для усунення з зображень будь-яких перешкод.

*Дудинець С.В.*

*Національний авіаційний університет*

## **FLUTTER ЯК ЗАСІБ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ**

У наш час мобільні додатки стають неймовірно популярними, їх розробка стає все більш необхідною. Серед множини засобів для розробки таких інструментів Flutter зарекомендував себе досить добре. Flutter – це набір інструментів для розробки мобільного користувацького інтерфейсу, створений компанією Google, що побачив світ у 2017 році.

Flutter поширюється безкоштовно та має відкритий код. Він дозволяє створити мобільний додаток з одним масивом коду. Це означає, що для створення декількох додатків (наприклад для iOS та Android) можна використовувати єдину мову програмування й базу коду. Flutter опирається на дві важливі речі.

Для розробки з Flutter використовується мова програмування Dart. Це також розробка Google, що була створена в 2011 році, але значно вдосконалена за останні роки. Ідеї Dart є розвитком технології верстки веб-сторінок, його також без проблем можна використовувати для створення як мобільних так і веб-додатків. Чому саме Flutter? Він дозволяє змінювати код та бачити результати в реальному часі, це називають Hot-Reload. Деякі зміни все ж можуть перезавантажити додаток, але якщо ви займаєтеся, наприклад, розміткою то це значний плюс в продуктивності праці. Розробка мобільного додатку на Flutter дешевша, тому що не потрібно створювати і підтримувати два додатки (наприклад для iOS та Android).

Віджети, які надаються Flutter легко змінити під будь-яку потребу, що зменшує час розробки і уніфікує всі елементи. З освоєнням допоможе безліч навчальних матеріалів та форумів розробників. А підтримка різних IDE дозволить обрати зручне середовище. Основними IDE вважають Android Studio (IntelliJ) та VS Code, необхідно лише завантажити потрібні для роботи з Flutter та Dart плагіни і можна приступати до роботи.

*Швиндя А.М.*

*Київський національний університет імені Тараса Шевченка*

## ГЕЙМІФІКАЦІЯ ПРОЦЕСІВ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ

Розроблено програмний модуль «Словник-перекладач» для вивчення різних іноземних мов у ігровій формі. Програмний модуль поєднує у собі дві частини: інформативну і навчальну. Інформативна частина містить інформацію про слово, транскрипцію, переклади, тлумачення і приклади вживання. Навчальна частина складається з тестів до слова. Мета роботи – полегшення і пришвидшення процесу вивчення іноземних мов шляхом розроблення програмного модуля «Словник-перекладач».

Закріплення знань про слово відбувається шляхом проходження тестів. Такий підхід, по-перше, уніфікує новий спосіб навчання, по-друге, передбачає адаптивність до нових трендів у вивченні іноземних мов. Зміст і направленість тестів можна адаптувати «на льоту», наприклад, при зміні акцентів з навичок «Use of English» на «Listening». Надано можливість створювати нові тести з аудіо- і відеоматеріалами.

Програмний модуль має два графічних інтерфейси: для користувача (рис. 1) та для backend-розробника.



Рис. 1. Екранна форма застосунку користувача

Програмний модуль «Словник-перекладач» дає змогу якнайшвидше закріпити знання про слова, які вивчаються вибраною іноземною мовою, що важливо в умовах стрімкого сучасного життя, і передбачає легке та швидке масштабування в умовах мінливості методів вивчення мов.

*Росінська Г.П., к.т.н.; Криворучко І.О.  
Національний авіаційний університет*

## **ОСОБЛИВОСТІ ЗМІНИ БІЗНЕС-ПРОЦЕСІВ ПІД ЧАС ВПРОВАДЖЕННЯ ERP-СИСТЕМ**

Самий складний етап впровадження ERP-систем – це переконання власників бізнесу, що ERP проект треба розглядати не якяк ІТ-проект, а не як бізнес-проект.

Це важливий аспект застосування ERP. Організації, які достатньо уваги приділяють управлінню змінами, зазвичай бачать такі результати:

- легше прийняття системи;
- зростання продуктивності;
- досягнення намічених вигод;

Число організацій, які виявляють від помірної до посиленої уваги до управління змінами, збільшилося за останній рік, можливо завдяки тому, що компанії вдалися до послуг консультантів. 85% організацій, які вдавалися до послуг консультантів, присвятили достатню увагу управлінню змінами.

Одна з найважливіших частин управління змінами – це активна розмова зі співробітниками, в якій позначається хто, як і коли впроваджуватиме зміни.

Ранній початок розмови зі співробітниками дуже важливий для успіху проекту. Усі організації, які рано почали розмовляти зі співробітниками перед вибором ERP, зазначили, що їхній проект був успішним. При цьому серед тих, хто почав говорити про зміни під час впровадження або після, лише 86% визнають проект успішним.

Це відбувається тому, що співробітники, з якими не розмовляють, більше пручаються, ніж співробітники, які розуміють значення для всього бізнес-проекту їхньої індивідуальної роботи.

Збільшення масштабу проекту та організаційні питання найчастіші причини, чому бюджет перевищують організації, які відмовляються від застарілих самописних систем. Найчастіша причина – збільшення масштабів проекту, і швидше за все, це пов'язано з тим, що ці організації погано уявляють, наскільки складно влаштовані сучасні технології, а вендори ERP не дають реалістичної інформації.



*Легкий С.Ю.*

*Міжнародний науково-навчальний центр інформаційних технологій та систем НАН України і МОН України*

## **ВИКОРИСТАННЯ REST API СЕРВЕРА У ОБРОБЦІ ЗАМОВЛЕНЬ РЕСТОРАНУ**

На сьогоднішній день неможливо уявити роботу закладів громадського харчування без використання інформаційно-комунікаційних технологій. На ринку багато спеціального програмного забезпечення для автоматизації контролю роботи ресторану і обслуговування клієнтів. Чим більша мережа закладів, тим більше різноманітного ПЗ вона використовує.

Мета роботи – спростити інтеграцію програмного забезпечення мережі ресторанів з різноманітними зовнішніми системами. REST – це стиль архітектури програмного забезпечення для побудови розподілених масштабованих веб-сервісів, заснований на маніпулюванні ресурсами і специфікації HTTP. Він дозволяє охопити всю потужність шляхом побудови інтерфейсів, які можуть використовуватися практично з будь-якого пристрою чи операційної системи. REST ілюструє розвиток Web-архітектури, характеризуючи і регулюючи макровзаємодію чотирьох Web-компонентів, а саме серверів, мережевих шлюзів, проксі і клієнтів, без застосування обмежень до індивідуальних учасників. Таким чином, REST визначає правила поведінки учасників.

В технології REST методи HTTP використовуються для передачі команд, які потрібно виконати на сервері. Використовуються методи GET, POST, PUT, DELETE, які часто порівнюють з CREATE, READ, UPDATE, DELETE (CRUD). Команди та параметри передаються у вигляді URI та в тілі повідомлення.

Використовуючи технології REST маємо можливість інтегрувати програмне забезпечення служби доставки, власний сайт з кошиком замовлень, CRM та багато інших.

Сервер REST API надає зовнішнім системам єдине API для створення, перегляду, видалення, контролю стану замовлень. Розробникам лишається обрати мову програмування та операційну систему для взаємодії з сервером REST API.

*Глазок О.М., к.т.н.; Станіщук К.А.  
Національний авіаційний університет*

## **ЗАСОБИ ОБРОБКИ ТА ВІЗУАЛІЗАЦІЇ ДАНИХ У МОВІ PYTHON**

Засоби Python широко використовують для наукових розрахунків. Мабуть, найбільш популярним у науковців є пакет NumPy, що надає засоби для ефективної роботи із багатовимірними масивами, виконання операцій лінійної алгебри, перетворення Фур'є, генерації випадкових чисел тощо.

Часто використовується і пакет Pandas, який надає функції та структури даних для поліпшення організації роботи із структурованими даними. Пакет надає можливість будувати зведені таблиці, виконувати групування, надає доступ до табличних даних. Pandas представляє дві основні структури даних: DataFrame та Series. DataFrame є фактично табличною структурою даних, бо має стовпці і рядки. Стовпцями в об'єкті DataFrame виступають об'єкти Series, рядки яких є їхніми безпосередніми елементами.

SciPy – збірка пакетів для вирішення деяких стандартних обчислювальних задач: численне інтегрування, вирішення диференціальних рівнянь (scipy.integrate), алгоритми роботи із розрідженими матрицями, розв'язання розріджених систем лінійних рівнянь.

Оскільки графічна форма представлення числових результатів часто буває більш наочною та презентабельною, то в Python присутні і бібліотеки для візуалізації числових даних.

Matplotlib – бібліотека для створення графіків, діаграм та інших візуальних подань двовимірних даних. Так, для візуалізації даних з pandas потрібно імпортувати модуль matplotlib.pyplot та викликати метод plot() на об'єкті DataFrame. Seaborn – це бібліотека для створення статистичної графіки на Python. Він побудований на основі matplotlib і тісно інтегрований зі структурами даних pandas. NetworkX – бібліотека, призначена для роботи з графами та іншими мережевими структурами.

Отже, Python є чудовим інструментом для аналізу та візуалізації даних, а його простота та лаконічний синтаксис дозволить розробникам швидко вирішувати поставлені задачі.

*Нечипорук О.П., Брановицька І.В.*  
*Національний авіаційний університет*

## **РОЗРОБКА МОБІЛЬНИХ ДОДАТКІВ ДЛЯ ОПЕРАЦІЙНОЇ СИСТЕМИ IOS**

Розробка мобільних додатків відіграє все більш важливу роль для організацій, яким необхідно спілкуватися зі співробітниками або клієнтами за допомогою вбудованих додатків. На сьогоднішній день існує великий вибір мов програмування для розробки мобільних додатків. Це пов'язано з тим, що для різних мобільних пристроїв доводиться використовувати різні мови програмування, що обумовлене тим, що мобільні пристрої мають різні операційні системи (ОС).

Цільова платформа (iOS, Android, Windows Phone) буде мати значний вплив на мову розробки, яка буде використовуватися. Наприклад, можна розробляти рідні додатки для кожної платформи або використовувати сторонній інструмент для оптимізації своїх додатків на різних платформах. Другий підхід може заощадити час і зусилля, хоча це може вплинути на зручність використання. Сучасні мобільні пристрої пропонують широкий спектр варіантів розробки.

Процес розробки програмного забезпечення для мобільних пристроїв вимагає підтримувати певні обмеження щодо специфіки роботи додатків у мобільних операційних системах. Особливості роботи мобільних додатків стосуються перш за все:

- на витрати живлення акумуляторної батареї мобільного пристрою;
- обмеження на кількість даних, що передаються через мережу Інтернет;
- зменшену швидкість передачі пакетів в мобільному Інтернет з'єднанні;
- можливість втрати пакетів при передачі в мобільному Інтернет з'єднанні;
- кількість обмінів даними з зовнішніми пристроями на Bluetooth;
- безпеку даних користувача;
- значну фрагментацію версій операційних систем та фреймворків;

- велику різноманітність розмірів екранів мобільних пристроїв;
- обмеження запитів до системи визначення місцезнаходження абонента;
- обмеження на розмір файлу додатку;
- порівняно невеликий ліміт оперативної пам'яті мобільного пристрою;
- порівняно обмежену швидкість передачі даних в мобільному Інтернет з'єднанні.

Тема розробки мобільних застосувань під платформу iOS є досить цікавою і представляє широке поле для подальших досліджень в галузі розробки мобільного ПЗ. Специфіка даного сегмента полягає в тому, що розробка iOS-додатків повинна проводитися з урахуванням особливостей мобільних пристроїв: відмінностями інтерфейсу, іншим розміром екрану, сенсорним управлінням. Актуальність теми підкреслюється широким спектром можливостей для втілення ідей у вигляді мобільного додатку.

*Шмалюк Д.В., Граф М.С., Ph.D*

*Державний університет «Житомирська політехніка»*

## **АНАЛІЗ ECOMMERCE ПЛАТФОРМ ДЛЯ ПРОДУКТОВИХ КОМПАНІЙ ЯКІ ПРАЦЮЮТЬ У СЕГМЕНТІ B2B**

розвитком мережі інтернет та напруженою ситуацією яка нині панує у світі, більшість продуктових компаній які працюють у різних сферах, а особливо у сфері B2B (англ. business-to-business), вирішують питання створення власного електронного магазину. На сьогодні, продуктові компанії мають декілька варіантів створення електронного магазину, перший варіант, створити платформу з нуля, тобто розробити весь функціонал самостійно, але цей варіант не тільки вимагає багато часу, але й є досить дорогим в плані коштів, тому більшість компаній розглядають ті платформи, які вже існують та функціонують.

Постановка проблеми. Проведення аналізу існуючих платформ для B2B та формулювання вимог для удосконалення програмної частини такої платформи з можливістю масштабування, інтеграції з іншими сервісами, додавання та модифікації певної функціональності та максимально можливий рівень захисту даних.

Платформа електронної комерції Shopify [1] - це ідеальний вибір для підприємців малого бізнесу, тобто для тих, які тільки створюють свій бізнес електронної комерції. Shopify надає багато рішень для електронної комерції B2B для створення та розвитку веб-сайту, з якого можливо продавати та відправляти свої продукти. Shopify є популярною CMS для електронної комерції, оскільки має дуже низьку вартість запуску та має простий у використанні інтерфейс. Для того щоб додати або змінити певну функціональність не потрібно бути веб-розробником або кваліфікованим кодером, оскільки Shopify пропонує безліч безкоштовних і преміальних тем, а також інструменти для редагування.

Платформа електронної комерції BigCommerce B2B [2] – чудовий вибір для підприємств малого та середнього бізнесу. Основні функції, що надає платформа – це безкоштовні та платні теми, які не складно налаштувати під потреби фірми, можливість інтеграції з платіжними системами такими як: Google Pay, PayPal,

Apple Pay та іншими, безліч маркетингових інструментів що допомагають прискорити бізнес та, найголовніше, - те що BigCommerce інтегрується з іншими додатками і софтом, що полегшують управління повсякденними операціями магазину. Наприклад це QuickBooks, Olark LiveChat, Optimizely, тощо.

У той же час, існує платформа електронної комерції, яка стане надійним помічником для розширення бізнесу. Ця платформа електронної комерції B2B дає вам можливість масштабуватися від продавця з обмеженою лінійкою продуктів до бізнесу, який обробляє десятки тисяч продуктів. Magento 2 Commerce [3] - це популярна платформа електронної комерції B2B, побудована на технології з відкритим вихідним кодом, яка дозволяє продавцям електронної комерції скористатися перевагами гнучкої системи кошиків для покупок, а також мати можливість контролювати зовнішній вигляд, вміст і функціональність своїх оптових електронних систем. Magento може інтегруватися зі сторонніми рішеннями для електронної комерції B2B. Таким чином, бізнес може легко подолати будь-які прогалини у функціях і спроектувати свій магазин і навіть бізнес так, щоб він працював саме так, як потрібно бізнесу.

Таким чином, було проаналізовано найпопулярніші платформи електронної комерції з точки зору функціональності та потреб бізнесу. Аналіз проводився для малого, середнього та великого бізнесу. Кожна з платформ має свої переваги та певні недоліки. Розглянувши та проаналізувавши кількість клієнтів, які використовують вище проаналізовані платформи та питання, що вирішуються за допомогою даних платформи, було вирішено, що найкращим варіантом є розробка додаткового функціоналу та функцій, взявши за основу Magento 2 Commerce. Дана платформа надає максимальну кількість інструментів для вирішення будь-яких питань, а також, можливість масштабування бізнесу.

Список використаних джерел

1. Детальна інформація про платформу Shopify. URL: <https://www.shopify.com/plus/solutions/b2b-ecommerce>
2. Детальна інформація про платформу BigCommerce B2B. URL: <https://www.bigcommerce.com/solutions/b2b-ecommerce-platform/>
3. Детальна інформація про платформу Magento 2 Commerce. URL: <https://magento.com/products/magento-commerce>

*Мартинов М.Є.*

*Фаховий коледж інженерії та управління*

## **ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ІГРОВІЙ ІНДУСТРІЇ**

Штучний інтелект (ШІ) - це система або машина, яка здатна імітувати людську поведінку для виконання певних завдань і може поступово навчатися, використовуючи отриману інформацію.

Ігровий штучний інтелект - набір програмних методик, які використовуються в комп'ютерних іграх для створення ілюзії інтелекту у поведінці персонажів, керованих комп'ютером.

Нині штучний інтелект вже застосовують на вирішення низки завдань розробки відеоігор. Насамперед, ШІ-алгоритми дозволяють суттєво підвищити якість графіки та природність динаміки різних об'єктів: людей, транспорту, тварин, погодних проявів. Машинне навчання дозволяє виявляти найбільш релевантні інтегральні показники, що відповідають за моделювання цих процесів, що призводить до появи нових ігор вкрай реалістичної графіки. По-друге, всі геймери мріють, щоб алгоритми, що відповідають за моделювання дій суперника у грі, стали хоч трохи наближеними до реальних.

Також використання машинного навчання дозволить реалізувати в ряді ігор таку довгоочікувану можливість, як варіація сюжетних ліній. Зараз більшість ігор мають певний авторами сюжет, довкола якого збудований ігровий світ. Машинне навчання, яке дозволяє моделювати не заданий алгоритм (відповідь певну дію), а формувати новий відповідь системи залежно від інших факторів. Так, вибираючи ті чи інші дії, а також залежно від інших параметрів ігрового світу, геймер зможе вибрати, в який бік скривиться сюжетна лінія.

Також ШІ-рішення дозволять персоналізувати ігровий світ. Скажімо, наприклад, API, що дозволяє аналізувати дані і поведінку в соціальних мережах і передає ці дані в ігровий світ, дозволять цей світ моделювати таким чином, щоб він був найбільш цікавий, приємний геймеру.

Крім комп'ютерних ігор, ігровий штучний інтелект активно застосовується до створення вузькоспеціалізованих військових симуляторів. Військові симулятори вже давно стали частиною системи тренувань Збройних Сил деяких країн (США, Великобританія). При розробці подібних систем особливої значущості набуває правдоподібної поведінки віртуальних солдатів на полі бою.

*Наукове видання*

# СУЧАСНІ ТЕНДЕНЦІЇ РОЗВИТКУ СИСТЕМНОГО ПРОГРАМУВАННЯ

Тези доповідей науково-практичної конференції

25-26 листопада 2021 року

*Тези доповідей надруковані в авторській редакції трьома робочими мовами конференції: українською, російською, англійською*



Підп. до друку \_\_.\_\_.2021. Формат 60x84/16. Папір офс.  
Офс. друк. Ум. друк. арк. 4. Обл.-вид. арк. 4.  
Тираж 30 пр. Замовлення № \_\_\_\_.

Видавець і виготівник  
Національний авіаційний університет  
03680. Київ – 58, проспект Космонавта Комарова, 1

Свідоцтво про внесення до Державного реєстру ДК № 977 від 05.07.2002