

# АЛГОРИТМЫ БЕРЕЖЛИВОГО УПРАВЛЕНИЯ СКЛАДСКИМИ ЗАПАСАМИ НА СОВРЕМЕННОМ ПРОИЗВОДСТВЕ

## ALGORITHMS OF SAFE MANAGEMENT OF STORAGE RESOURCES IN MODERN PRODUCTION

**O. Nuyya**  
**M. Sergeantova**  
**A. Slabogorskaya**  
**D. Krylov**

*Summary.* The article presents a study on warehouse inventory management. The review of processes and strategies aimed at controlling the level of stocks and ensuring continuous supply of goods or services is made. The task is set based on existing methods and technologies of inventory management automation to create a console application for tracking the life cycle of the product nomenclature in the warehouse and further analysis and forecasting of its activity. Lean approach in the issue of inventory management will make it possible to optimize production and warehousing processes. The study presents the results of the algorithms developed by the authors, which were successfully implemented in the production program of the enterprise. As a result of the research, a software solution that meets the requirements of lean manufacturing was created, which will allow the enterprise to integrate technology into the production process, which will be the first stage of its transition to industry 5.0, where the interaction between humans and software reaches a new level.

*Keywords:* automation of inventory management, online warehouse logistics, warehouse digitalization, lean manufacturing, warehouse complex analysis, console application, QR code of the item.

### Введение

Исследуемая тема представляет собой важнейшую часть деятельности любого производства, независимо от его размера и отрасли. В условиях перехода промышленной революции в фазу «Индустрия 5.0», характеризующейся всепроникающей цифровизацией во все сферы производства, ключевой составляющей функционирования предприятия становится рациона-

**Нуйя Ольга Святославовна**  
 кандидат технических наук, доцент,  
 Санкт-Петербургский государственный университет  
 аэрокосмического приборостроения  
 olga.nuyya@ya.ru

**Сержантова Майя Вячеславовна**  
 кандидат технических наук, доцент,  
 Санкт-Петербургский государственный университет  
 аэрокосмического приборостроения  
 хуз43210@mail.ru

**Слабогорская Алиса Алексеевна**  
 Санкт-Петербургский государственный университет  
 аэрокосмического приборостроения  
 alisaslabogorskaya@yandex.ru

**Крылов Дмитрий Максимович**  
 Санкт-Петербургский государственный университет  
 аэрокосмического приборостроения  
 dm.krylov666@gmail.com

*Аннотация.* В статье представлено исследование, посвященное управлению складскими запасами. Произведен обзор процессов и стратегий, направленных на контроль уровня запасов и обеспечение непрерывной поставки товаров или услуг. Ставится задача на основе существующих методов и технологий автоматизации управления запасами создать консольное приложение для отслеживания жизненного цикла номенклатуры товара на складе и дальнейшего анализа и прогнозирования ее активности. Бережливый подход в вопросе управления запасами позволит произвести оптимизацию производственных и складских процессов. В исследовании представлены результаты разработанных авторами алгоритмов, которые были успешно внедрены в производственную программу предприятия. В результате проведенных исследований было создано программное решение, отвечающее требованиям бережливого производства, которое позволит предприятию интегрировать технологию в производственный процесс, что будет первым этапом его перехода к индустрии 5.0, где взаимодействие человека и программного обеспечения достигает нового уровня.

*Ключевые слова:* автоматизация управления складским запасом, складская логистика в режиме онлайн, цифровизация склада, бережливое производство, анализ складского комплекса, консольное приложение, qr-код номенклатуры.

лизация процедур контроля над товарными остатками с использованием цифровых технологий.

Ускоренное продвижение инноваций и постоянно меняющиеся запросы потребителей заставляют производство быть адаптивным и оперативно трансформироваться в духе времени. В настоящее время создание программного обеспечения для автоматизации контроля над хранением товарных запасов явля-

ется важной и обязательной задачей для увеличения способности контролировать жизненный цикл номенклатуры производимых товаров. Программное обеспечение, собирающее всю информацию о продукции на хранении, представляет собой превосходное решение, поскольку обеспечивает полную автоматизацию контроля над запасами на каждом этапе, потому что охватывает весь процесс от закупки товаров до их доставки потребителям.

Выделим основные цели управления запасами:

1. Обеспечение непрерывности производства: гарантирование наличия необходимых товаров для бесперебойного производства и выполнения заказов.
2. Минимизация издержек: снижение затрат на хранение, управление и обработка запасов через приложение.
3. Удовлетворение потребностей клиентов: обеспечение высокого уровня обслуживания клиентов за счет доступа к базе данных, где хранится вся информация о наличии товара.
4. Оптимизация уровня запасов: нахождение оптимального баланса товара на складе без риска нехватки или излишка товара.
5. Улучшение логистических и складских процессов: повышение эффективности и производительности складских операций, сокращение времени на поиск нужного товара на складе.
6. Минимизация рисков: уменьшение возможных потерь товара на складе [1-6].

**Существующие методы и технологии автоматизации управления запасам**

Грамотное управление складскими запасами на производстве позволяет эффективно распределять имеющиеся ресурсы предприятия и осуществлять увеличение рентабельности производства. Однако отметим, что программное обеспечение служит только инструментом для воплощения в жизнь методов управления запасами на складе с учетом поставленной задачи [6–8, 13–14].

В этой связи авторами произведён анализ основных вариантов классификации способов управления запасами (по сложности принятия решений, по периодичности заказа, по объему заказа) и сведен в таблицу 1.

Управление складскими запасами может вестись одним из вышеперечисленных методов или их комбинацией. Наиболее часто используемыми на практике методами являются следующие:

1. Метод фиксированного запаса на складе.
2. Метод с фиксированным периодом заказов у поставщика.
3. Метод фиксированного заказа.

Таблица 1.

Классификации способов управления запасами

Сложность принятия решения	Периодичность заказа	Объем заказа
Одноуровневые, решения в которых принимаются на основании анализа одного показателя, например снижения уровня запаса склада до минимального уровня	Без фиксированной периодичности, при котором заказ делается по мере необходимости	С возможностью произвольно выбирать объем поставки
Многоуровневые, решения в которых принимаются на основе анализа нескольких складских показателей	С фиксированной периодичностью, которая обусловлена преимущественно минимально возможными сроками между поставками	С фиксированным размером заказа

4. Комбинированный метод фиксируемого уровня складских запасов и фиксированной периодичности.
5. Статический метод.
6. Экспертный подход.

Бережливое управление запасами основано на информационных технологиях, таких как программное обеспечение для управления запасами и штрих коды, QR-коды и единицы хранения (SKU) для отслеживания каждого товара. Это позволяет отслеживать запасы в режиме реального времени и поддерживать баланс между заказами товаров у поставщиков и продажами клиентам и как следствие избегать дефицита или избытка товаров [9,15].

Система бережливого управления запасами направлена на сокращение расходов, уменьшение потерь запасов и поиск способов постоянного совершенствования управления запасами.

**Алгоритм разработки приложения для автоматизации управления складскими запасами**

Первым шагом в разработке приложения является продумывание и построение архитектуры приложения. Архитектура приложения — это фундаментальная организация системы, включающая её компоненты, свойства этих компонентов и отношения между ними.

При проектировании архитектуры необходимо учитывать, как функциональные, так и нефункциональные требования. Функциональные требования описывают, что система должна делать, в то время как нефункцио-

нальные требования касаются качества, которым должна обладать система, таких как скорость обработки данных, удобство использования и уровень защищенности информации.

Дизайн архитектуры начинается с выбора подходящего стиля — это может быть многослойная архитектура, сервисно-ориентированная архитектура или микрослужбы. Каждый из этих стилей имеет свои преимущества и ограничения, и выбор зависит от специфики проекта и потребностей бизнеса. Важно провести тщательный анализ и моделирование, чтобы все компоненты системы взаимодействовали эффективно, обеспечивая при этом высокое качество конечного продукта.

Разработка Архитектуры включает в себя несколько взаимосвязанных этапов, каждый из которых играет ключевую роль в создании эффективной и оптимизированной системы. На первом этапе осуществляется выбор структурных элементов и их интерфейсов, где важно учитывать, как функциональные требования, так и особенности взаимодействия между элементами. Эти элементы должны быть проектированы так, чтобы их поведение в рамках системы способствовало продуктивному сотрудничеству и минимизировало возможные конфликты.

На втором этапе происходит соединение выбранных элементов в более крупные системы, что требует внимательного контроля за интеграцией и взаимодействием функциональных и нефункциональных требований. Этот процесс зачастую требует итеративного подхода, позволяющего адаптировать архитектуру по мере накопления опыта и выявления новых деталей.

Последний, но не менее важный этап — это определение архитектурного стиля, который задаёт рамки для всей организации. Он настраивает взаимосвязь между всеми элементами, их интерфейсами и способами сотрудничества, обеспечивая целостность и согласованность архитектурного решения. Это становится основой для дальнейшего развития системы и её адаптации к изменяющимся условиям.

Существует несколько различных типов архитектур, которые были созданы за все время существования разработки приложений и данных. Среди них: монолитная, клиент-серверная, распределенная, микросервисная.

Авторами используется клиент-серверная архитектура. В этой архитектуре сервер предоставляет ресурсы или сервисы, а клиент выполняет запросы к серверу для получения этих ресурсов или сервисов. В нашем случае сервер управляет базой данных, а приложения на Android и ПК выступают в роли клиентов, которые обращаются к серверу для выполнения операций с данными

(добавление, удаление, чтение). Архитектура «клиент-сервер» представляет собой мощный и гибкий подход к организации взаимодействия между пользователями и системами. Благодаря централизованному управлению данными, все необходимые данные накапливаются на сервере, что облегчает их администрирование и защиту. Это позволяет администраторам эффективно выполнять обновления и создавать резервные копии, минимизируя риск потери информации.

Специализация архитектуры также играет ключевую роль: серверы, оптимизированные для обработки запросов и выполнения бизнес-логики, обеспечивают надежную и быструю работу, в то время как клиентские устройства фокусируются на создании интуитивно понятного пользовательского интерфейса. Такой подход позволяет максимально эффективно использовать ресурсы и улучшает взаимодействие пользователей с системой.

Масштабируемость системы является еще одним важным преимуществом архитектуры. Возможность увеличить мощность существующего сервера или добавить новые серверы позволяет системе адаптироваться к растущим потребностям бизнеса. Выходя за рамки простого обслуживания, архитектура «клиент-сервер» предлагает централизованное обновление, что гарантирует, что все клиенты всегда используют актуальные версии сервисов и функций.

В рассматриваемом проекте сервер выполняет централизованное хранение и управление данными, а клиенты зависят от сервера для доступа к данным, это является архитектурой «клиент-сервер». Блок-схема проекта представлена на рисунке 1.

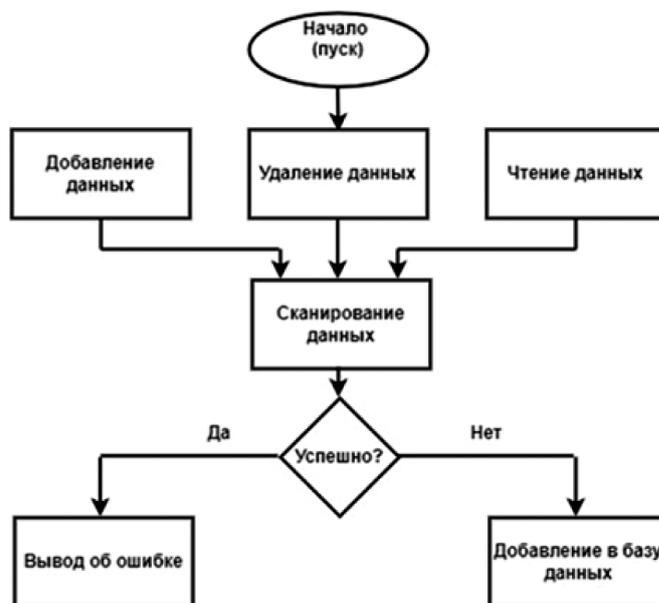


Рис. 1. Блок-схема проекта

**Создание мобильного приложения в Android Studio**

Android Studio — это официальная интегрированная среда разработки (IDE) для создания приложений на операционную систему Android. Разработанная компанией Google, она предоставляет разработчикам комплексный набор инструментов для проектирования, написания, тестирования и отладки мобильных приложений. Android Studio поддерживает языки программирования Java и Kotlin, и включает в себя такие функции, как визуальный редактор интерфейсов, который позволяет перетаскивать элементы для создания экранов приложения, а также мощную систему сборки на основе Gradle, которая даёт возможность настраивать процесс сборки приложения.

Как и любая IDE, Android Studio упрощает написание кода, она умеет автоматически искать и исправлять ошибки в программах и даже писать за вас код. Для последнего используется Studio Bot — встроенный искусственный интеллект (ИИ), с которым можно общаться на естественном языке. Когда код будет написан и протестирован, Android Studio приведёт его к формату, подходящему для публикации в Google Play, AppGallery, RuStore и других магазинах приложений для Android. Для создания проекта нужно выбрать Empty Activity, потому что в других Activity есть уже различные заготовки, которые не отвечают требованиям поставленной задачи [10–16].

**Разработка структуры и дизайна мобильного приложения**

Структура приложения разработана для удобной работы с товарами с помощью QR-кодов и включает в себя три основные функции: добавление товара, удаление товара и предоставление информации о товаре. Интер-

фейс пользователя должен быть максимально простым и интуитивно понятным, чтобы обеспечить легкость в использовании даже для тех, кто не имеет опыта работы с подобными приложениями. Интерфейс каждой из функций разработан таким образом, чтобы минимизировать количество шагов, необходимых для выполнения задачи, и исключить возможность путаницы или ошибки со стороны пользователя. На главном экране приложения расположены три кнопки, каждая из которых соответствует одной из функций:

1. Кнопка «Добавить товар». Эта кнопка запускает процесс сканирования QR-кода для добавления нового товара в базу данных. После сканирования кода открывается форма, где пользователь может ввести всю необходимую информацию о товаре. Все шаги сопровождаются подсказками, что делает процесс добавления товара понятным и легким.
2. Кнопка «Удалить товар». При выборе этой функции пользователь сканирует QR-код товара, который следует удалить. После идентификации товара система запрашивает подтверждение на удаление, чтобы исключить случайные ошибки. Процесс удаления товара интуитивно понятен и не требует дополнительных сложных действий.
3. Кнопка «Информация о товаре». Эта кнопка позволяет пользователю получить детальную информацию о товаре путем сканирования его QR-кода. Информация отображается в четком и удобочитаемом формате, предоставляя все необходимые сведения о товаре, такие как название и остаток на складе.

Разработанное авторами консольное приложение обладает простым и интуитивно понятным интерфейсом, что позволит осуществлять загрузку и обработку информации из базы данных эффективнее. Функционал приложения обеспечивает быстрый онлайн доступ

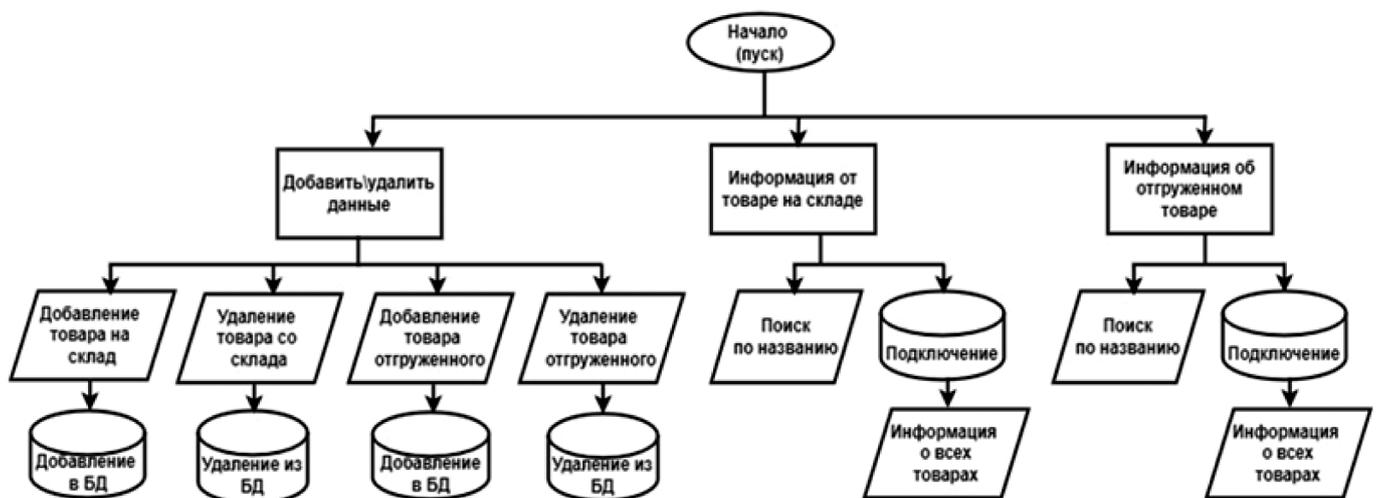


Рис. 2. Блок-схема консольного приложения

к данным о номенклатуре товара, что решает проблему оперативного контроля и управления складскими запасами. В дальнейшем информация из консольного приложения позволит анализировать историю отгрузок для оптимизации закупочной деятельности и стимулирования продаж.

Разработанная структура и дизайн приложения сформированы с учетом лучших практик проектирования пользовательского интерфейса, чтобы обеспечить работу с приложением, отвечая требованиям технического задания. Для генерации QR-кодов номенклатуры продукции разработана программа на языке програм-

мирования Python с использованием библиотеки qrcode (рисунок 6) [10–16]. Блок-схема консольного приложения представлена на рисунке 2. Страница консольного приложения представлена на рисунке 3.

На рисунке 4 и 5 представлена страница приложения об информации об отгрузке товара и информация о товаре на складе.

Код для генерации QR-кода написан с помощью языка программирования Python и представлен на рисунке 6.



Рис. 3. Страница консольного приложения

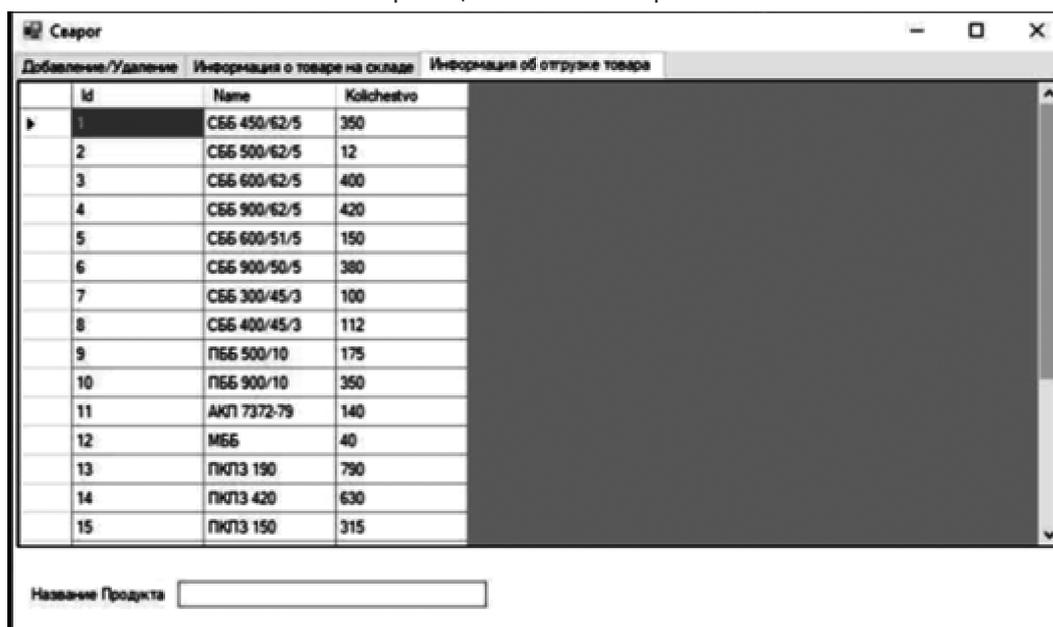


Рис. 4. Информация об отгрузке товара

Склад

Добавление/Удаление | Информация о товаре на складе | Информация об отгрузке товара

№	Наим.	Количество	Место хранения
1	СББ 450/52/5	350	2Б
2	СББ 500/52/5	250	1А
3	СББ 600/52/5	400	2А
4	СББ 900/52/5	420	3А
5	СББ 600/51/5	150	4А
6	СББ 900/50/5	300	5А
7	СББ 300/45/3	100	1Б
8	СББ 400/45/3	112	2Б
9	ПББ 500/10	175	1В
10	ПББ 900/10	250	2В
11	АКП 7373-79	140	3В
12	МББ	40	3Б
13	ПКПЗ 190	790	1Г
14	ПКПЗ 420	630	2Г
15	ПКПЗ 150	315	3Г
16	СББ 900/100/5	400	4Г

Название продукта:

Рис. 5. Информация о товаре на складе

```

qr.py x qr_code.png
1 import qrcode
2
3 # Создаем объект QRCode
4 qr = qrcode.QRCode(
5     version=1,
6     error_correction=qrcode.constants.ERROR_CORRECT_L,
7     box_size=10,
8     border=4,
9 )
10
11 # Добавляем данные в QR-код
12 qr.add_data('Пример текста или URL')
13 qr.make(fit=True)
14
15 # Создаем изображение QR-кода
16 img = qr.make_image(fill='black', back_color='white')
17
18 # Сохраняем изображение в файл
19 img.save('qr_code.png')

```

Рис. 6. Генерация QR-кодов номенклатуры продукции

### Заключение

В результате проведенного исследования разработано мобильное приложение, предназначенное для сканирования QR-кодов товаров и их последующего внесения в базу данных. Создано консольное приложение, предоставляющее специалистам возможность осуществлять полный контроль над номенклатурой товаров на складе.

В результате авторами разработано программное решение, которое соответствует принципам бережливого производства и позволяет предприятиям экологично интегрироваться в индустрию 5.0, где взаимодействие человека и программного обеспечения выходит на новый уровень.

### ЛИТЕРАТУРА

1. Вейдер Майкл. Инструменты бережливого производства II. Карманное руководство по практике применения Lean. — М.: Альпина Паблишер, 2017. — 125 с.
2. Womack J.P., Jones D.T. Lean Thinking: Banish Waste and Create Wealth in Your Corporation, Revised and Updated [2nd ed.] — Free Press. — 2003. — 398 p.
3. Савич Ю.А. Инструментарий стратегического управления конкурентоспособностью предприятия / Ю.А. Савич // Экономинфо. -2019. — Т. 16. — № 1. — С. 27–32.
4. Fang X., & He C. (2021). Using Vendor Management inventory system for goods inventory management in IoT manufacturing. Enterprise Information Systems, 16(7). <https://doi.org/10.1080/17517575.2021.1885743>
5. Colaković A., Caušević S., Kosovac A., & Muharemović E. (2020). A review of enabling technologies and solutions for iot based smart warehouse monitoring system. Lecture Notes in Networks and Systems, 128 LNNS, 630–637. [https://doi.org/10.1007/978-3-030-46817-0\\_73](https://doi.org/10.1007/978-3-030-46817-0_73)
6. Kumar D., Kr Singh R., Mishra R., & Fosso Wamba, S. (2022). Applications of the internet of things for optimizing warehousing and logistics operations: A systematic literature review and future research directions. In Computers & Industrial Engineering (Vol. 171, p. 108455). Elsevier BV. <https://doi.org/10.1016/j.cie.2022.108455>
7. R.A. Sabitov, G.S. Smirnova, Sh.R. Sabitov, B.M. Morozov, B.R. Sirazetdinov Adaptive control and operational management system of machine-tool fleet of the manufacturing enterprise IFAC-PapersOnLine, 48 (3) (2015), pp. 1236–1241
8. S. Zhang et al. A hybrid multi-objective approach for real-time flexible production scheduling and rescheduling under dynamic environment in Industry 4.0 context Comput. Oper. Res. (2021)
9. QR-код (быстрый отклик). [Электронный ресурс]. — URL:<https://www.geeksforgeeks.org/how-to-build-a-qr-code-android-app-usingfirebase/> (дата обращения 15.02.2025).
10. Приложение на Kotlin под Android [Электронный ресурс]. — URL:<https://habr.com/ru/articles/275255/> (дата обращения 25.04.2024). использование REST API [Электронный ресурс]. — URL:<https://code.tutsplus.com/ru/android-from-scratch-using-rest-apis--cms-27117t> (дата обращения 25.02.2025).
11. Создание простого консольного приложения C# в Visual Studio [Электронный ресурс]. — URL: <https://learn.microsoft.com/ruru/visualstudio/get-started/csharp/tutorial-console?view=vs-2022> (дата обращения 20.02.2025).
12. C# консоль, подключение базы данных [Электронный ресурс]. — URL: <https://ru.stackoverflow.com/questions/577859/c-консоль-подключениебазы-данных> (дата обращения 13.02.2025).
13. Программы для складского учета в 2024 году [Электронный ресурс]. —URL: <https://www.kp.ru/money/biznes/luchshie-programmy-dlya-skladskogoucheta/> (дата обращения 29.01.2025).
14. Методы управления запасами на складе [Электронный ресурс]. — URL:<https://www.ekam.ru/blogs/pos/metody-upravleniya-zapasami-na-sklade> (дата обращения 25.01.2025).
15. Создание QR-кодов на Python [Электронный ресурс]. — URL:<https://proglib.io/p/generaciya-krasivyh-qr-kodov-s-pomoshchyu-python-2023-11-07> (дата обращения 13.02.2025).
16. SQL — ЯЗЫК РЕЛЯЦИОННЫХ БАЗ ДАННЫХ Учебное пособие [Электронный ресурс]. — URL: [https://elar.ufr.ru/bitstream/10995/40612/1/978-5-7996-1622-9\\_2016.pdf](https://elar.ufr.ru/bitstream/10995/40612/1/978-5-7996-1622-9_2016.pdf) (дата обращения 17.02.2025).

© Нуйя Ольга Святославовна (olga.nuyya@ya.ru); Сержантова Майя Вячеславовна (хуз43210@mail.ru); Слабогорская Алиса Алексеевна (aliskaslabogorskaya@yandex.ru); Крылов Дмитрий Максимович (dm.krylov666@gmail.com)  
Журнал «Современная наука: актуальные проблемы теории и практики»