

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Одеська національна академія харчових технологій**  
**Університет Інформатики і прикладних знань, м.Лодзь, Польща**  
**Національний технічний університет України «Київський**  
**політехнічний інститут»**  
**Навчально-науковий інститут комп'ютерних систем і технологій**  
**«Індустрія 4.0» ім. П.М. Платонова**

**XXI Всеукраїнська науково-технічна конференція**  
**молодих вчених, аспірантів та студентів**

**«СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ**  
**ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ»**

*Матеріали конференції*



Одеса

22-23 квітня 2021 р.

Стан, досягнення та перспективи інформаційних систем і технологій / Матеріали XXI Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 22-23 квітня 2021 р. - Одеса, Видавництво ОНАХТ, 2021 р. – 229 с.

Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.

## ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Голова - д.т.н., проф., **Єгоров Б.В.**, ректор ОНАХТ.

### Співголови:

**Поварова Н.М.** – к.т.н., доц., проректор з наукової роботи ОНАХТ,  
**Котлик С.В.** – к.т.н., доц., директор ННІКСіТ "Індустрія 4.0" ОНАХТ,  
**Даріуш Долива**, д.математичн.наук, уповноважений декана факультету Інформатики УІтаПЗ, м.Лодзь, Польща,  
**Ковалюк Т.В.** - к.т.н., доц. кафедри АСОІтаУ НТУУ «Київський політехнічний інститут»

### Члени оргкомітету:

**Плотніков В. М.** – д.т.н., проф., завідувач кафедри ІТтаКБ ОНАХТ,  
**Артеменко С.В.** – д.т.н., проф., завідувач кафедри КІ ОНАХТ,  
**Хобін В.А.** – д.т.н., проф., завідувач кафедри АТПтаРС ОНАХТ,  
**Тарасенко В.П.** – д.т.н., проф., завідувач кафедри СКС НТУУ «Київський політехнічний інститут»,  
**Невлюдов І.Ш.** – д.т.н., проф., завідувач кафедри КІТАМ ХНУРЕ,  
**Мельник А.О.** – д.т.н., проф., завідувач кафедри ЕОМ НУ “Львівська політехніка”,  
**Жуков І.А.** – д.т.н., проф., завідувач кафедри КСтаМ НАУ.

Матеріали подано українською, російською та англійською мовами.  
Редактор збірника Котлик С.В.

университет информатики и радиоэлектроники, Республика Беларусь)	
THE STATE OF CYBER SECURITY DEVELOPMENT FOR CERTAIN CRITICAL DOMAINS IN THE REPUBLIC OF MOLDOVA. <b>AURELIAN BUZDUGAN</b> (Moldova State University, Republic of Moldova)	38
АНАЛІЗ ШИФРІВ У БЕЗДРОТОВИХ МЕРЕЖАХ. <b>КУЛЯ Ю.Е.</b> (Харківський національний університет радіоелектроніки), <b>ГАВРИЛОВА А.А.</b> (Харківський національний економічний університет імені Семена Кузнеця)	40
ВИКОРИСТАННЯ СИСТЕМИ ДЛЯ РОЗПОДІЛЕНОГО ЗБЕРІГАННЯ ІНФОРМАЦІЇ В АНТИ-ФОРЕНЗИЦІ. <b>МАКАРЕНКО А.О.</b> (Харківський національний економічний університет імені Семена Кузнеця)	42
DDOS-АТАКИ НА ОСВІТНІ ВЕБ-РЕСУРСИ. <b>КОРОЛЕВИЧ Є.М., ПЛОТНИКОВ В.М., ЗІНЧЕНКО І.І.</b> (Одеська національна академія харчових технологій)	44
ОЦІНКА ПРОБЛЕМ ЗАХИСТУ ІНФОРМАЦІЇ В ІНФРАСТРУКТУРІ “РОЗУМНИЙ БУДИНОК”. <b>ЄРЕЩЕНКО О.Д.</b> , (Харківський національний університет імені Семена Кузнеця)	46
ПРО ВРАХУВАННЯ СТАВЛЕННЯ ДО РИЗИКУ В ПРОСТОРОВИХ СИСТЕМАХ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ. <b>КЛЕПАТСЬКА В.В., БУЧИНСЬКА І.В., КУЗНІЧЕНКО С.Д.</b> (Одеський державний екологічний університет)	47
КЛАСИФІКАЦІЯ ЗАГРОЗ ВЕБ-ЗАСТОСУНКІВ. <b>ЛАВРЕНОВ В.А., СІРЕНКО О.І.</b> , (Одеська національна академія харчових технологій)	49
PROOF OF ZERO-KNOWLEDGE IN THE TASKS OF ANONYMIZATION OF FINANCIAL TRANSACTIONS. <b>ПРОКОПОВ Е.К.</b> (Odessa I.I. Mechnikov National University)	51
РОЗРОБКА ПРОГРАМИ ДЛЯ ПЕРЕВІРКИ ТЕКСТІВ НА ОРИГІНАЛЬНІСТЬ. <b>БЕВЗ С.В., БУРБЕЛО С.М., ВОЙТКО В.В., ЗАВАЛЬНЮК Є.К.</b> (Вінницький національний технічний університет)	52
АУТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ З ВИКОРИСТАННЯМ АНАЛІЗУ КЛАВІАТУРНОГО ПОЧЕРКУ. <b>КАСІЯНЕНКО Д.В.</b> (Київський національний університет імені Тараса Шевченка)	54
РОЗРОБКА УНІВЕРСАЛЬНОЇ ІНФОРМАЦІЙНОЇ УПРАВЛЯЮЧОЇ СИСТЕМИ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ СКЛАДУ. <b>КРИВИЙ Є.О., ШВЕЦЬ Н.В.</b> (Одеська національна академія харчових технологій)	56
ВИКОРИСТАННЯ ГРАФІЧНОГО ФРЕЙМВОРКУ LIBGDX ДЛЯ РОЗРОБКИ КРОСПЛАТФОРМНИХ ІГОР. <b>РОМАНЮК О.Н., ВЕРЕНЬКО А.І., МИРГОРОДСЬКИЙ А. В.</b> (Вінницький національний технічний університет)	58
DEVELOPMENT OF MODELS AND ALGORITHMS FOR THREE-FACTOR AUTHENTICATION SYSTEM. <b>DONETS O.V.</b> (V. N. Karazin Kharkiv National University), <b>RADOUTSKA A.K.</b> (Kharkiv National University of Radio Electronics)	60
КОМП'ЮТЕРИЗОВАНИЙ ВІДБІР ОПЕРАТОРІВ БПЛА. <b>МАРУЩАК А.В., ШМАЛЮХ В.А., РОМАНЮК О.Н., КОВАЛЬ Л.Г.</b> (Вінницький національний технічний університет)	61
ПАСИВНИЙ МЕРЕЖЕВИЙ АНАЛІЗ ТА ЗАСОБИ ЙОГО ВДОСКОНАЛЕННЯ. <b>ЖОЛНЕР І.Д., МИРУТЕНКО Л.В., ШЕСТАК Я.В.</b> (Київський національний університет ім. Тараса Шевченка)	63
АНАЛІЗ ХМАРНОЇ ТЕХНОЛОГІЇ GOOGLE DRIVE. <b>РОМАНЮК О.Н., БОРИСОВА К.О., КАТЄЛЬНИКОВ Д.І.</b> (Вінницький національний технічний університет)	65
АНАЛІЗ МЕХАНІЗМІВ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ В ДЕЦЕНТРАЛІЗОВАНИХ СИСТЕМАХ. <b>ТРОЦІЙ А.О.</b> (Харківський національний економічний університет імені Семена Кузнеця)	67

## **РОЗРОБКА УНІВЕРСАЛЬНОЇ ІНФОРМАЦІЙНОЇ УПРАВЛЯЮЧОЇ СИСТЕМИ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ СКЛАДУ**

КРИВИЙ Є.О. , ШВЕЦЬ Н.В. (*jackgm555@gmail.com, shvetsnv0601@gmail.com*)  
Одеська національна академія харчових технологій

*У даній роботі розглядається розробка інформаційної управляючої системи для роботи складу. Система дозволить автоматизувати та оптимізувати роботу складу.*

Головною метою є створення додатку для комп'ютера, який повинен надавати зручний та лаконічний користувацький інтерфейс, та забезпечувати оптимізовану роботу процесів складського обліку, таких як:

- облік прийому вантажів на склад;
- облік відправки вантажів зі складу;
- відстеження вантажів на складі, їх кількість та умови зберігання;
- робота з постачальниками та споживачами.

Однією із задач є аналіз та ознайомлення з існуючими аналогами, визначення їх переваг та недоліків. Додаток повинен перейняти основні переваги існуючих аналогів та вирішити їх недоліки.

Методи розробки. Створення бази даних за допомогою СУБД PostgreSQL, створення додатку, що буде містити користувацький інтерфейс та бізнес-логіку, на платформі .NET, використовуючи мову C#, а також бібліотеку Npgsql для зв'язку з базою даних.

У даній роботі розглядається розробка програмного забезпечення для оптимізації роботи складу. Ця тема є достатньо актуальною, адже на сьогоднішній день майже кожне підприємство має в розпорядженні склад. Оптиміальна робота складу забезпечує зниження собівартості або накладних витрат для продукції підприємства. Крім того, поширення та доступність комп'ютерів дедалі збільшується. Впровадження систем автоматизації має низку переваг: збільшення темпу обробки замовлень, зниження залежності від певної людини на складі, зниження впливу людського фактору на роботу складу, зменшення роботи з паперовими документами, контроль роботи персоналу, проведення інвентаризації без зупинки роботи складу, тощо. Тому впровадження програмно-апаратних систем у процеси роботи підприємства (в тому числі і в процеси складського обліку) є привабливим кроком для якісного росту для будь-якого підприємства.[1]

Абстрагуючись від особливостей роботи підприємства, можна виділити основні функції, що властиві будь-якому складу, а саме: прийом вантажу, збереження вантажу та відправка вантажу. В умовах стрімкого росту населення планети [2], та збільшенню об'ємів споживання продукції [3], неефективний складський облік може завдати значних збитків підприємству, особливо при масштабуванні виробництва продукції.

В основі проекту лежить розробка загальної частини (без прив'язки до певних типів підприємства) програмно-апаратного комплексу для оптимізації роботи складу. Для цього необхідно створити серверну частину: базу даних за допомогою СУБД PostgreSQL, а також клієнтську частину: додаток на платформі .NET, який надає доступ до бази даних та дозволяє зручно працювати з даними. Крім того, клієнтський додаток повинен мати спосіб авторизації, щоб захистити систему від несанкціонованого доступу до бази даних.

Універсальність додатку досягається тим, що користувачу дається змога налаштувати топологію складу. Для цього користувач повинен мати можливість задати розміри щодо площі складу, а також розмірів стелажів (довжина, ширина, висота, максимальна вага, тощо) та їх розташування. Варто зазначити, що для зручності процес розташування повинен мати лаконічний графічний інтерфейс, який схематично буде відображати розташування стелажів.

Таким чином, під час отримання нового вантажу, система може розрахувати його оптимальне розташування на складі, враховуючи доступні стелажі, їх розміри, габарити вантажів. Ці розрахунки повинні подаватися у вигляді рекомендації, користувач повинен мати можливість змінити розташування вантажу під час прийому, якщо це необхідно, адже система не може врахувати певні зовнішні умови.

Інший аспект універсальності додатку – модульність та незалежність. Не дивлячись на те, що процеси виробництва тісно пов'язані зі складським обліком, розроблювана система не повинна інтегруватись з системою виробництва. Тобто, якщо певний вантаж повинен зі складу потрапити, наприклад, на конвеєр того самого виробництва, у системі це реєструється як звичайна відправка вантажу зі складу. Для цього системі керування виробництвом можна надати доступ до певних даних з БД складу, щоб відстежувати пересування вантажу між відділами. Перевезення зі складу на склад, або відправка зі складу на інше виробництво реєструється в системі аналогічним способом – як звичайна відправка зі складу, лише буде відрізнятися отримувач.

За замовчуванням, клієнтський додаток працює з СУБД PostgreSQL. Але необхідно передбачити легкий перехід на будь-яку іншу СУБД або структуру бази даних. Для досягнення цієї мети необхідно створити додатковий рівень абстракції – деякий інтерфейс роботи з СУБД, який буде використовуватись для створення необхідної логіки. Інтерфейс повинен містити усі необхідні методи для повного забезпечення роботи додатку. Маючи такий інтерфейс, безпосередньо реалізацію можна виконати у вигляді плагіну, який буде приховувати від основної частини додатку усю специфічну логіку, яка відноситься до певної СУБД по БД. Таким чином, в будь-який момент можна реалізувати новий плагін для роботи з іншою СУБД, не змінюючи основну реалізацію додатку. Більш того, з таким підходом додаток не залежить від структури самої бази даних. Головна умова – плагін повинен виконувати усю необхідну логіку для реалізації роботи інтерфейсу.

Для забезпечення універсальності, додаток повинен бути кросплатформними. Як правило, підприємства для організації своєї роботи використовують операційні системи Windows або Linux. Не дивлячись на те, що C# створювався для систем Windows, сьогодні платформа .Net вже портована на системи Linux та MacOS. Таким чином, додаток, створений з використанням мови програмування C#, буде мати усі необхідні переваги щодо підтримки декількох платформ.

Кінцевий результат повинен мати вигляд додатку для комп'ютера на базі операційної системи Windows або Linux. Додаток повинен мати лаконічний інтерфейс, задовільняти потреби складського обліку, щодо прийому, збереженню та відправки вантажів. Крім того, додаток повинен надавати користувачу необхідний інструментарій для налаштування для потреб певного підприємства. Архітектура додатку повинна передбачати легку міграцію на будь-яку СУБД. Впровадження такого додатку повинно зменшити витрати на обслуговування складу підприємства.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Автоматизация склада: снижение убытков и увеличение прибыли [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gd.ru/articles/8412-avtomatizatsiya-sklada>
2. World population [Електронний ресурс] – Режим доступу до ресурсу: <https://www.worldometers.info/world-population/>
3. Food demand [Електронний ресурс] – Режим доступу до ресурсу: <https://farmingfirst.org/Post2015-Food#Slide2>

УДК 004.925

## **ВИКОРИСТАННЯ ГРАФІЧНОГО ФРЕЙМВОРКУ LIBGDX ДЛЯ РОЗРОБКИ КРОСПЛАТФОРМНИХ ІГОР**

РОМАНЮК О.Н., ВЕРЕНЬКО А. І., МИРГОРОДСЬКИЙ А. В.  
Вінницький національний технічний університет

*Розглянуто особливості використання графічного фреймворку LIBGDX для розробки кросплатформних ігор*

Графічні API дозволяють напряму використовувати новий функціонал, але сучасний ринок мультимедійних додатків та ігор вимагає швидкого процесу розробки нових продуктів, а для максимізації прибутку додатково необхідна підтримка великої кількості платформ з різним апаратним забезпеченням. Кросплатформні бібліотеки та фреймворки, такі як libGDX, покликані спростити ці задачі для розробників.

Метою дослідження є вивчення варіантів використання та можливих переваг графічного фреймворку libGDX для розробки кросплатформних ігор.

Графічний фреймворк libGDX – це високопродуктивний, кросплатформний ігровий Java фреймворк з відкритим кодом, що базується на графічному API OpenGL ES та має підтримку WebGL для роботи в веб-браузерах [1]. На відміну від багатьох популярних платформ, заснованих на редакторах, libGDX не обмежує розробника певною методологією, окремим дизайном чи стилем кодування. Він повністю орієнтований на код і дозволяє керувати кожним аспектом створюваного ПЗ. Не зважаючи на це, libGDX пропонує велику кількість додаткових інструментів та бібліотек, які значно розширюють можливості розробників.

Код LibGDX ліцензований під Apache 2.0 – популярною ліцензією для вільного програмного забезпечення від Apache Software Foundation, тому фреймворк активно підтримується спільнотою розробників [2]. Використання Java та JVM в якості платформи для розробки відкриває доступ до цілої екосистеми потужних інструментів: різноманітні середовища розробки, повна підтримка багатьох систем контролю версій, відлагоджувачі з широким спектром функцій для відслідковування недоліків та зручного профілювання додатків, а також велика кількість надійних бібліотек з обширною документацією.

LibGDX використовується для розробки ігор і додатків, що працюють у режимі реального часу. Такі додатки називають RTA. Їхньою особливістю є те, що вони виконують потрібні дії у момент отримання завдання. Додатки реального часу мають ряд переваг, серед яких можна виділити: гнучкість, адаптивність, швидкість доступу, зменшена фрагментація даних тощо.

LibGDX має власний інструмент налаштування проекту. Це застосунок, що дозволяє сформувати проект за параметрами, які задає розробник. Згенерований проект доступний для редагування в будь-якій IDE для мови програмування Java або для розробки ПЗ для конкретної платформи, наприклад, Android Studio.

Під час першого відкриття згенерованого проекту система автоматичного збирання Gradle закінчить підготовку даних і файлів з вихідним кодом для роботи. Отриманий проект поділений на кілька модулів: основний, що містить платформно-незалежний код логіки гри або мультимедійного додатку, та платформно-специфічні: Android, Desktop, Web і iOS. Взаємодія між ними організована таким чином, щоб код для певної платформи залежав та використовував основний код логіки. Тобто, для більшості розробникам необхідно буде написати код один раз і він відразу запрацює на більшості пристроїв, лише за певної потреби або використання специфічних можливості з'явиться необхідність редагувати додатковий код для конкретної платформи. Результуючий початковий стан проекту залежить від обраних в конфігураторі налаштувань.

Архітектура libGDX та його взаємодія з апаратним і програмним забезпеченням показана на рисунку 1.

**XXI Всеукраїнська науково-технічна конференція  
молодих вчених, аспірантів та студентів**

**«СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ  
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ»**

Одеса

22-23 квітня 2021 р.

Збірник включає доповіді учасників конференції. Тези доповідей публікуються у вигляді, в якому вони були подані авторами.

Відповідальність за зміст і форму подачі матеріалу несуть автори статей.

**Редакційна колегія:** Котлик С.В., Корнієнко Ю.К.

**Комп'ютерний набір і верстка:** Соколова О.П.

**Відповідальний за випуск:** Котлик С.В.