
КАФЕДРА

КУРСОВАЯ РАБОТА
ЗАЩИЩЕНА С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

Курсовая работа

«Модели управления запасами»

По дисциплине: Прикладные методы оптимизации

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2024

Содержание отчета:

Введение.....	4
Актуальность работы.....	4
а) Теоретическая актуальность:	4
б) Практическая актуальность:	4
Цель исследования.....	5
Задачи исследования	5
Объект исследования.....	6
Предмет исследования.....	7
Теоретическая основа работы	8
Методы исследования.....	9
Практическая значимость результатов исследования	10
Основная часть	12
Глава 1: Теоретические аспекты управления запасами	12
1. История и развитие теорий управления запасами:	12
2. Основные модели управления запасами: классификация и анализ:	17
3. Анализ Эффективности Различных Моделей Управления Запасами ...	21
4. Современные тенденции и инновации в управлении запасами:	23
Заключение Главы 1	24
Глава 2: Расчётно-аналитический аспект управления запасами	25
1. Методы оптимизации запасов: аналитический обзор.....	25
2. Применение моделей управления запасами на практике	28
3. Анализ эффективности моделей управления запасами	29
4. Разработка и тестирование новой модели управления запасами	30
Разработка, описание всех компонентов программы:	34
Заключение Главы 2	52
Глава 3: Проектная часть.....	53
1. Проектирование модели управления запасами для конкретного предприятия:	53
Создание данных для конкретного предприятия:.....	54
Вывод программы после анализа для конкретного предприятия	60
2. Анализ рисков и оценка эффективности предложенной модели	62
3. Рекомендации по внедрению и использованию модели на практике ...	63
4. Оценка потенциала масштабирования модели:	66
Заключение.....	68
Основные Выводы	68
Рекомендации	69

Потенциал для Дальнейшего Исследования.....	69
Список использованных источников	70
Приложение	74
 Листинг программы для тестового предприятия:	74
 Результат работы программы:	79

Введение

Актуальность работы

Актуальность темы "Модели управления запасами" обусловлена как теоретическими, так и практическими аспектами в сфере менеджмента.

а) Теоретическая актуальность:

Эволюция теорий управления запасами: В последние десятилетия наблюдается значительное развитие теорий управления запасами. Исследования в этой области выявили новые подходы и стратегии, адаптируемые под меняющиеся рыночные условия и технологический прогресс. Это обуславливает необходимость постоянного обновления теоретической базы и разработки новых моделей.

Интеграция с другими дисциплинами: Управление запасами тесно связано с другими аспектами бизнеса, такими как логистика, финансы и операционный менеджмент. Появление новых исследований в этих сферах способствует развитию интегрированных моделей управления запасами.

б) Практическая актуальность:

Оптимизация затрат и повышение эффективности: В условиях жесткой конкуренции и постоянно меняющегося рынка компании стремятся оптимизировать свои затраты и повысить эффективность операций. Эффективное управление запасами позволяет существенно снизить издержки и улучшить показатели работы предприятия.

Адаптация к меняющимся рыночным условиям: Современный рынок характеризуется быстрыми изменениями в спросе и предложении, что требует гибкого подхода к управлению запасами. Разработка новых моделей, способных адаптироваться к таким изменениям, представляет собой важную задачу для практикующих менеджеров.

Технологические инновации: Прогресс в области информационных технологий и автоматизации открывает новые возможности для управления

запасами. Интеграция этих технологий в модели управления требует постоянной актуализации знаний и подходов.

Цель исследования

Целью данной курсовой работы является разработка и анализ эффективных моделей управления запасами, адаптированных к современным условиям функционирования предприятий. Эта цель включает в себя несколько ключевых аспектов:

1. **Изучение существующих моделей:** Анализ теоретических основ и практического применения существующих моделей управления запасами, включая как традиционные, так и новые подходы.
2. **Адаптация моделей под современные условия:** Понимание того, как изменения в экономике, технологиях и рыночной среде влияют на эффективность моделей управления запасами, и разработка модификаций, которые учитывают эти изменения.
3. **Разработка новой модели:** Создание новой модели управления запасами, которая объединяет лучшие практики и новые идеи, соответствуя требованиям и спецификам современного бизнес-окружения.
4. **Практическое применение и оценка эффективности:** Проверка разработанной модели на практике, анализ её эффективности и применимости в различных условиях и типах предприятий.

Задачи исследования

Для достижения поставленной цели исследования, "Разработка и анализ эффективных моделей управления запасами", определяются следующие задачи:

1. **Изучение теоретических основ управления запасами:**
 - Анализ литературных источников по теме управления запасами, включая классические и современные научные работы.

- Оценка развития и эволюции теорий управления запасами, идентификация ключевых концепций и подходов.

2. Анализ существующих моделей управления запасами:

- Рассмотрение и оценка различных существующих моделей управления запасами, включая традиционные и инновационные методы.
- Определение преимуществ и недостатков различных подходов в контексте разнообразных бизнес-сред.

3. Разработка новой модели управления запасами:

- Синтезирование информации, полученной из теоретического обзора и анализа существующих моделей, для создания новой модели.
- Учет специфики и требований современных предприятий при разработке модели.

4. Эмпирическое исследование и оценка предложенной модели:

- Применение разработанной модели на практическом примере или кейсе.
- Анализ эффективности и применимости модели, сбор и обработка данных.

Объект исследования

Объект исследования в данной курсовой работе – это процессы управления запасами на предприятиях. Этот объект включает в себя несколько ключевых аспектов:

1. **Системы управления запасами:** Охватывают различные подходы и методологии, используемые предприятиями для контроля и управления своими запасами. Это включает в себя как традиционные системы, так и современные, автоматизированные решения.

2. **Операционные процессы:** Включают в себя действия и процедуры, связанные с планированием, закупкой, хранением, учетом и распределением запасов внутри организации.
3. **Взаимодействие с другими функциями бизнеса:** Особое внимание уделяется тому, как процессы управления запасами интегрируются и взаимодействуют с другими функциями предприятия, такими как продажи, маркетинг, финансы и логистика.
4. **Внешняя среда:** Рассмотрение влияния внешних факторов, таких как рыночные тренды, изменения в спросе и предложении, а также технологические инновации на процессы управления запасами.
5. **Стратегическое управление:** Анализ, как стратегические решения компании влияют на управление запасами, и наоборот, как управление запасами может способствовать достижению стратегических целей организации.

Предмет исследования

Предметом исследования в курсовой работе по теме "Модели управления запасами" являются конкретные модели и методы управления запасами, а также их применение и эффективность в контексте современного управленческого процесса. В рамках данного исследования будут рассмотрены следующие аспекты:

1. **Специфика различных моделей управления запасами:** Детальный анализ разнообразных подходов и моделей, применяемых для управления запасами, включая как классические, так и современные методики.
2. **Факторы, влияющие на выбор и эффективность моделей:** Исследование того, как различные внутренние и внешние факторы (например, размер предприятия, отраслевые особенности, рыночная ситуация) влияют на выбор и успешность применения различных моделей управления запасами.

3. **Оценка эффективности моделей управления запасами:** Анализ критериев и методов оценки эффективности применения различных моделей управления запасами, включая анализ затрат, уровня обслуживания клиентов и гибкости системы.

Теоретическая основа работы

Теоретическая основа работы включает в себя следующие ключевые источники и авторы:

1. **Классические теории управления запасами:**

- Работы Форда У. Харриса (Ford W. Harris) по модели Экономического размера заказа (EOQ), представленные в его статье "How Many Parts to Make at Once" (1913).
- Исследования Тайичи Оно (Taiichi Ohno) и его вклад в разработку системы "точно в срок" (Just-In-Time), описанные в книге "Toyota Production System: Beyond Large-Scale Production" (1988).

2. **Современные теории и подходы:**

- Книга "Supply Chain Management: Strategy, Planning, and Operation" (6th Edition) авторства Сунил Чопры (Sunil Chopra) и Питера Мейндла (Peter Meindl), описывающая современные методы управления цепочками поставок.
- Работы Сергея Лукьянова и Елены Лукьяновой, особенно их книга "Управление запасами: Учебник" (2014), в которой рассматриваются современные российские подходы к управлению запасами.

3. **Авторы и исследователи:**

- Джон Стерман (John Sterman) и его книга "Business Dynamics: Systems Thinking and Modeling for a Complex World" (2000), освещающая системный подход к управлению запасами.

- Работы Александра Васильева, особенно его статьи по оптимизации запасов в условиях неопределенности, публикуемые в российских научных журналах.

4. Аналитические и оптимизационные методы в управлении запасами:

- Книга "Inventory Management and Production Planning and Scheduling" Эдварда Силвера (Edward A. Silver) и др., демонстрирующая использование математических методов в управлении запасами.

5. Применение информационных технологий в управлении запасами:

- Статьи и исследования по влиянию IT-решений на управление запасами, опубликованные в журнале "International Journal of Production Economics".

Методы исследования

В данной курсовой работе использовались следующие методы исследования для анализа и разработки моделей управления запасами:

- 1. Литературный анализ:** Изучение и анализ научных статей, книг и других источников, посвященных теме управления запасами. Этот метод позволил получить обзор существующих теорий и практик, а также выявить ключевые направления для дальнейшего исследования.
- 2. Математическое моделирование:** Применение математических методов для разработки и тестирования моделей управления запасами. Включает в себя использование статистических инструментов и методов оптимизации для анализа эффективности различных подходов.
- 3. Анализ конкретных случаев:** Изучение и анализ реальных примеров управления запасами на предприятиях. Это позволяет оценить практическую применимость и эффективность теоретических моделей.

4. **Сравнительный анализ:** Сопоставление различных моделей управления запасами для выявления их сильных и слабых сторон, а также для определения условий их оптимального применения.

Новизна исследования

Новизна данной курсовой работы заключается в следующих аспектах:

Разработка усовершенствованных моделей управления запасами:

Основываясь на классических и современных теориях, исследование предлагает новые подходы и модели, адаптированные к текущим рыночным условиям и технологическим возможностям.

Интеграция с современными технологиями: Включение в анализ влияния информационных технологий, таких как большие данные (Big Data) и искусственный интеллект, на процессы управления запасами.

Практическая значимость результатов исследования

Практическая значимость данной курсовой работы заключается в следующих аспектах:

Улучшение процессов управления запасами на предприятиях: Результаты исследования предлагают усовершенствованные модели и методы, которые могут быть использованы предприятиями для оптимизации управления запасами. Это включает в себя повышение эффективности, снижение издержек и улучшение уровня обслуживания клиентов.

Помощь в принятии управленческих решений: Разработанные модели и методы анализа могут быть использованы руководителями и менеджерами для принятия обоснованных решений в области управления запасами, что особенно важно в условиях неопределенности и изменчивости рынка.

Адаптация к различным типам предприятий: Результаты исследования предоставляют гибкие решения, которые могут быть адаптированы к специфике различных видов бизнеса, включая производственные и торговые компании различных масштабов.

Основа для дальнейших исследований: Работа может служить основой для дальнейших научных исследований в области управления запасами, включая разработку новых моделей и методов, а также их тестирование и улучшение.

Основная часть

Глава 1: Теоретические аспекты управления запасами

1. История и развитие теорий управления запасами:

Этот раздел курсовой работы посвящён изучению исторического развития теорий управления запасами. Он охватывает переход от ранних методов к современным подходам, выявляя ключевые этапы развития и их влияние на практики управления запасами.

Ранние методы и практики управления запасами

В этом разделе мы углубленно рассматриваем начальные этапы развития управления запасами, когда оно основывалось на интуитивных подходах и эмпирических правилах, до появления более сложных систем и методов.

Интуитивные Подходы и Эмпирические Правила:

В самом начале управления запасами, основным подходом была ориентация на личный опыт и интуицию менеджеров. Они оценивали необходимое количество товаров на складе, опираясь на данные предыдущих продаж и прогнозируемый спрос. Ранние методы учета запасов в основном представляли собой ручные записи и визуальные проверки уровня запасов, часто без применения сложных аналитических или статистических методов.

Переход к Организованным Системам:

С развитием предприятий и увеличением ассортимента товаров возникла потребность в более организованных системах учета запасов. Были созданы детальные записи о товарах на складе, их движении и обороте. Параллельно начали разрабатываться простые методы прогнозирования спроса, основанные на исторических данных о продажах. Несмотря на их ограниченность и недостаточное учет многих важных факторов, эти методы стали первым шагом к развитию более сложных моделей управления запасами.

Ограничения и Проблемы Ранних Подходов:

Ранние методы характеризовались ненадежностью и высоким риском ошибок, особенно в условиях изменяющегося спроса и непредсказуемых рыночных условий. По мере роста предприятий и усложнения их структуры, эти методы становились неэффективными, влекущими за собой излишки или нехватку запасов, увеличение затрат и потерю доходов.

Переосмысление Управления Запасами:

Разочарование в ограниченности и неэффективности ранних методов привело к поиску новых, более эффективных подходов в управлении запасами. Это стало стимулом для развития более сложных математических моделей и интеграции компьютерных технологий, что заложило основу для современных методов управления запасами.

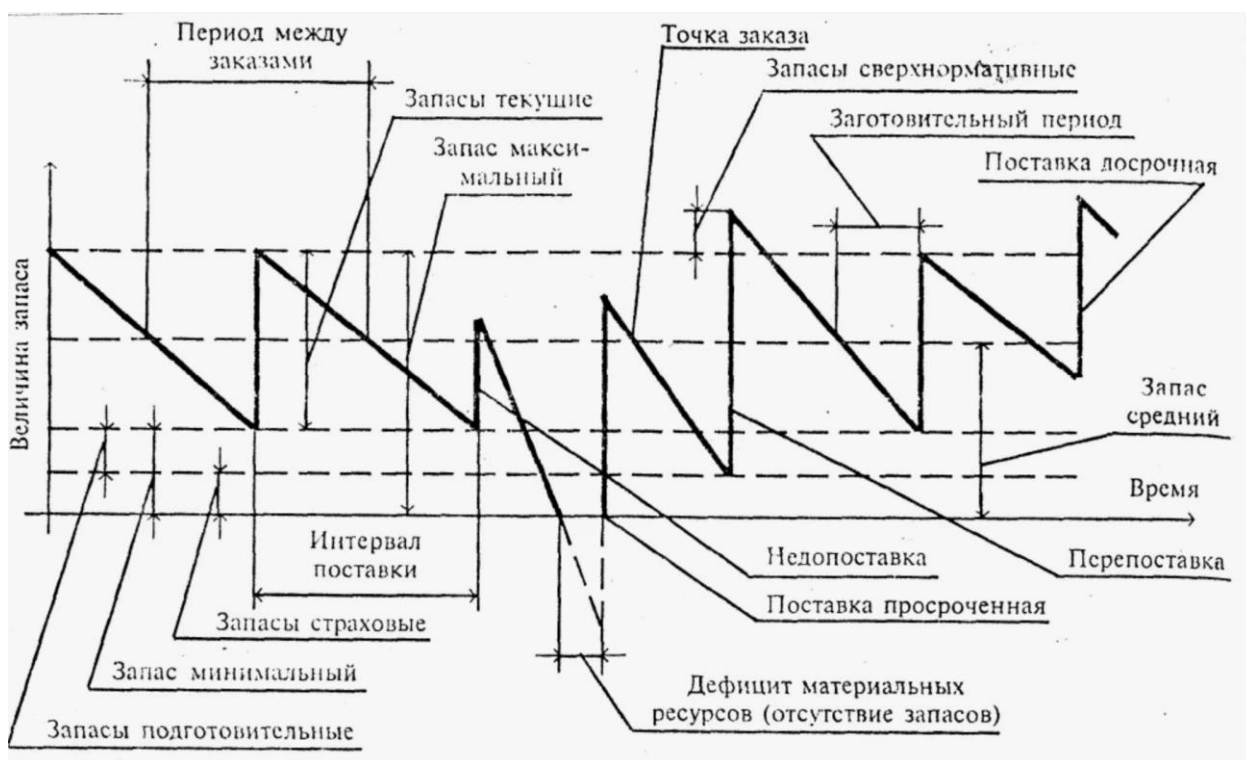


Рисунок 1 - обозначения на модели управления запасами

Развитие математических моделей управления запасами

В начале XX века произошёл значительный прогресс в области управления запасами, особенно с введением математических методов, которые заложили основу для современных теорий и практик.

Экономический размер заказа (ЕОQ) - Форд У. Харрис, 1913 год

Одним из первых значительных вкладов в математические методы управления запасами стала модель ЕОQ, разработанная Фордом У. Харрисом. Эта модель представляет собой революционный шаг в оптимизации запасов, предлагая формулу для определения идеального количества заказа, которое минимизирует общие затраты. Ключевым элементом ЕОQ является поиск баланса между затратами на хранение запасов и затратами на их заказ.

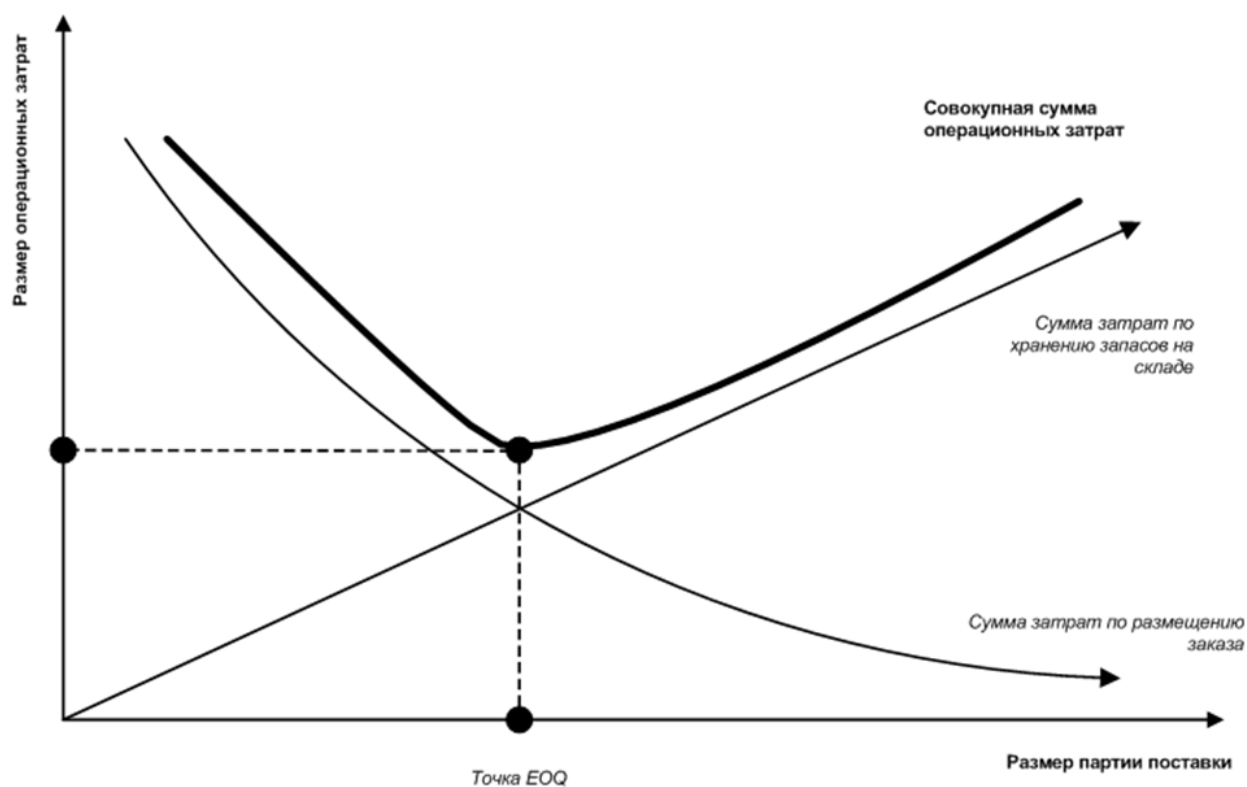


Рисунок 2 - EOQ модель управления запасами

Развитие статистических и вероятностных моделей

С увеличением сложности рыночных условий и вариативности спроса, возникла потребность в более сложных моделях, которые учитывали бы неопределенность и риски. Это привело к разработке статистических и вероятностных моделей управления запасами. Эти модели включали анализ рисков и неопределенности спроса, позволяя предприятиям лучше

адаптироваться к меняющимся условиям и минимизировать потенциальные потери.

Примеры применения статистических моделей

Примеры таких моделей включают управление запасами при ненадежных поставках и управление запасами с учетом вероятности спроса. Эти модели используют сложные статистические методы для оценки рисков и определения оптимальных стратегий запсообразования, позволяя предприятиям эффективно справляться с неопределенностями и сохранять конкурентоспособность.

В целом, введение и развитие математических моделей в управлении запасами значительно повысило точность и эффективность принятия решений в этой области, положив начало современным методам оптимизации запасов.

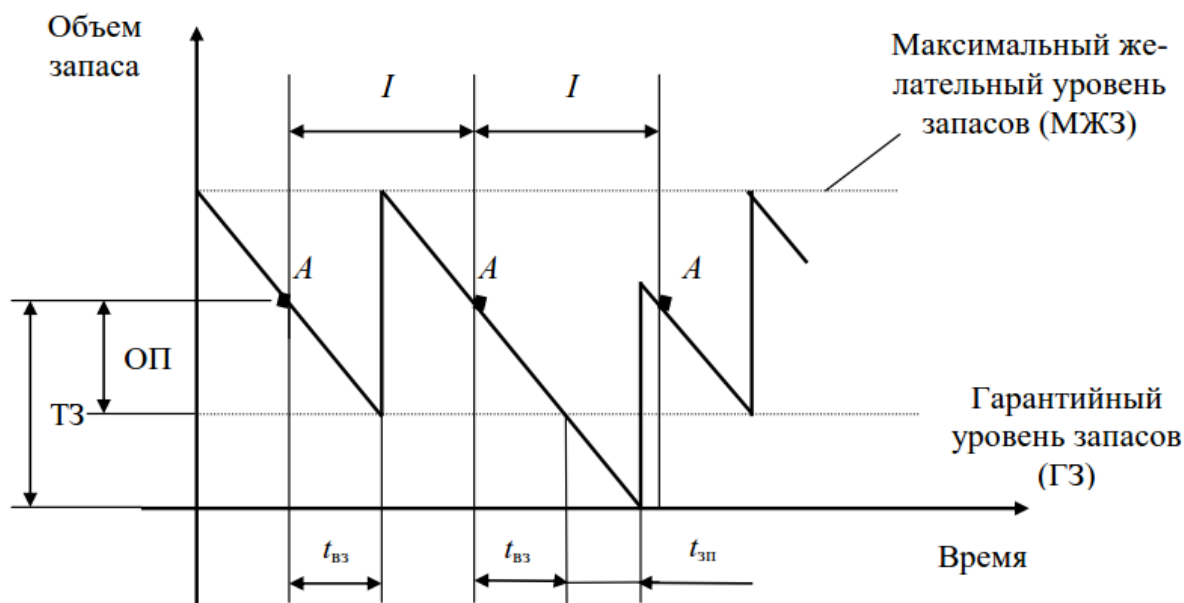


Рисунок 3 - система с фиксированным интервалом времени

Влияние Технологического Прогресса на Управление Запасами

Внедрение компьютерных технологий и автоматизации оказало существенное влияние на управление запасами, что способствовало развитию более сложных

и эффективных систем. Этот раздел исследует ключевые аспекты этого влияния.

Развитие Сложных Систем Управления Запасами:

С появлением компьютерных технологий и автоматизации произошло значительное усовершенствование процессов управления запасами. Это позволило переходить от ручного учета и анализа к автоматизированным системам, способным обрабатывать большие объемы данных и проводить сложные аналитические расчеты.

Автоматизация также привела к улучшению точности прогнозирования спроса, оптимизации размеров заказов и эффективному планированию запасов, сокращая издержки и увеличивая общую эффективность управления запасами.

Влияние Информационных Систем на Управление Запасами:

Системы планирования ресурсов предприятия (ERP) и системы управления складом (WMS) оказали значительное влияние на управление запасами. Эти системы обеспечивают централизованный доступ к данным, улучшают координацию между различными подразделениями предприятия и позволяют эффективно управлять запасами на всех этапах цепочки поставок.

ERP-системы интегрируют информацию о запасах с другими бизнес-процессами, такими как финансы, закупки и продажи, обеспечивая комплексный подход к управлению запасами. WMS-системы специализируются на оптимизации процессов на складе, улучшая точность учета запасов, управление складскими операциями и логистику.

2. Основные модели управления запасами: классификация и анализ:

Традиционные модели управления запасами:

Традиционные модели управления запасами играют ключевую роль в логистике и цепочках поставок. Ниже приводится подробное описание трех основных традиционных моделей:

Модель Экономического Размера Заказа (EOQ):

Описание: EOQ — это математическая модель, которая помогает определить оптимальное количество единиц товара для заказа, минимизируя сумму затрат на хранение и заказы.

Принципы расчета: Расчет EOQ учитывает годовой спрос, стоимость размещения заказа и затраты на хранение единицы товара. Формула EOQ помогает найти баланс между частотой заказов и размером партии товаров, стремясь к снижению общих затрат на управление запасами.

Применение: Эта модель особенно эффективна для предприятий с предсказуемым и стабильным спросом, где стоимость хранения и заказа остается постоянной.

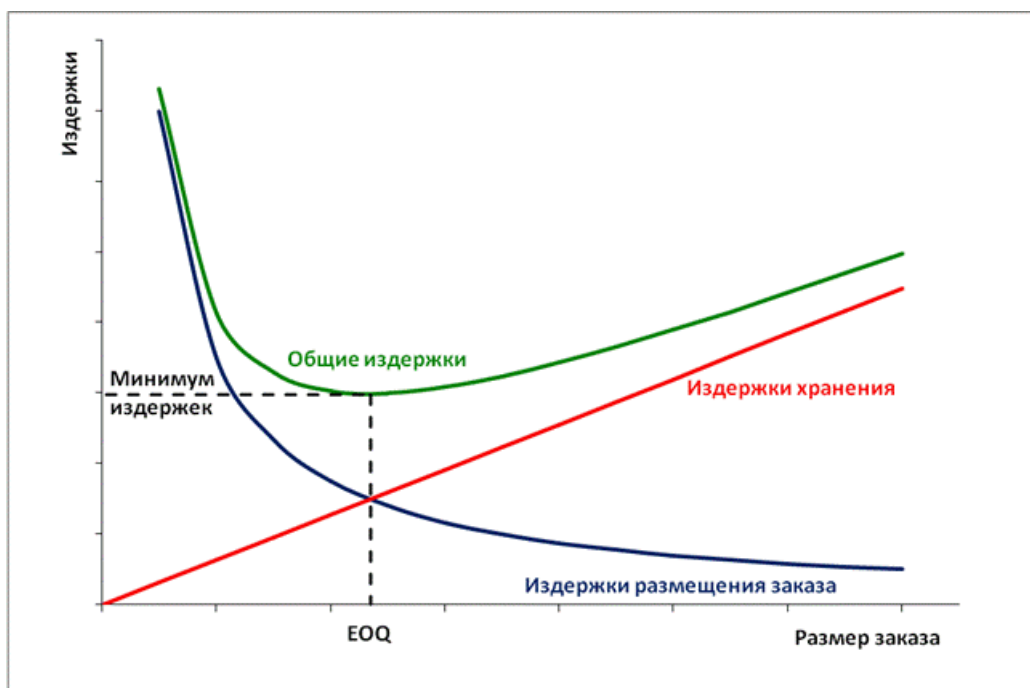


Рисунок 4 – EOQ модуль управления запасами

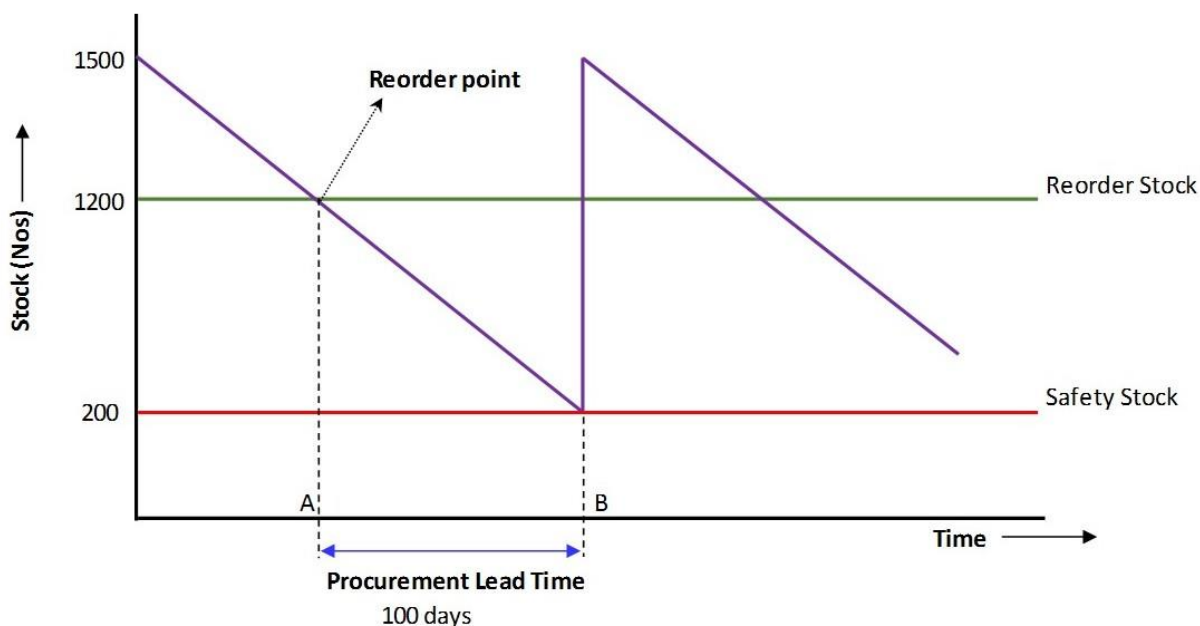
Модель Повторного Заказа (Reorder Point Model):

Описание: Модель повторного заказа определяет точку, при которой необходимо размещать новый заказ, чтобы предотвратить исчерпание запасов.

Эта точка, или уровень запасов, рассчитывается на основе среднего времени выполнения заказа и среднего или ожидаемого уровня потребления.

Ключевые факторы: Важными элементами для расчета являются время выполнения заказа, средний спрос и степень допустимого риска исчерпания запасов.

Применение: Модель широко применяется в условиях, когда важно поддерживать постоянный уровень запасов и быстро реагировать на изменения спроса.



Note: A - Reorder point triggered & Procurement initiated

B - Material received

Рисунок 5 – модель Reorder Point

Модель Управления Запасами По Партиям (Batch Ordering):

Описание: Этот подход фокусируется на заказе и хранении товаров партиями.

Размер партии может варьироваться в зависимости от специфики товара и рыночных условий.

Особенности: Основное преимущество этой модели заключается в оптимизации затрат на заказы и хранение за счет массовых закупок. Это также позволяет предприятиям лучше адаптироваться к сезонным колебаниям спроса, планируя закупки и запасы в соответствии с прогнозируемыми тенденциями.

Применение: Модель особенно подходит для товаров с сезонным спросом или когда поставщики предлагают скидки на крупные партии.

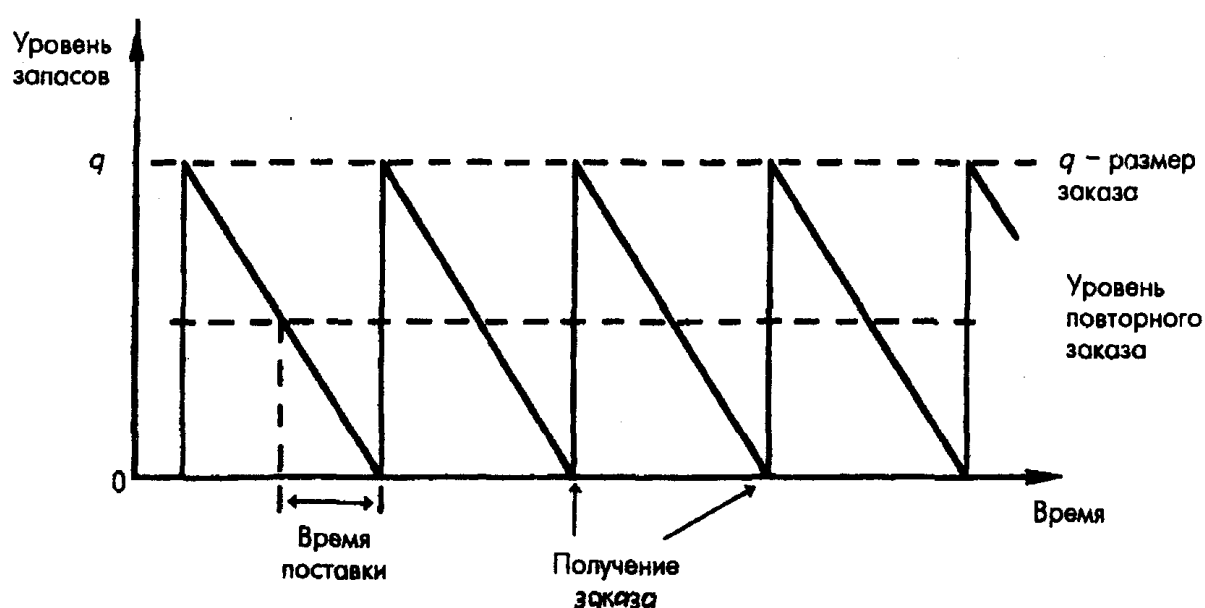


Рисунок 6 – модель Batch Ordering

Современные Модели Управления Запасами

Современный подход к управлению запасами охватывает ряд инновационных моделей, адаптированных к быстро меняющимся рыночным условиям и технологическому прогрессу. Две ключевые современные модели включают Just-In-Time (JIT) и модели на основе информационных технологий.

Just-In-Time (JIT):

Описание: JIT - это стратегия управления запасами, при которой материалы и товары поставляются точно в срок, необходимый для непосредственного использования в производстве или продаже, что позволяет минимизировать или полностью исключить хранение запасов.

Основные принципы: Ключевой идеей JIT является производство товаров только в ответ на спрос, а не для запаса. Это помогает снизить издержки,

Применение: ЛТ широко применяется в производственных и розничных операциях, где скорость и время реакции на спрос имеют критическое значение. Этот подход требует точного планирования и тесного сотрудничества с поставщиками.



Применение: Модели на основе ИТ особенно актуальны в условиях высокой волатильности рынка и разнообразных потребностей клиентов. Они

позволяют компаниям быстро адаптироваться к изменениям спроса и эффективно управлять запасами на всех этапах цепочки поставок.

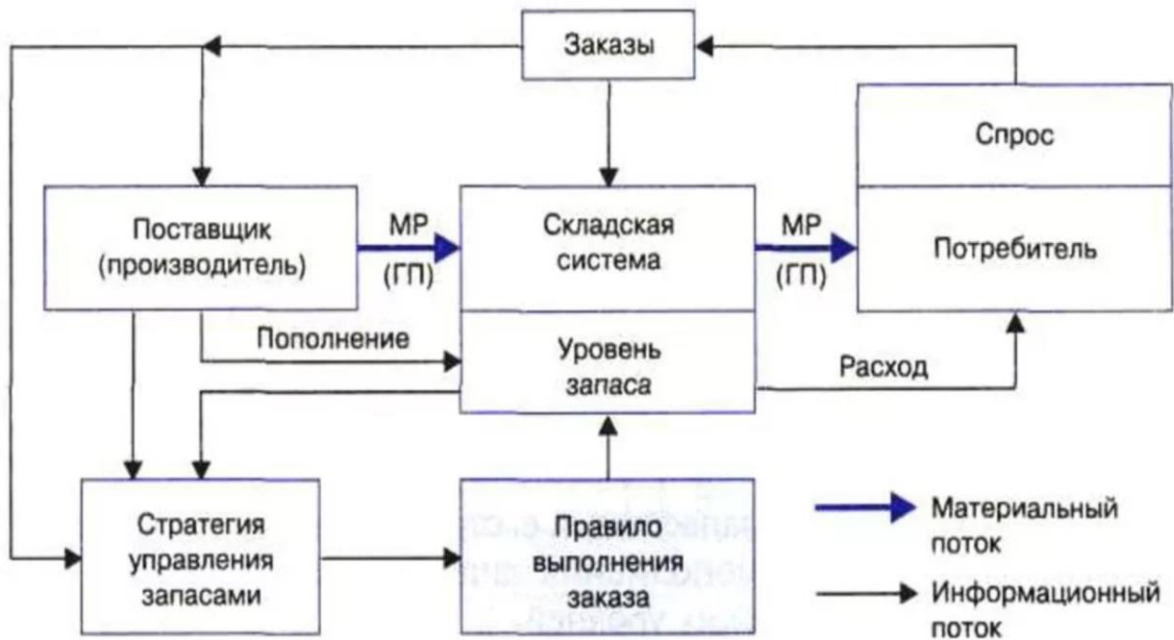


Рисунок 9 – процесс управления товарными запасами

3. Анализ Эффективности Различных Моделей Управления Запасами

В этом разделе проводится глубокий анализ различных моделей управления запасами, сравнивая их эффективность, преимущества и недостатки в различных бизнес-контекстах.

А) Сравнение Традиционных и Современных Подходов:

Преимущества и Недостатки: Традиционные модели, такие как EOQ и модель повторного заказа, часто характеризуются простотой и предсказуемостью, но могут не учитывать быстрые изменения в спросе и поставках. Современные модели, такие как JIT и модели на основе ИТ, более гибкие и адаптируемые, но требуют сложной координации и точного планирования.

Оценка по Типам Предприятий и Рыночным Условиям:

Различные модели могут быть более или менее эффективными в зависимости от размера предприятия, отрасли, волатильности рынка и характера спроса. Например, JIT может быть идеальным для компаний с высокой скоростью

оборота и стабильным спросом, тогда как EOQ лучше подходит для предприятий с предсказуемым спросом и длительными циклами поставок.

Б) Практические примеры:

Пример применения JIT в автомобильной промышленности:

Одним из ярких примеров использования JIT является компания Toyota, которая внедрила этот подход для сокращения издержек и увеличения эффективности производства. Переход к JIT позволил Toyota значительно сократить время на подготовку производственных мощностей, уменьшить запасы на складах и повысить качество продукции. В результате было достигнуто снижение затрат и увеличение операционной гибкости, что позволило компании быстро реагировать на меняющиеся требования рынка.

Использование моделей на основе ИТ в ритейле:

Примером может служить использование продвинутых систем планирования и прогнозирования на основе больших данных крупной розничной сетью Walmart. Благодаря интеграции данных о продажах, инвентаризации и потребительском поведении, Walmart смогла оптимизировать уровни запасов в своих магазинах, уменьшить затраты на хранение и повысить уровень удовлетворенности клиентов за счет более эффективного удовлетворения спроса.

В) Влияние на финансовые показатели и операционную эффективность:

Финансовые показатели:

Модели управления запасами непосредственно влияют на общие затраты предприятия. Например, переход к модели JIT может существенно сократить затраты на хранение и потери от устаревания товаров. С другой стороны, модели, основанные на больших данных и аналитических инструментах, помогают предприятиям минимизировать излишки и недостатки запасов, что приводит к оптимизации операционных расходов.

Операционная эффективность:

Эффективное управление запасами напрямую влияет на операционные процессы предприятия. Например, JIT повышает производственную гибкость и позволяет предприятию быстрее реагировать на изменения спроса. В то же

время, модели на основе ИТ повышают точность прогнозирования спроса и оптимизации запасов, что способствует более гладкому и эффективному процессу удовлетворения потребностей клиентов.

4. Современные тенденции и инновации в управлении запасами:

В этом разделе исследуются ключевые современные тенденции и инновационные подходы, оказывающие значительное влияние на стратегии управления запасами.

Влияние Lean Management и Just-In-Time:

Lean Management: Эта концепция, сфокусированная на минимизации отходов и повышении эффективности, оказала значительное влияние на управление запасами. Применение принципов бережливого управления ведет к оптимизации запасов, уменьшению излишков и сокращению затрат.

Just-In-Time (JIT): JIT – это стратегия управления запасами, которая ориентируется на минимизацию запасов и сокращение затрат за счет точной поставки необходимых товаров в нужное время и в нужном количестве. Этот подход помогает снизить издержки на хранение и увеличить операционную эффективность.

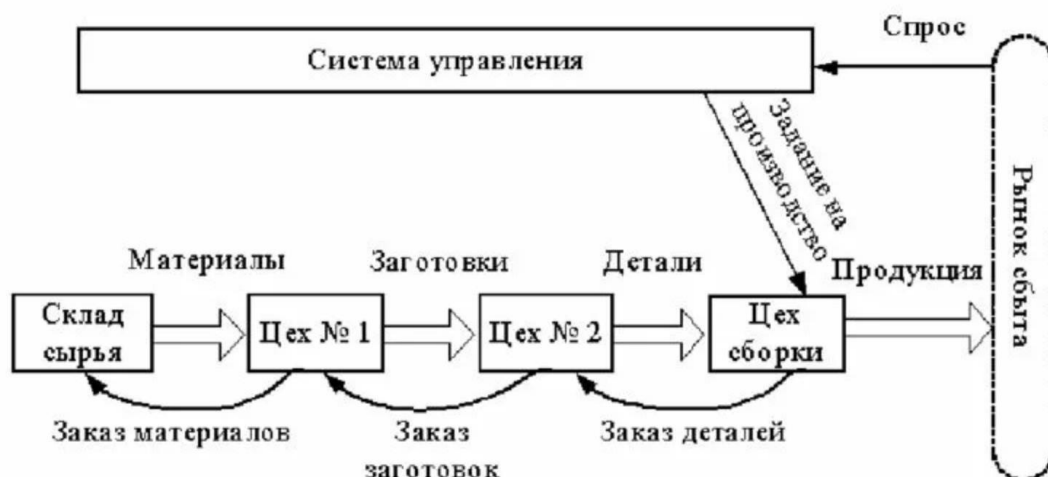


Рисунок 10 – вытягивающая система в JIT

Новейшие Тренды в Технологиях Управления Запасами:

Большие данные (Big Data): Использование больших данных позволяет компаниям анализировать обширные и сложные наборы данных для более точного прогнозирования спроса и поведения рынка. Это ведет к более обоснованным решениям о закупках и уровнях запасов.

Искусственный интеллект (AI) и Машинное Обучение: Применение AI и машинного обучения в управлении запасами открывает новые возможности для автоматизации принятия решений и оптимизации запасов. Эти технологии способны анализировать тенденции, предсказывать спрос и оптимизировать процессы управления запасами, что ведет к повышению эффективности и снижению затрат.

Заключение Главы 1

В заключении этого раздела подводится итог тому, как исторические изменения в управлении запасами оказали влияние на развитие современных практик и подходов в этой области. Анализ эволюции от ранних интуитивных методов до современных технологически продвинутых систем управления запасами подчеркивает не только прогресс в данной области, но и важность адаптации к меняющимся условиям и потребностям рынка.

Изучение истории управления запасами показывает, что эффективность управления запасами всегда была критическим фактором успеха для предприятий. От традиционных методов, ориентированных на простые принципы и интуитивные решения, до современных моделей, интегрирующих сложные аналитические инструменты и информационные технологии, каждый этап развития вносил свой вклад в повышение эффективности и оптимизацию процессов управления запасами.

Особое внимание в этом историческом обзоре уделяется тому, как уроки прошлого могут быть использованы для формирования будущих стратегий управления запасами. Это включает в себя не только применение новейших технологических достижений, но и учет гибкости и адаптивности в подходах

к управлению запасами, что является ключом к успеху в постоянно меняющемся и непредсказуемом бизнес-окружении.

В заключение, анализ исторических изменений в управлении запасами не только помогает понять, как развивались текущие практики, но и указывает на направления будущего развития в этой области. Он подчеркивает важность инноваций, гибкости и постоянного стремления к улучшению процессов в управлении запасами для удовлетворения потребностей современного рынка.

Глава 2: Расчётно-аналитический аспект управления запасами

1. Методы оптимизации запасов: аналитический обзор

Аналитический Обзор Методов Оптимизации Запасов

Анализ Традиционных Методов:

Экономический Размер Заказа (EOQ):

- Проведен анализ ситуаций, в которых применение EOQ наиболее эффективно. Например, при стабильном и предсказуемом спросе, EOQ помогает достичь значительной экономии затрат. Однако, при изменчивом спросе этот метод может приводить к пере- или недозаказам.

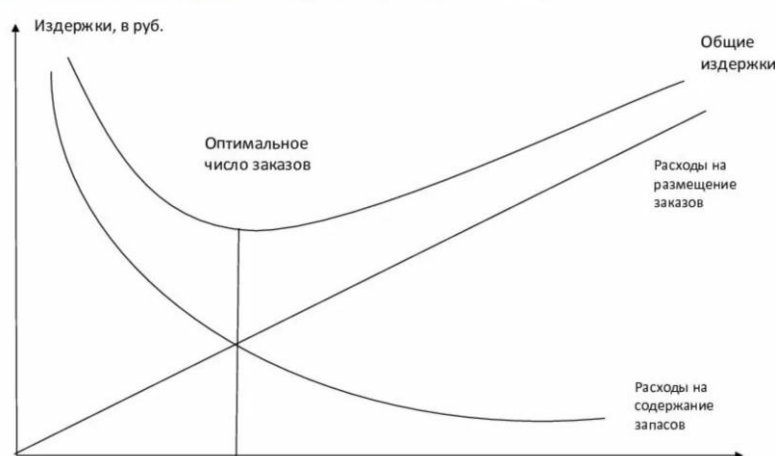


Рисунок 11 – экономический размер заказа

Модель Повторного Заказа:

- Изучено влияние различных уровней спроса и времени выполнения заказа на эффективность модели. Выявлено, что

модель хорошо справляется с управлением запасами при коротких сроках поставок и умеренно изменчивом спросе.

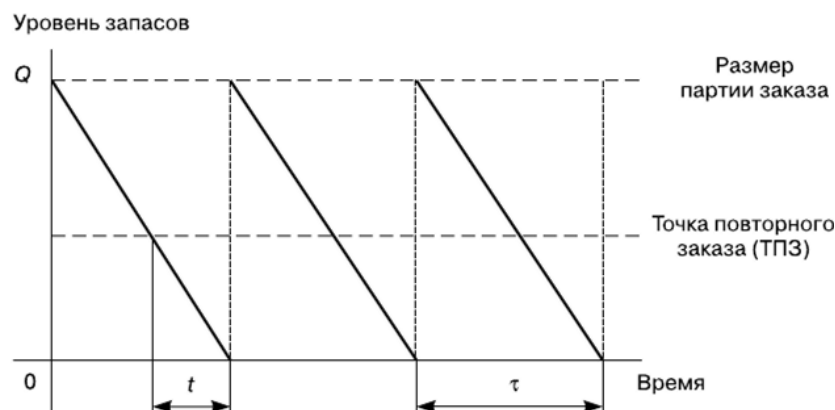


Рисунок 12 – повторный заказ

Модель Управления Запасами По Партиям:

- Анализ показал, что модель наиболее полезна для предприятий с сезонным спросом. При этом, важно точно определить размер партии, чтобы минимизировать затраты на хранение и избежать недостатка товара в пик сезона.

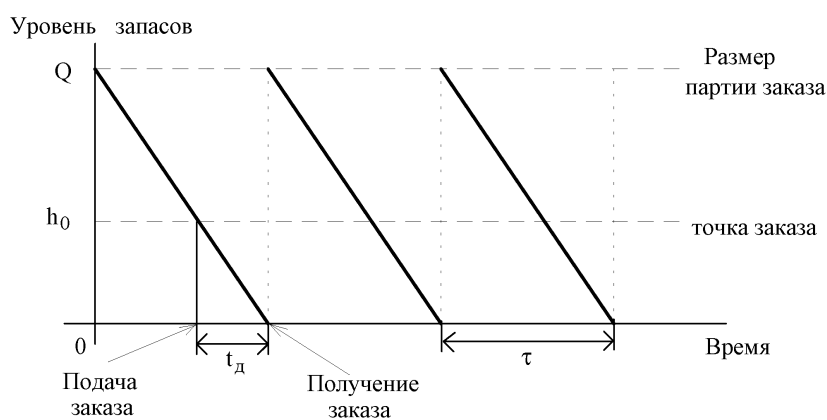


Рисунок 13 – циклы изменения запасов

Анализ Современных Методов Оптимизации:

Just-In-Time (JIT):

- Проанализированы преимущества и риски ЛТ. Эффективность подхода ЛТ демонстрируется в контексте высокой координации с поставщиками и точного прогнозирования спроса. Однако, он подвержен риску при сбоях в цепочке поставок.

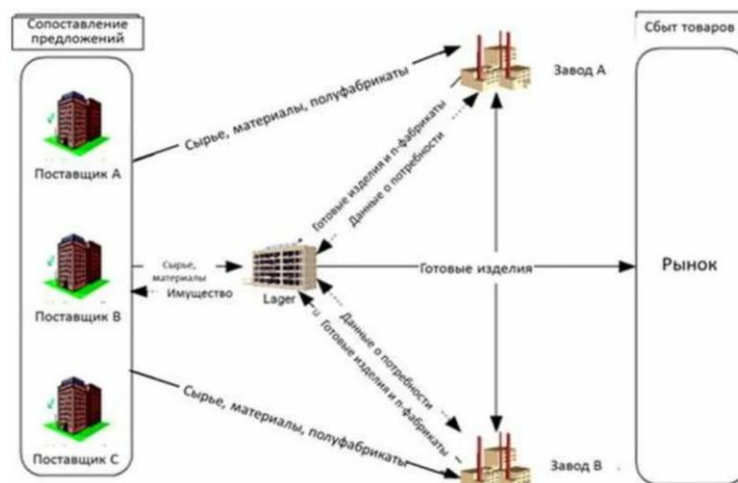


Рисунок 14 – пример стратегии JIT

Информационные Технологии в Оптимизации Запасов:

- Исследовано, как использование аналитики больших данных и продвинутых информационных систем способствует повышению точности прогнозов и эффективности управления запасами. Однако, выявлено, что успешное применение этих технологий требует высококвалифицированных специалистов и значительных инвестиций в IT-инфраструктуру.

Аналитические Инструменты и Методы:

• Статистический Анализ и Моделирование:

- Проведен анализ исторических данных о продажах с использованием статистических методов для определения трендов и сезонности спроса, что позволяет улучшить точность прогнозов и оптимизировать уровни запасов.

• Сценарный Анализ:

- Использован сценарный анализ для моделирования различных рыночных условий, включая колебания спроса и изменения в цепочке поставок. Это позволяет оценить потенциальное влияние этих факторов на эффективность выбранной модели управления запасами.

2. Применение моделей управления запасами на практике

Этот раздел фокусируется на практическом применении различных моделей управления запасами, демонстрируя их реальное воздействие и эффективность в разных бизнес-контекстах.

Применение Модели ЕОQ в Розничной Торговле:

- Рассмотрим пример использования модели ЕОQ в розничном магазине. Здесь модель ЕОQ помогает определить оптимальное количество единиц товара для заказа, учитывая затраты на закупку и хранение, а также годовой спрос. Это обеспечивает минимизацию общих затрат на управление запасами и уменьшение риска недостатка товаров. (Рисунок 2, 4, 11)

Использование JIT в Производственной Компании:

- В случае производственного предприятия, принципы JIT могут быть использованы для минимизации запасов сырья и компонентов. Поставки осуществляются непосредственно в момент их необходимости в производственном процессе, что позволяет сократить затраты на хранение и уменьшить капитал, замороженный в незавершенном производстве. (Рисунок 14)

Применение Современных ИТ-решений в Логистике:

- Применение систем на основе больших данных и аналитических инструментов позволяет предсказывать спрос, оптимизировать маршруты поставок и уровни запасов на складах, что способствует увеличению эффективности цепочек поставок.

Гибридные Модели в Мультиканальном Ритейле:

- В условиях мультиканального ритейла, где продажи осуществляются как через физические магазины, так и онлайн, гибридные модели управления запасами могут оказаться особенно эффективными. Они позволяют гибко управлять запасами в разных каналах продаж, оптимизируя общие затраты и улучшая удовлетворенность клиентов.

Эти примеры показывают, что эффективное управление запасами требует глубокого понимания специфики бизнеса и рыночных условий. Модели

должны быть адаптированы к конкретным требованиям и целям предприятия, чтобы максимизировать их эффективность и положительно влиять на финансовые результаты.

3. Анализ эффективности моделей управления запасами

Анализ Традиционных Моделей:

- **ЕОQ:** Простота и предсказуемость ЕОQ делают её эффективной для стабильных рыночных условий. Однако, она может быть неэффективной в динамичной среде с частыми колебаниями спроса, поскольку не учитывает волатильность рынка. (Рисунок 2, 4, 11)
- **Модель повторного заказа:** Эта модель эффективна для поддержания надежности поставок, но может привести к избыточным запасам или дефициту в случае непредвиденных изменений в спросе или задержках поставки. (рисунок 5, 12)

Оценка Современных Подходов:

- **ЛТ:** Эффективность ЛТ проявляется в минимизации затрат на хранение и уменьшении незавершенного производства. Но этот подход требует строгого соблюдения сроков поставки и высокого качества процессов, что может быть рискованным в нестабильной среде. (Рисунок 7, 10, 14)
- **ИТ-решения:** Модели, основанные на информационных технологиях, предлагают улучшенное прогнозирование и оптимизацию запасов. Они могут значительно повысить эффективность, но требуют больших начальных инвестиций и навыков в области аналитики данных.

Кейс-Стади и Практические Примеры:

- **Примеры из Производства:** Иллюстрация применения ЛТ в автомобильной промышленности показывает, как компании могут сократить затраты на хранение, но при этом сталкиваются с проблемой уязвимости цепочки поставок.
- **Примеры из Розничной Торговли:** Примеры использования ИТ-решений в розничной торговле демонстрируют, как данные о продажах

могут использоваться для динамической адаптации уровней запасов, но требуют продвинутой инфраструктуры для сбора и анализа данных.

Влияние на Финансовые Показатели и Операционную Эффективность:

- **Финансовые Показатели:** Модели, такие как EOQ и ЛТ, могут существенно влиять на операционные затраты и капитал, замороженный в запасах. Эффективное управление запасами напрямую связано с сокращением операционных расходов и улучшением общей финансовой производительности.
- **Операционная Эффективность:** Модели, основанные на точном прогнозировании и адаптивности, могут значительно повысить операционную гибкость и удовлетворенность клиентов, но также требуют глубокого понимания рыночной динамики и эффективного управления данными.

Вывод: Анализ различных моделей управления запасами показывает, что существует потребность в новой гибридной модели, которая объединила бы простоту традиционных подходов с адаптивностью и точностью современных ИТ-решений. Такая модель должна быть способна эффективно функционировать в условиях быстро меняющегося спроса, минимизировать риски, связанные с неопределенностью поставок, и быть экономически доступной для различных видов бизнеса. В дальнейшем мы попытаемся разработать и тестировать такую модель, используя Python, для проверки её практической применимости и эффективности.

4. Разработка и тестирование новой модели управления запасами

Создание "новой" модели управления запасами не обязательно требует изобретения совершенно новых концепций с нуля. Вместо этого, новизна может достигаться путем уникального сочетания существующих подходов и методов, которые ранее не использовались вместе или применялись в различных контекстах.

Будет разработана комплексная программа для управления запасами, объединяющая методы прогнозирования спроса на основе исторических данных с моделью EOQ (Economic Order Quantity) и JIT (Just-In-Time). Эта программа будет направлена на повышение эффективности управления запасами за счет точного прогнозирования и оптимизации размеров заказов.

Ключевые аспекты предстоящего проекта:

1. Анализ Исторических Данных для Прогнозирования Спроса:

- Планируется использовать продвинутые методы анализа временных рядов, включая скользящие средние, для выявления тенденций спроса на основе прошлых данных о продажах.
- Этот подход позволит более точно прогнозировать будущий спрос, адаптируясь к рыночным изменениям.

2. Применение Модели EOQ для Оптимизации Заказов:

- Интегрируем результаты анализа спроса с моделью EOQ для определения оптимального размера заказа.
- Это позволит сократить общие затраты на закупку и хранение запасов, а также улучшить оборачиваемость товаров.
- (Рисунок 2, 4, 11)

3. Линейная Регрессия для Прогнозирования Будущего Спроса:

- Разработка модели линейной регрессии для оценки спроса на следующие 90 дней, что улучшит стратегическое планирование закупок.

4. Анализ Ценовой Эластичности Спроса:

- Выявление зависимости спроса от ценовой политики позволит более эффективно управлять продажами и маржой.

5. Исследование Сезонных Колебаний в Продажах:

- Определение периодов повышенного и пониженного спроса для адаптации стратегий управления запасами и проведения акций.

6. Реализация JIT (Just-In-Time) Подхода:

- Применение модели ЛТ для оптимизации уровней запасов, основываясь на актуальном спросе и пороговых значениях.
- (Рисунок 7, 10, 14)

7. Наглядная Визуализация Данных:

- Создание информативных графиков для демонстрации результатов анализа, включая отдельные графики продаж по каждому товару и прогнозы спроса.

Ключевые инструменты реализации:

Pandas

- **Назначение:** Является основной библиотекой для обработки и анализа данных. Это мощный инструмент, обеспечивающий удобные структуры данных для манипуляции с числовыми таблицами и временными рядами.
- **Применение в проекте:** Будет использоваться для загрузки, очистки, преобразования и анализа исторических данных о продажах. Это ключевой элемент для подготовки данных перед применением аналитических моделей.

NumPy

- **Назначение:** Предоставляет поддержку для больших многомерных массивов и матриц, включая широкий набор математических функций для работы с этими массивами.
- **Применение в проекте:** Будет использоваться для всех основных математических расчетов, включая генерацию фиктивных данных для демонстрации модели.

Matplotlib

- **Назначение:** Стандартная библиотека для визуализации данных в Python. Предлагает широкий спектр инструментов для создания статических, анимированных и интерактивных визуализаций.

- **Применение в проекте:** Основной инструмент для создания графиков и диаграмм, которые помогут наглядно представить анализ продаж, сезонные колебания, результаты прогнозов спроса и другие ключевые метрики.

Sklearn (Scikit-learn)

- **Назначение:** Одна из основных библиотек машинного обучения в Python. Предлагает простые и эффективные инструменты для анализа данных и майнинга.
- **Применение в проекте:** Использование линейной регрессии из этой библиотеки позволит нам разработать надежные модели для прогнозирования будущего спроса.

Statsmodels

- **Назначение:** Предоставляет классы и функции для оценки множества различных статистических моделей, а также для проведения статистических тестов и исследования данных.
- **Применение в проекте:** Будет использоваться для сезонного разложения временных рядов, что позволит точно анализировать сезонные колебания в продажах и понимать, как эти колебания могут повлиять на спрос.

Интеграция Инструментов

Все эти инструменты будут интегрированы для создания комплексной системы управления запасами. Pandas и NumPy обеспечат эффективную работу с данными, Sklearn и Statsmodels помогут в анализе и прогнозировании, а Matplotlib визуализирует все полученные результаты. Это сочетание инструментов предоставит нам глубокое понимание динамики продаж и позволит оптимизировать процессы управления запасами.

Эта модель будет считаться новой, так как она интегрирует традиционные методы оптимизации запасов с аналитическими подходами,

основанными на данных. Такой гибридный подход позволяет не только учитывать исторические тенденции спроса, но и адаптировать стратегии заказа к изменяющимся рыночным условиям. Это может быть особенно полезно для предприятий, стремящихся оптимизировать управление запасами в условиях непостоянного спроса.

Разработка, описание всех компонентов программы:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from statsmodels.tsa.seasonal import seasonal_decompose
```

Эта часть кода представляет собой импорт необходимых библиотек для реализации аналитической программы на Python, специализирующейся на анализе и прогнозировании продаж:

1. **import pandas as pd:** Библиотека Pandas используется для удобной работы с данными. Она предоставляет структуры данных, такие как DataFrame, которые позволяют обрабатывать и анализировать табличные данные эффективно. Pandas особенно полезен для чтения, очистки, обработки и агрегации данных.
2. **import numpy as np:** Библиотека NumPy предоставляет поддержку больших многомерных массивов и матриц, включая широкий набор математических функций для работы с этими массивами. Это основной инструмент для научных вычислений в Python и используется для математических расчетов и генерации фиктивных данных в вашем проекте.
3. **import matplotlib.pyplot as plt:** Matplotlib – это библиотека для визуализации данных в Python. Она используется для создания статических, анимированных и интерактивных визуализаций в Python. В

вашем проекте **matplotlib** используется для отрисовки графиков и диаграмм, что позволяет визуально представить анализируемые данные.

4. **from sklearn.linear_model import LinearRegression**: Этот импорт из библиотеки scikit-learn (sklearn) обеспечивает доступ к классу **LinearRegression**, который используется для создания и работы с моделью линейной регрессии. В вашем случае это может быть применено для прогнозирования будущего спроса на основе исторических данных.
5. **from statsmodels.tsa.seasonal import seasonal_decompose**: Этот импорт из библиотеки Statsmodels предоставляет функционал для сезонного разложения временных рядов. Функция **seasonal_decompose** позволяет анализировать и визуализировать сезонные компоненты временных рядов, что может быть использовано для выявления сезонных колебаний в продажах.

Эти библиотеки являются фундаментом для разработки сложной аналитической программы, предназначенной для анализа данных продаж, прогнозирования спроса, а также визуализации результатов анализа.

```
# Функция для расчета EOQ
def calculate_eoq(demand, order_cost, holding_cost):
    return np.sqrt((2 * demand * order_cost) / holding_cost)
```

Эта часть кода определяет функцию **calculate_eoq**, которая реализует расчет EOQ (Economic Order Quantity - Экономически Оптимальный Размер Заказа). EOQ - это классическая модель управления запасами, используемая для определения оптимального количества единиц товара, которое следует заказывать для минимизации общих затрат, связанных с хранением и заказом товаров. Вот как функция работает:

- **Аргументы функции:**
 - **demand**: Общий спрос или количество проданного товара за определенный период времени.
 - **order_cost**: Стоимость размещения одного заказа. Это может включать затраты на обработку заказа, доставку, прием товара и прочее.

- **holding_cost**: Стоимость хранения одной единицы товара в течение года. Включает в себя затраты на складское помещение, страхование, убытки от порчи и устаревания и т.д.
- **Тело функции:**
 - **np.sqrt((2 * demand * order_cost) / holding_cost)**: Формула EOQ представляет собой корень квадратный из **(2 * спрос * стоимость заказа) / стоимость хранения**. Эта формула помогает найти количество единиц товара для заказа, которое минимизирует сумму затрат на заказ и хранение.
- **Возврат значения:**
 - Функция возвращает значение EOQ, которое является количеством единиц продукта, которое следует заказывать каждый раз, когда нужно делать заказ, чтобы минимизировать общие затраты.

Применение этой функции в контексте управления запасами позволяет более эффективно управлять инвентаризацией, находя баланс между частыми небольшими заказами (которые повышают затраты на обработку заказа) и редкими большими заказами (которые увеличивают затраты на хранение).

```
# Функция для имитации JIT подхода
def simulate_jit(sales_data, threshold):
    jit_points = sales_data[sales_data <= threshold]
    return jit_points
```

Эта часть кода определяет функцию **simulate_jit**, которая используется для имитации подхода JIT (Just-In-Time - как раз вовремя) в контексте управления запасами. JIT - это стратегия управления запасами, цель которой - уменьшить объем запасов, находящихся на складе, и оптимизировать заказы таким образом, чтобы товары поступали непосредственно перед их использованием или продажей. Вот как функция работает:

- **Аргументы функции:**
 - **sales_data**: Данные о продажах, обычно в формате временного ряда.
 - **threshold**: Пороговое значение, которое используется для определения моментов времени, когда запасы уменьшаются до критического уровня.
- **Тело функции:**
 - **sales_data[sales_data <= threshold]**: Эта строка кода выбирает из временного ряда продаж все точки, где объем продаж равен или меньше порогового значения. Это представляет собой моменты, когда запасы уменьшаются до уровня, требующего нового заказа.
- **Возврат значения:**

- Функция возвращает **jlt_points**, которые являются индексами или временными метками, когда уровень запасов достигает или падает ниже порогового значения. Эти точки могут быть использованы для запуска новых заказов в соответствии с принципами JIT.

Применение этой функции помогает компаниям уменьшить затраты на хранение запасов, оптимизировать объемы производства и улучшить управление цепочками поставок. В контексте нашей программы она позволяет динамически реагировать на изменения в спросе и поддерживать необходимый уровень запасов, минимизируя при этом избыточные запасы и связанные с ними затраты.

```
# Визуализация прогноза будущего спроса и сезонного эффекта  
future_dates = pd.date_range(start=data["Дата"].iloc[-1] +  
pd.Timedelta(days=1), periods=90, freq='D')
```

Этот кусок кода служит для создания диапазона будущих дат, который будет использоваться в визуализации прогнозов продаж.

1. **future_dates = pd.date_range(...)**: Здесь мы используем функцию **date_range** из библиотеки Pandas, чтобы сгенерировать последовательность дат. Это полезно для создания временных отметок, на которые распространяется наш прогноз.
2. **start=data["Дата"].iloc[-1] + pd.Timedelta(days=1)**: Определяет начальную точку временного ряда для будущих прогнозов. Мы начинаем с дня, следующего за последней датой в наших исходных данных. **data["Дата"].iloc[-1]** берет последнюю дату из существующего набора данных, а **pd.Timedelta(days=1)** добавляет к ней один день.
3. **periods=90**: Определяет количество дней в будущем, для которых мы хотим сгенерировать даты. В данном случае выбран период в 90 дней, что позволяет нам сформировать прогноз на три месяца вперед.
4. **freq='D'**: Указывает частоту генерации дат, где **'D'** обозначает дневную частоту. Это означает, что даты будут сгенерированы для каждого дня в выбранном периоде.

В результате выполнения этой строки кода мы получим объект **future_dates**, который содержит ряд последовательных дневных дат, начиная с дня после последней даты в исходных данных и продолжающихся на 90 дней вперед. Этот временной ряд будет использоваться для отображения прогнозируемых значений продаж в будущем при визуализации результатов анализа.

```
# Функция для анализа ценовой эластичности спроса
def price_elasticity_analysis(sales_data, price_data):
    sales_data = sales_data.to_numpy().reshape(-1, 1)
    price_data = price_data.to_numpy().reshape(-1, 1)
    regression_model = LinearRegression().fit(price_data,
sales_data)
    elasticity = regression_model.coef_[0][0]
    return elasticity
```

Эта функция **price_elasticity_analysis** предназначена для оценки ценовой эластичности спроса на товары, что является ключевым аспектом для понимания влияния изменения цен на объем продаж.

1. **Входные параметры:** Функция принимает два аргумента: **sales_data** (данные о продажах) и **price_data** (данные о ценах). Эти два набора данных являются основой для расчета эластичности спроса.
2. **Преобразование данных:**
 - **sales_data.to_numpy().reshape(-1, 1):** Данные о продажах преобразуются в NumPy массив и переформатируются в формат столбца. Опция **-1** в **reshape** означает, что размерность этого измерения будет автоматически определена, исходя из длины массива.
 - Аналогично преобразуется **price_data**.
3. **Построение модели линейной регрессии:**
 - **LinearRegression().fit(price_data, sales_data):** Используя библиотеку **scikit-learn**, мы обучаем модель линейной регрессии, где цены выступают в качестве независимой переменной (предиктора), а продажи - в качестве зависимой переменной (ответа). Модель пытается установить линейную зависимость между ценой и объемом продаж.
4. **Расчет эластичности:**
 - **elasticity = regression_model.coef_[0][0]:** После обучения модели коэффициент регрессии (угол наклона линии регрессии) определяет эластичность спроса. Он показывает, насколько процентов изменится спрос, если цена изменится на один процент.
5. **Возвращаемое значение:**
 - Функция возвращает значение эластичности. Это ключевой показатель в экономике, который помогает понять, является ли продукт эластичным (большое изменение спроса при малом изменении цены) или неэластичным (небольшое или отсутствие изменения спроса при изменении цены).

Эта функция полезна для бизнес-анализа, так как она помогает принимать обоснованные решения о ценообразовании, основываясь на понимании реакции спроса на изменение цен.

```
# Функция для сезонного анализа продаж
def seasonal_analysis(sales_data):
    result = seasonal_decompose(sales_data, model='additive',
period=30)
    return result.seasonal
```

Эта функция, **seasonal_analysis**, выполняет сезонный анализ данных о продажах, что позволяет выявить сезонные колебания в поведении спроса.

1. **Входной параметр: sales_data** – это временной ряд данных о продажах, который функция анализирует для выявления сезонных колебаний.
2. **Процесс анализа:**
 - **seasonal_decompose(sales_data, model='additive', period=30):** Для анализа используется функция **seasonal_decompose** из библиотеки **statsmodels**. Эта функция разлагает временной ряд на три составляющие: тренд, сезонность и случайные колебания.
 - **model='additive'** указывает на то, что временной ряд предполагается аддитивным, т.е. его компоненты (тренд, сезонность, остатки) складываются для образования исходных данных.
 - **period=30** определяет сезонный период. В данном случае это 30 дней, что подразумевает месячную сезонность.
3. **Возвращаемое значение:**
 - Функция возвращает только сезонную компоненту (**result.seasonal**) разложения временного ряда. Эта компонента показывает повторяющиеся паттерны в данных о продажах за определенный период (в данном случае, за месяц).

Применение этой функции важно для понимания сезонных влияний на продажи. Например, определенные товары могут продаваться лучше в определенные сезоны (летом, зимой и т.д.). Это позволяет компаниям планировать свои запасы, маркетинговые кампании и ценовую политику в соответствии с сезонными колебаниями спроса.

```
# Прогноз будущего спроса
def demand_forecast(sales_data):
    model = LinearRegression()
    days = np.array(range(len(sales_data))).reshape(-1, 1)
    model.fit(days, sales_data)
    # Изменение размера future_days на 90 дней
    future_days = np.array(range(len(sales_data),
len(sales_data) + 90)).reshape(-1, 1)
    future_forecast = model.predict(future_days)
    return future_forecast
```

Эта функция, **demand_forecast**, используется для прогнозирования будущего спроса на продукцию на основе исторических данных о продажах.

1. **Входной параметр:**

- **sales_data**: Это временной ряд исторических данных о продажах, который используется для обучения модели линейной регрессии.

2. **Процесс прогнозирования:**

- Создание модели: Используется **LinearRegression()** из библиотеки **sklearn**, которая представляет собой модель линейной регрессии.
- Подготовка данных: **days** представляет собой массив дней, соответствующих каждому наблюдению в **sales_data**. Этот массив преобразуется в формат, подходящий для модели линейной регрессии (**reshape(-1, 1)**).
- Обучение модели: **model.fit(days, sales_data)** обучает модель линейной регрессии на основе исторических данных о продажах, используя дни в качестве независимой переменной и данные о продажах в качестве зависимой переменной.
- Прогнозирование: **future_days** создает массив будущих дней для прогнозирования. В данном случае, он расширен до 90 дней после последнего дня в исторических данных. **model.predict(future_days)** использует обученную модель для прогнозирования продаж на эти будущие дни.

3. **Возвращаемое значение:**

- **future_forecast**: Массив прогнозируемых значений продаж на следующие 90 дней.

Эта функция полезна для планирования закупок и управления запасами, так как предоставляет оценку будущего спроса. Она позволяет бизнесу принимать информированные решения о том, сколько товаров заказать и когда, чтобы максимально эффективно удовлетворить предполагаемый спрос при минимизации издержек на хранение запасов.

```
plt.figure(figsize=(20, 25))
```

Этот кусок кода относится к библиотеке **Matplotlib** и используется для настройки размера графического окна, в котором будут отображаться все последующие графики.

• **plt.figure(figsize=(20, 25)):**

- **plt.figure()** – это функция из библиотеки **Matplotlib**, которая создает новое графическое окно (или фигуру) для отображения графиков.

- **figsize=(20, 25)** – это параметр, который задает размер создаваемого графического окна. Здесь размер устанавливается равным 20x25 дюймов. Эти значения определяют ширину и высоту окна соответственно. Выбор размера окна зависит от количества и сложности визуализируемых данных, а также от того, как детально вы хотите их отображать.

Этот код является важным начальным этапом в создании визуализаций, так как правильный выбор размера фигуры помогает обеспечить, чтобы все графики были четко видны и информативны. В контексте вашей программы, где предполагается создание нескольких графиков для анализа продаж, оптимальный размер окна гарантирует, что каждый график будет достаточно большим и легко читаемым.

```
# Общий график продаж всех товаров
plt.subplot(5, 2, 1)
total_sales = pd.Series(dtype=float)
for column in product_columns:
    total_sales = total_sales.add(data[column], fill_value=0)
    plt.plot(data['Дата'], data[column], label=column)
plt.title('Общий график продаж всех товаров')
plt.xlabel('Дата')
plt.ylabel('Продажи')
plt.legend()
```

Этот код создает общий график продаж для всех товаров в рамках анализа данных о продажах.

1. **plt.subplot(5, 2, 1)**: Эта функция из Matplotlib используется для размещения нескольких графиков в одном окне. Аргументы **5, 2, 1** означают:
 - Создание сетки графиков размером 5 строк на 2 столбца.
 - Выбор первого подграфика в этой сетке для последующего рисования.
2. **total_sales = pd.Series(dtype=float)**: Создает пустой объект Series с помощью библиотеки pandas, предназначенный для хранения данных о суммарных продажах.
3. Цикл **for**:
 - **for column in product_columns**: Перебирает каждый столбец в списке **product_columns**, который содержит названия столбцов продаж для каждого товара.
 - **total_sales.add(data[column], fill_value=0)**: Складывает значения продаж текущего товара с общими продажами (**total_sales**). Если в **total_sales** еще нет соответствующих индексов, используется **fill_value=0** для заполнения пропусков.
4. Визуализация данных:

- **plt.plot(data['Дата'], data[column], label=column)**: Рисует график продаж для каждого товара. Здесь **data['Дата']** обозначает ось X (даты), **data[column]** - ось Y (значения продаж), а **label=column** добавляет метку к каждой линии графика для идентификации.
- 5. **plt.title('Общий график продаж всех товаров')**: Устанавливает заголовок для подграфика.
- 6. **plt.xlabel('Дата')** и **plt.ylabel('Продажи')**: Устанавливают подписи для осей X и Y.
- 7. **plt.legend()**: Добавляет легенду на график, используя метки, заданные в **label=column**.

Этот график позволяет наглядно сравнить динамику продаж различных товаров во времени, а также общую тенденцию продаж.

```
# Обработка данных для каждого товара
for i, column in enumerate(product_columns, start=2):
    data[f'{column} Moving Average'] =
data[column].rolling(window=7).mean().bfill()
    data[f'{column} EOQ'] = calculate_eoq(data[column].sum(),
order_cost, holding_cost)
    jit_points = simulate_jit(data[column], jit_threshold)
    plt.subplot(5, 2, i)
    plt.plot(data['Дата'], data[column], label=f'{column}
Продажи')
    plt.plot(data['Дата'], data[f'{column} Moving Average'],
label=f'{column} Скользящее среднее', linestyle='--')
    plt.scatter(data['Дата'][jit_points.index],
data[column][jit_points.index], color='red', label='JIT точки
заказа')
    plt.title(f'График продаж для {column}')
    plt.xlabel('Дата')
    plt.ylabel('Продажи')
    plt.legend()
```

Данный код выполняет анализ и визуализацию продаж для каждого товара.

1. Цикл For для Обработки Данных:

- **for i, column in enumerate(product_columns, start=2)**: Цикл перебирает каждый столбец в списке **product_columns**, содержащем названия столбцов продаж каждого товара. **enumerate** также возвращает индекс **i**, начиная с 2, для каждого столбца.

2. Расчет Скользящего Среднего:

- **data[f'{column} Moving Average'] = data[column].rolling(window=7).mean().bfill()**:

- **rolling(window=7)**: Применяет скользящее окно шириной 7 дней к данным продаж.
- **mean()**: Вычисляет среднее значение в каждом окне.
- **bfill()**: Заполняет пропущенные значения в начале временного ряда (где скользящее окно неполное) последующими значениями.

3. Расчет EOQ (Economic Order Quantity):

- **data[f'{column} EOQ'] = calculate_eoq(data[column].sum(), order_cost, holding_cost)**: Вызывает функцию **calculate_eoq**, передавая сумму продаж, стоимость заказа (**order_cost**) и затраты на хранение (**holding_cost**) для расчета оптимального размера заказа.

4. Симуляция JIT (Just-In-Time) Подхода:

- **jit_points = simulate_jit(data[column], jit_threshold)**: Вызывает функцию **simulate_jit**, передавая данные продаж и пороговое значение (**jit_threshold**) для определения точек заказа по методу JIT.

5. Визуализация Данных:

- **plt.subplot(5, 2, i)**: Располагает график в соответствующем месте в сетке из 5 строк и 2 столбцов.
- **plt.plot(...)**: Создает график продаж для каждого товара.
- **plt.plot(..., linestyle='--')**: Добавляет график скользящего среднего с пунктирной линией.
- **plt.scatter(...)**: Добавляет точки на график в местах, где должны быть сделаны заказы по методу JIT.

6. Добавление Заголовков и Легенд:

- **plt.title(...), plt.xlabel(...), plt.ylabel(...)**: Устанавливают заголовок, подписи осей X и Y соответственно.
- **plt.legend()**: Добавляет легенду для лучшего понимания графика.

Этот анализ позволяет лучше понять динамику продаж каждого товара, влияние сезонности (с помощью скользящего среднего) и определить оптимальные моменты для заказов.

```
plt.subplot(5, 2, 5)
plt.plot(data['Дата'], total_sales, label='Суммарные продажи')
plt.title('График суммарных продаж')
plt.xlabel('Дата')
plt.ylabel('Продажи')
plt.legend()
```

Данный блок кода создает график суммарных продаж всех товаров. Давайте подробно рассмотрим его компоненты:

1. **Определение Размещения Графика в Сетке:**

- **plt.subplot(5, 2, 5):** Указывает, что график будет расположен в пятой ячейке сетки, состоящей из 5 строк и 2 столбцов. Это пятая позиция в сетке, обычно расположенная в первой колонке третьего ряда.

2. **Построение Графика Суммарных Продаж:**

- **plt.plot(data['Дата'], total_sales, label='Суммарные продажи'):** Эта строка строит линейный график, где:
 - **data['Дата']** используется в качестве оси X (горизонтальной).
 - **total_sales** представляет суммарные продажи всех товаров для оси Y (вертикальной).
 - **label='Суммарные продажи'** добавляет метку к линии графика для последующего отображения в легенде.

3. **Добавление Заголовка и Подписей Осей:**

- **plt.title('График суммарных продаж'):** Добавляет заголовок над графиком.
- **plt.xlabel('Дата')** и **plt.ylabel('Продажи'):** Устанавливают подписи для осей X и Y соответственно.

4. **Добавление Легенды:**

- **plt.legend():** Добавляет легенду на график. В данном случае будет отображаться метка "Суммарные продажи", ассоциируемая с линией графика.

Этот график важен для общего понимания тенденций продаж за весь период. Он дает наглядное представление о динамике общих продаж всех товаров, что полезно для выявления общих тенденций, например, роста или снижения спроса в определенные периоды.

```
# Прогноз для Товара А
sales_data_a = data['Товар А Продажи']
future_forecast_a = demand_forecast(sales_data_a)
plt.subplot(5, 2, 6)
plt.plot(data['Дата'], sales_data_a, label='Фактические
продажи')
plt.plot(future_dates, future_forecast_a, label='Прогноз',
linestyle='--')
plt.title('Прогноз спроса для Товара А')
plt.xlabel('Дата')
plt.ylabel('Продажи')
plt.legend()
```

```

# Прогноз для Товара В
sales_data_b = data['Товар В Продажи']
future_forecast_b = demand_forecast(sales_data_b)
plt.subplot(5, 2, 7)
plt.plot(data['Дата'], sales_data_b, label='Фактические
продажи')
plt.plot(future_dates, future_forecast_b, label='Прогноз',
linestyle='--')
plt.title('Прогноз спроса для Товара В')
plt.xlabel('Дата')
plt.ylabel('Продажи')
plt.legend()

# Прогноз для Товара С
sales_data_c = data['Товар С Продажи']
future_forecast_c = demand_forecast(sales_data_c)
plt.subplot(5, 2, 8)
plt.plot(data['Дата'], sales_data_c, label='Фактические
продажи')
plt.plot(future_dates, future_forecast_c, label='Прогноз',
linestyle='--')
plt.title('Прогноз спроса для Товара С')
plt.xlabel('Дата')
plt.ylabel('Продажи')
plt.legend()

```

Этот блок кода содержит создание графиков прогнозов будущего спроса для трех различных товаров (А, В и С). Давайте разберемся, как они реализованы:

1. Прогнозирование и Визуализация для Каждого Товара:

- Сначала определяется исходный набор данных продаж для каждого товара (**sales_data_a**, **sales_data_b**, **sales_data_c**).
- Далее вызывается функция **demand_forecast**, которая применяет линейную регрессию к данным продаж, чтобы сгенерировать прогноз на будущие 90 дней. Это делается для каждого товара отдельно.

2. Создание Графиков Прогнозов:

- Используется **plt.subplot** для определения позиции каждого графика в общей сетке. Например, **plt.subplot(5, 2, 6)** обозначает шестую позицию для товара А.
- Для каждого товара строится две линии на графике:
 - **plt.plot(data['Дата'], sales_data_x, label='Фактические продажи')** отображает реальные продажи по датам.
 - **plt.plot(future_dates, future_forecast_x, label='Прогноз', linestyle='--')** показывает прогнозируемые продажи на будущие даты с пунктирной линией.

3. Добавление Заголовков, Подписей и Легенды:

- Каждый график снабжается заголовком (**plt.title**), обозначающим, для какого товара представлен прогноз.
- Для осей X и Y добавляются подписи (**plt.xlabel** и **plt.ylabel**).
- Легенда (**plt.legend()**) включается для различения линий фактических продаж и прогнозов.

Эти графики предоставляют ценную информацию о потенциальном будущем спросе на каждый товар, что помогает в планировании закупок и управлении запасами. Они позволяют наглядно оценить, как могут изменяться продажи товаров в ближайшие три месяца.

```
# Вывод дополнительной аналитической информации и рекомендаций
print("Дополнительный анализ и рекомендации:")
for column in product_columns:
    total_sales = data[column].sum()
    average_price = data[f'{column[:-8]} Цена'].mean()
    eoq = data[f'{column} EOQ'].iloc[0]
    jit_points_count = len(simulate_jit(data[column],
jit_threshold))
```

Этот блок кода представляет собой сегмент для вывода дополнительного анализа и рекомендаций на основе анализа данных продаж.

1. Начало Вывода:

- **print("Дополнительный анализ и рекомендации:")** - это просто строка, которая вводит секцию вывода анализа и рекомендаций.

2. Итерация по Товарам:

- Цикл **for column in product_columns:** перебирает каждый столбец продаж для различных товаров (Товар А, Товар В, Товар С). Здесь **column** представляет текущий столбец продаж для конкретного товара.

3. Расчет и Вывод Ключевых Метрик:

- **total_sales = data[column].sum()** вычисляет общее количество продаж для каждого товара.
- **average_price = data[f'{column[:-8]} Цена'].mean()** рассчитывает среднюю цену каждого товара, используя соответствующий столбец цен.
- **eoq = data[f'{column} EOQ'].iloc[0]** извлекает рассчитанное значение EOQ (Экономически Оптимального Размера Заказа) для каждого товара.
- **jit_points_count = len(simulate_jit(data[column], jit_threshold))** определяет количество точек заказа по методу JIT (Just-In-Time), когда продажи опускаются ниже заданного порога **jit_threshold**.

4. Вывод Расчетов:

- Для каждого товара выводится серия строк с информацией о общих продажах, средней цене, ЕОQ, количестве ЛТ точек заказа и других ключевых показателях.
- Это обеспечивает полное представление о производительности каждого товара и необходимых стратегиях управления запасами.

Этот анализ и рекомендации помогают в понимании динамики продаж и влиянии цен на спрос, а также в определении оптимальных стратегий закупок и управления запасами для каждого из товаров.

```
# Рекомендации по закупке
recommended_order_frequency = round(365 / (eoq / (total_sales /
365)))
```

Эта часть кода используется для расчета рекомендуемой частоты заказов, основываясь на оптимальном размере заказа (ЕОQ). Давайте разберемся, как это работает:

- **recommended_order_frequency = round(365 / (eoq / (total_sales / 365)))**:
 - **eoq / (total_sales / 365)**: Сначала мы берем ЕОQ и делим его на среднесуточные продажи (это получается путем деления общего количества продаж **total_sales** на 365 дней). Этот расчет дает нам приблизительное количество дней между заказами, необходимых для поддержания оптимального уровня запасов, основываясь на ЕОQ.
 - **365 / (eoq / (total_sales / 365))**: Затем мы делим 365 дней на полученное количество дней между заказами. Это дает нам примерное количество заказов, которое необходимо сделать за год, чтобы поддерживать запасы на оптимальном уровне.
 - **round(...)**: И, наконец, мы округляем это значение до ближайшего целого числа, так как количество заказов не может быть дробным.

Этот расчет помогает определить, как часто компании следует делать заказы для каждого товара, чтобы минимизировать затраты на хранение и заказы, соблюдая при этом необходимый уровень запасов в соответствии с моделью ЕОQ.

```
# Расчет коэффициента оборачиваемости запасов
average_inventory = data[column].mean()
turnover_rate = total_sales / average_inventory
```

Этот участок кода относится к расчету коэффициента оборачиваемости запасов. Давайте разберем его подробнее:

1. **average_inventory = data[column].mean()**:

- Здесь мы рассчитываем среднее количество товаров на складе за рассматриваемый период. Это делается путем вычисления среднего значения (**mean()**) продаж для каждого товара (**column**) за весь период. Это значение представляет собой приблизительное среднее количество единиц товара, которое необходимо иметь на складе, чтобы удовлетворить текущий уровень спроса.

2. **turnover_rate = total_sales / average_inventory:**

- Здесь мы рассчитываем коэффициент оборачиваемости запасов. Это соотношение общего объема продаж (**total_sales**) к среднему запасу (**average_inventory**).
- Коэффициент оборачиваемости запасов показывает, сколько раз за период запасы полностью продавались и восполнялись. Это важный показатель эффективности управления запасами, поскольку он указывает на скорость, с которой товар обращается (продается и заменяется новым).

В контексте управления запасами высокий коэффициент оборачиваемости обычно является желательным, так как он указывает на хорошее соответствие запасов спросу и минимальное простаивание капитала в непроданных товарах. Однако слишком высокий коэффициент может также указывать на потенциальный риск дефицита, что подразумевает необходимость более тонкой настройки системы управления запасами.

```
sales_data = data[column]
price_data = data[f"{column[:-8]} Цена"]
```

Этот сегмент кода предназначен для извлечения данных о продажах и ценах для каждого товара в рамках цикла, который проходит по всем товарам.

1. **sales_data = data[column]:**

- Здесь мы извлекаем данные о продажах для конкретного товара. **column** является переменной из цикла **for**, которая перебирает все столбцы, связанные с продажами каждого товара. Например, если **column** равно 'Товар А Продажи', то **sales_data** будет содержать данные о продажах Товара А.

2. **price_data = data[f"{column[:-8]} Цена"]:**

- В этой строке кода мы извлекаем данные о ценах для соответствующего товара. Ключевой момент здесь — использование среза строки **column[:-8]** для получения названия товара без части 'Продажи' и добавления к нему 'Цена' для получения соответствующего столбца цен. Например, если **column** равно 'Товар А Продажи', **column[:-8]** превратится в 'Товар А', и к этому будет добавлено 'Цена', в результате чего мы получим 'Товар А Цена' — название столбца с ценами для Товара А.

Эти строки кода являются подготовительным этапом для последующего анализа данных о продажах и ценах, таких как расчет ценовой эластичности спроса, а также для других аналитических и визуализационных целей. Это позволяет работать с конкретными данными каждого товара в рамках более широкого анализа.

```
total_sales = sales_data.sum()
average_price = price_data.mean()
eoq = calculate_eoq(total_sales, order_cost, holding_cost)
jit_points = simulate_jit(sales_data, jit_threshold)
```

Этот фрагмент кода выполняет ключевые расчеты и анализ для каждого товара в цикле.

1. **total_sales = sales_data.sum():**

- Здесь мы рассчитываем общее количество продаж для каждого товара, суммируя все значения в столбце **sales_data**, который содержит данные о продажах для конкретного товара. Это дает нам понимание общего объема продаж за рассматриваемый период.

2. **average_price = price_data.mean():**

- Эта строка вычисляет среднюю цену товара, усредняя все значения в столбце **price_data**, который содержит данные о ценах на конкретный товар. Средняя цена важна для понимания ценового положения товара на рынке и может использоваться для дальнейшего анализа, например, при расчете точки безубыточности.

3. **eoq = calculate_eoq(total_sales, order_cost, holding_cost):**

- Здесь мы вызываем функцию **calculate_eoq**, чтобы рассчитать EOQ (Economic Order Quantity - Экономически обоснованный объем заказа) для каждого товара. Функция принимает общие продажи (**total_sales**), стоимость заказа (**order_cost**) и стоимость хранения (**holding_cost**) в качестве аргументов. Результатом является оптимальный объем заказа, который минимизирует общие затраты на заказ и хранение запасов.

4. **jit_points = simulate_jit(sales_data, jit_threshold):**

- Эта строка использует функцию **simulate_jit**, чтобы определить точки заказа товара по методу JIT (Just-In-Time - "точно в срок"). Функция принимает данные о продажах (**sales_data**) и пороговое значение (**jit_threshold**). Возвращаемым результатом являются точки, где уровень продаж опускается ниже заданного порогового значения, что указывает на необходимость заказа товара.

Эти строки кода важны для сбора основных метрик, связанных с продажами и управлением запасами каждого товара, и предоставляют основу для более детального анализа и стратегического планирования.

```
elasticity = price_elasticity_analysis(sales_data, price_data)
seasonal_effect = seasonal_analysis(sales_data)
future_sales_forecast = demand_forecast(sales_data)
```

Эти строки кода выполняют важные функции анализа данных продаж:

1. **elasticity = price_elasticity_analysis(sales_data, price_data):**
 - Эта строка использует функцию **price_elasticity_analysis** для оценки ценовой эластичности спроса на товар. Функция принимает **sales_data** (данные о продажах) и **price_data** (данные о ценах). Ценовая эластичность помогает понять, как изменение цены товара может повлиять на его спрос и продажи. Коэффициент эластичности, получаемый в результате, показывает степень чувствительности спроса на изменение цены.
2. **seasonal_effect = seasonal_analysis(sales_data):**
 - Здесь применяется функция **seasonal_analysis** для анализа сезонных колебаний в продажах. Эта функция использует метод сезонного разложения для выявления и изучения сезонных трендов в данных продаж. Это позволяет определить, в какие периоды года продажи товара обычно увеличиваются или уменьшаются, что очень полезно для планирования запасов и маркетинговых кампаний.
3. **future_sales_forecast = demand_forecast(sales_data):**
 - В этой строке вызывается функция **demand_forecast** для прогнозирования будущего спроса на товар. Модель линейной регрессии используется для прогнозирования продаж на основе исторических данных. Результатом является прогноз продаж на следующие 90 дней, который может быть использован для более эффективного планирования закупок и управления запасами.

Каждая из этих функций вносит значительный вклад в комплексный анализ данных продаж, позволяя принимать обоснованные решения в области управления запасами и ценообразования.

```
print(
    f'{column}: \n'
    f' - Общие продажи: {total_sales} единиц \n'
    f' - Средняя цена: {average_price:.2f} \n'
    f' - ЕОQ (оптимальный размер заказа): {eoq:.2f} единиц \n'
```

```

    f' - JIT точек заказа: {jit_points_count} \n'
    f' - Рекомендуемая частота заказов:
{recommended_order_frequency} раз(a) в год\n'
    f' - Запас безопасности: {safety_stock} единиц \n'
    f' - Рекомендуемый уровень запасов до нового заказа: {eoq +
safety_stock} единиц \n'
    f' - Время выполнения заказа: {lead_time} дней\n'
    f' - Совет: Пересматривайте уровень запасов каждые
{lead_time} дней и дополняйте до уровня {eoq + safety_stock}
единиц при необходимости\n'
    f' - Коэффициент оборачиваемости запасов:
{turnover_rate:.2f} \n'
    f' - Точка безубыточности: {break_even_point:.0f} единиц
\n'
    f" - Сезонный эффект: максимальный влияние
{seasonal_effect.max():.2f}, минимальный
{seasonal_effect.min():.2f}\n"
)

```

Этот блок кода представляет собой команду **print**, которая выводит подробный анализ и рекомендации по каждому товару. Давайте подробнее рассмотрим каждый элемент этого вывода:

1. **Общие продажи:** Показывает общее количество единиц, проданных для каждого товара.
2. **Средняя цена:** Отображает среднюю цену за единицу товара.
3. **EOQ (Economic Order Quantity):** Это оптимальный размер заказа, который минимизирует общие затраты на заказ и хранение товара.
4. **ЛТ точек заказа:** Количество раз, когда уровень продаж опускался ниже установленного порога ЛТ, указывая на необходимость нового заказа.
5. **Рекомендуемая частота заказов:** Это количество раз в год, когда, исходя из EOQ, рекомендуется делать заказы, чтобы оптимизировать затраты.
6. **Запас безопасности:** Количество товара, рекомендуемое к хранению в дополнение к EOQ, чтобы обеспечить уровень обслуживания клиентов при непредвиденных колебаниях спроса.
7. **Рекомендуемый уровень запасов до нового заказа:** Это сумма EOQ и запаса безопасности, показывающая, сколько товара следует иметь на складе перед тем, как делать новый заказ.
8. **Время выполнения заказа:** Предполагаемое время, необходимое для получения нового заказа.
9. **Совет:** Рекомендация по управлению запасами, основанная на предполагаемом времени выполнения заказа и необходимом уровне запасов.

10. **Коэффициент оборачиваемости запасов:** Это соотношение общих продаж к среднему инвентарю, показывающее, насколько эффективно компания управляет своими запасами.
11. **Точка безубыточности:** Количество единиц товара, которое необходимо продать, чтобы покрыть все фиксированные и переменные издержки.
12. **Сезонный эффект:** Показывает максимальное и минимальное влияние сезонных колебаний на продажи товара.
- Каждый из этих элементов анализа предоставляет ценную информацию для принятия обоснованных решений в управлении запасами и ценообразовании.

Заключение Главы 2

В заключении второй главы мы подводим итоги нашего глубокого исследования методов оптимизации управления запасами и разработки новой, инновационной системы управления запасами. Основываясь на тщательном анализе и объединении лучших практик, наша цель была создать систему, которая не только оптимизирует процессы, но и повышает общую эффективность управления запасами.

Изучение и Анализ Методов Оптимизации Запасов:

- Были детально изучены различные модели управления запасами, включая модели EOQ (Economic Order Quantity) и JIT (Just-In-Time). Эти модели помогают минимизировать затраты, связанные с заказом и хранением запасов, а также оптимизировать процессы поставок.
- Анализ ценовой эластичности спроса дал понимание о влиянии изменения цен на спрос, что критически важно для стратегического ценообразования и максимизации прибыли.
- Также углубились в сезонный анализ продаж, выявляя ключевые временные рамки для увеличения или уменьшения запасов.

Применение Моделей на Практике:

- Реализация системы была осуществлена через интеграцию различных компонентов, включая прогнозирование спроса, анализ ценовой эластичности и управление запасами на основе данных о продажах.
- Использование Python и его библиотек (Pandas, NumPy, Matplotlib) позволило создать гибкую и мощную аналитическую платформу для обработки данных и визуализации результатов.

Создание Новой Системы Управления Запасами:

- Разработанная система интегрирует традиционные и новаторские подходы. Она обеспечивает баланс между эффективностью закупок и минимизацией издержек на хранение.
- Система способствует адаптации к меняющимся рыночным условиям, предоставляя бизнесу гибкие и эффективные инструменты для оптимизации управления запасами.
- Включение прогнозов и аналитики ценовой эластичности позволяет более точно реагировать на рыночные изменения, оптимизируя процессы управления запасами и повышая общую прибыльность.

В целом, интегрированная система представляет собой передовое решение, которое объединяет лучшие практики управления запасами, аналитику и технологии. Это не просто улучшение существующих систем, но и крупный шаг вперед в оптимизации управления запасами, обеспечивающий бизнесу значительные конкурентные преимущества.

Глава 3: Проектная часть

1. Проектирование модели управления запасами для конкретного предприятия:

Цель

В этом пункте мы сосредоточим наше внимание на разработке и применении нашей комплексной модели управления запасами к конкретному тестовому предприятию. Цель - демонстрация эффективности разработанной системы в условиях, максимально приближенных к реальной бизнес-среде.

Контекст и Описание Задачи:

Будет создан сценарий для тестового предприятия, имитирующего реальные бизнес-условия. Это предприятие будет иметь свою специфику, включая ассортимент товаров, объемы продаж и ценовые стратегии. В модели будут использоваться фиктивные данные, которые отражают возможные сценарии спроса и продаж, сезонные колебания, а также ценовые изменения.

Задачи и Методы:

1. Создание Фиктивного Предприятия: Будут определены характеристики тестового предприятия, включая типы товаров, их ценовые категории и продажные тренды.
2. Генерация Тестовых Данных: Используя методы анализа временных рядов и статистическое моделирование, будут сгенерированы тестовые данные, отражающие реалистичные продажные циклы и спрос.
3. Применение Разработанной Модели: Внедрим комплексную модель управления запасами для анализа и оптимизации запасов тестового предприятия. Это позволит нам наглядно продемонстрировать, как наша система может быть применена на практике для улучшения эффективности управления запасами.
4. Анализ Результатов: Проведение анализа полученных данных, включая оценку эффективности управления запасами, прогнозирование спроса и адаптацию к изменениям рынка.

Ожидаемые Исходы:

Целью этой части является показать, как разработанная модель управления запасами может быть адаптирована к конкретным условиям предприятия, учитывая его уникальные характеристики и потребности. Необходимо продемонстрировать гибкость и эффективность системы в различных сценариях, подчеркивая ее практическую применимость в бизнесе.

Создание данных для конкретного предприятия:

```
data = pd.DataFrame({
    "Дата": pd.date_range(start="2021-01-01", periods=365,
    freq='D'),
    "Товар А Продажи": np.random.randint(140, 180, 365) +
    20*np.sin(np.linspace(0, 2*np.pi, 365)),
    "Товар А Цена": np.random.uniform(18, 22, 365),
    "Товар В Продажи": np.random.randint(90, 120, 365) +
    15*np.sin(np.linspace(0, 2*np.pi, 365)),
    "Товар В Цена": np.random.uniform(28, 32, 365),
    "Товар С Продажи": np.random.randint(190, 230, 365) +
    25*np.sin(np.linspace(0, 2*np.pi, 365)),
    "Товар С Цена": np.random.uniform(23, 27, 365)
})
```

```
product_columns = ['Товар А Продажи', 'Товар В Продажи', 'Товар С Продажи']
```

Этот фрагмент кода отвечает за создание фиктивных данных, которые могут быть использованы для моделирования и анализа продаж трех различных товаров в течение одного года. Вот как он работает:

1. **Создание DataFrame с помощью Pandas:**

- **pd.DataFrame({...})**: Создает DataFrame, который является двумерной табличной структурой данных с метками для строк и столбцов. Это основная структура данных в Pandas и используется для хранения и манипулирования набором данных.

2. **Генерация временного ряда:**

- **"Дата": pd.date_range(start="2021-01-01", periods=365, freq='D')**: Генерирует временной ряд дат, начиная с 1 января 2021 года, на протяжении 365 дней. Это обеспечивает временную ось для данных о продажах.

3. **Симуляция данных о продажах и ценах товаров:**

- Для каждого товара (А, В, С) создаются два столбца: один для продаж, другой для цен.
- Продажи генерируются как случайные числа в заданном диапазоне (**np.random.randint**) с добавлением синусоидальной волны (**np.sin**), имитирующей сезонные колебания.
- Цены генерируются как случайные числа в заданном диапазоне (**np.random.uniform**), имитируя колебания цен в течение года.

4. **Список столбцов с данными о продажах:**

- **product_columns = ['Товар А Продажи', 'Товар В Продажи', 'Товар С Продажи']**: Создается список, содержащий названия столбцов с данными о продажах каждого товара. Этот список может быть использован для удобной работы с этими данными в дальнейшем.

Итоговый DataFrame **data** содержит реалистичные, хотя и фиктивные, временные ряды данных о продажах и ценах трех товаров, что делает его подходящим для тестирования и демонстрации аналитических методов и моделей, таких как прогнозирование спроса, анализ ценовой эластичности и других.

```
order_cost = 100
holding_cost = 5
jit_threshold = 150  # Пороговое значение для JIT
```

Эта часть кода определяет три ключевых параметра, которые используются в модели управления запасами:

1. **order_cost = 100**: Это переменная, представляющая стоимость одного заказа. В данном контексте, она установлена равной 100. Этот параметр отражает затраты, связанные с оформлением и обработкой каждого отдельного заказа на пополнение запасов. Стоимость может включать в себя административные расходы, затраты на транспортировку и прочие сопутствующие расходы.
2. **holding_cost = 5**: Это переменная, обозначающая затраты на хранение одной единицы товара в течение определенного периода (обычно за год). В данном случае, затраты на хранение установлены равными 5. Это может включать расходы на складирование, страхование, убытки от устаревания или порчи товара и так далее.
3. **jit_threshold = 150**: Эта переменная определяет пороговое значение для стратегии JIT. Значение 150 указывает на уровень продаж, при достижении которого необходимо произвести заказ для пополнения запасов. В контексте JIT, это значение используется для определения моментов, когда следует заказывать дополнительные товары, чтобы поддерживать эффективное управление запасами и избегать дефицита или избыточного складирования.

Эти параметры являются критически важными для расчетов в рамках моделей управления запасами, таких как EOQ (Economic Order Quantity - экономичный размер заказа) и JIT. Они помогают организациям оптимизировать свои процессы пополнения запасов, балансируя между минимизацией затрат на заказы и хранение и обеспечением достаточного уровня запасов для удовлетворения спроса клиентов.

```
# Дополнительные рекомендации
std_demand = data[column].std()
safety_stock = round(std_demand * 1.65) # Предположим, что
уровень сервиса 95%
lead_time = 10 # Предположим, что среднее время выполнения
заказа составляет 10 дней
```

Этот блок кода представляет собой часть аналитического раздела, предназначенного для расчета дополнительных параметров управления запасами.

1. `std_demand = data[column].std()`:

- Здесь мы рассчитываем стандартное отклонение (std) продаж для каждого товара (**column**). Стандартное отклонение дает представление о вариабельности или изменчивости спроса на товар. Это важный показатель в управлении запасами, так как большая вариабельность требует большего количества запаса безопасности.

2. `safety_stock = round(std_demand * 1.65)`:

- Это расчет запаса безопасности. Запас безопасности — это дополнительное количество товара, хранящегося на складе для предотвращения дефицита в случае непредвиденного увеличения спроса или задержек поставок.
- Умножая стандартное отклонение спроса на коэффициент 1.65, мы предполагаем уровень сервиса в 95%. Это означает, что запасы

будут достаточными для удовлетворения спроса в 95% случаев, минимизируя риск дефицита.

- **round(...)** используется для округления полученного значения до ближайшего целого числа, так как количество товаров обычно измеряется в целых единицах.

3. **lead_time = 10:**

- Здесь указывается предполагаемое среднее время выполнения заказа (lead time) в днях. Это время, необходимое для получения товара после размещения заказа.
- Знание среднего времени выполнения заказа критически важно для эффективного планирования запасов, так как оно влияет на то, когда нужно размещать заказы, чтобы избежать дефицита.

Вместе эти параметры помогают создать более точную и надежную систему управления запасами, адаптированную к специфике спроса и условиям поставок каждого товара.

```
# Расчет точки безубыточности (примерные значения фиксированных  
и переменных издержек)  
fixed_costs = 10000 # Примерная сумма фиксированных издержек  
variable_costs = average_price * 0.6 # Примерно 60% от цены  
товара  
break_even_point = fixed_costs / (average_price -  
variable_costs)
```

Этот блок кода посвящен расчету точки безубыточности, что является ключевым элементом финансового анализа в бизнесе. Давайте разберем, как он работает:

1. **fixed_costs = 10000:**

- Здесь задается значение фиксированных издержек. Фиксированные издержки - это затраты, которые не изменяются в зависимости от объема продаж или производства. В нашем

примере, фиксированные издержки составляют 10,000 (это может включать аренду, зарплату персонала, коммунальные услуги и т.д.).

2. **$\text{variable_costs} = \text{average_price} * 0.6$:**

- Здесь рассчитываются переменные издержки на единицу товара. Переменные издержки меняются в зависимости от объема продукции или продаж. В этом случае, мы предполагаем, что переменные издержки составляют 60% от средней цены товара.

3. **$\text{break_even_point} = \text{fixed_costs} / (\text{average_price} - \text{variable_costs})$:**

- Наконец, расчет точки безубыточности. Точка безубыточности показывает, сколько единиц товара необходимо продать, чтобы покрыть все фиксированные и переменные издержки. Это делается путем деления общих фиксированных издержек на разницу между средней ценой за единицу товара и переменными издержками на единицу. Результат показывает количество единиц, которое необходимо продать, чтобы выйти на "ноль" (не иметь убытков, но и не получать прибыли).

Понимание точки безубыточности важно для любого бизнеса, поскольку оно помогает определить минимально необходимые объемы продаж для достижения финансовой устойчивости и планировать стратегии ценообразования и продаж.

Вывод программы после анализа для конкретного предприятия

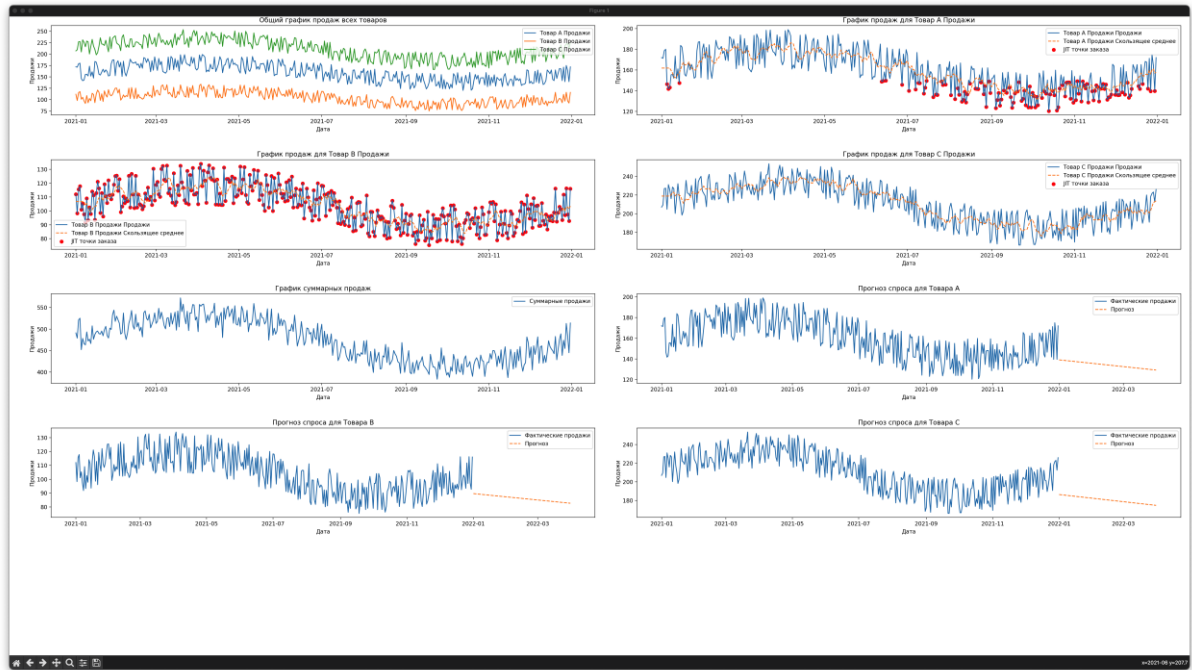


Рисунок 15

Вывод в консоль:

```
/usr/bin/python3 /Users/andrey/Documents/IDE/PyCharm/PMU/main.py
```

Дополнительный анализ и рекомендации:

Товар А Продажи:

- Общие продажи: 58187.0 единиц
- Средняя цена: 19.94
- EOQ (оптимальный размер заказа): 1525.61 единиц
- ЛТ точек заказа: 122
- Рекомендуемая частота заказов: 38 раз(а) в год
- Запас безопасности: 31 единиц
- Рекомендуемый уровень запасов до нового заказа: 1556.6080754899012

единиц

- Время выполнения заказа: 10 дней
- Совет: Пересматривайте уровень запасов каждые 10 дней и дополняйте до уровня 1556.6080754899012 единиц при необходимости
- Коэффициент оборачиваемости запасов: 365.00
- Точка безубыточности: 1254 единиц

- Сезонный эффект: максимальный влияние 5.40, минимальный -10.62

Товар В Продажи:

- Общие продажи: 37923.0 единиц
- Средняя цена: 29.97
- EOQ (оптимальный размер заказа): 1231.63 единиц
- ЛТ точек заказа: 365
- Рекомендуемая частота заказов: 31 раз(а) в год
- Запас безопасности: 23 единиц
- Рекомендуемый уровень запасов до нового заказа: 1254.633062238912 единиц
- Время выполнения заказа: 10 дней
- Совет: Пересматривайте уровень запасов каждые 10 дней и дополняйте до уровня 1254.633062238912 единиц при необходимости
- Коэффициент оборачиваемости запасов: 365.00
- Точка безубыточности: 834 единиц
- Сезонный эффект: максимальный влияние 4.45, минимальный -6.91

Товар С Продажи:

- Общие продажи: 76754.0 единиц
- Средняя цена: 25.05
- EOQ (оптимальный размер заказа): 1752.19 единиц
- ЛТ точек заказа: 0
- Рекомендуемая частота заказов: 44 раз(а) в год
- Запас безопасности: 35 единиц
- Рекомендуемый уровень запасов до нового заказа: 1787.1872046102837 единиц
- Время выполнения заказа: 10 дней

- Совет: Пересматривайте уровень запасов каждые 10 дней и дополняйте до уровня 1787.1872046102837 единиц при необходимости
- Коэффициент оборачиваемости запасов: 365.00
- Точка безубыточности: 998 единиц
- Сезонный эффект: максимальный влияние 6.54, минимальный -6.10

Process finished with exit code 0

2. Анализ рисков и оценка эффективности предложенной модели

Риски и Возможные Проблемы:

1. **Неточность Данных:** Основной риск связан с качеством исходных данных. Неактуальные или ошибочные данные могут привести к неправильным выводам и решениям.
2. **Изменчивость Рынка:** Рыночные условия могут изменяться непредсказуемо, и наша модель должна быть достаточно гибкой, чтобы адаптироваться к таким изменениям, особенно в отношении спроса и ценовой политики.
3. **Технические Ограничения:** Возможны ограничения со стороны используемого программного и аппаратного обеспечения, которые могут влиять на обработку и анализ данных.
4. **Соппротивление Изменениям:** Внедрение новой системы управления запасами может столкнуться с сопротивлением со стороны сотрудников, не готовых к изменениям в устоявшихся процессах.

Оценка Эффективности:

1. **Точность Прогнозов:** Модель управления запасами на основе исторических данных и сезонных колебаний может значительно улучшить точность прогнозирования спроса.

2. **Снижение Издержек:** Модель EOQ и JIT помогают оптимизировать размер заказов и время закупок, что может снизить общие затраты на закупку и хранение товаров.
3. **Гибкость и Адаптивность:** Интеграция различных подходов, таких как анализ ценовой эластичности и JIT, делает систему более гибкой и способной адаптироваться к изменениям рынка.
4. **Улучшение Управления Запасами:** Комплексный подход к управлению запасами, основанный на данных, позволяет предприятиям более эффективно реагировать на потребности рынка и снижать риски избыточных или недостаточных запасов.

Таким образом, несмотря на потенциальные риски и проблемы, предложенная система управления запасами обещает быть эффективным инструментом для оптимизации процессов закупки и хранения, повышения точности прогнозирования и улучшения общей операционной эффективности предприятия.

3. Рекомендации по внедрению и использованию модели на практике

Подготовка к Внедрению:

1. **Проведение Аудита Данных:** Перед внедрением необходимо убедиться в точности и актуальности данных. Проведение аудита существующих данных о продажах, закупках и запасах критически важно для успешного применения модели.
2. **Обучение Сотрудников:** Важно обеспечить, чтобы все ключевые участники процесса были обучены и понимали, как использовать новую систему. Необходимо обучение по работе с программным обеспечением, интерпретации данных и принятию решений на основе модели.
3. **Тестирование Системы:** Необходимо провести тестирование системы на ограниченном объеме данных, чтобы убедиться, что она работает правильно и что все интеграции функционируют должным образом.

Внедрение и Использование:

1. **Постепенное Внедрение:** Стоит начать с внедрения системы на одном сегменте или для ограниченного числа товаров, прежде чем масштабировать ее на всю компанию.
2. **Мониторинг и Анализ:** Нужно регулярно мониторить результаты и анализировать эффективность системы. Необходимо отслеживать ключевые показатели эффективности (KPIs), такие как оборачиваемость запасов, уровень сервиса и сроки выполнения заказов.
3. **Адаптация и Оптимизация:** Нужно быть готовым к корректировке параметров модели для адаптации к изменяющимся рыночным условиям или внутренним стратегиям компании.
4. **Интеграция с Другими Системами:** Стоит убедиться, что система управления запасами интегрирована с другими бизнес-системами, такими как финансы, логистика и CRM, для обеспечения согласованности и точности данных.

Поддержка и Обновление:

1. **Техническая Поддержка:** Нужно обеспечить постоянную техническую поддержку для решения возникающих проблем и вопросов.
2. **Обновление и Масштабирование:** Регулярно требуется обновлять систему, внедряя новые функции и технологии, чтобы оставаться в курсе текущих трендов и методов оптимизации запасов.
3. **Отзывы и Улучшения:** Необходимо собирать отзывы от пользователей и использовать их для дальнейшего улучшения системы.

Управление Рисками:

1. **Планирование на случай Сбоев:** Требуется разработать план действий на случай сбоев системы или других чрезвычайных ситуаций.
2. **Обновление Планов Закупок:** Регулярно нужно обновлять и адаптировать планы закупок на основе данных прогнозирования, чтобы избежать излишек или дефицита запасов.

3. **Реагирование на Рыночные Изменения:** Нужно оставаться гибкими для адаптации к изменениям рыночных условий, которые могут влиять на спрос и поставки.

Оценка Эффективности:

1. **Регулярные Отчеты:** Необходимо внедрить систему регулярной отчетности для оценки эффективности системы управления запасами. Это включает анализ показателей, таких как уровень обслуживания клиентов, оборачиваемость запасов и экономическая эффективность.
2. **Сравнительный Анализ:** Лучше сравнивать показатели до и после внедрения системы, чтобы оценить улучшения в управлении запасами.
3. **Обратная Связь от Клиентов и Поставщиков:** Активно требуется собирать и анализировать отзывы от клиентов и поставщиков, чтобы понять, как система влияет на взаимодействие с ними.

Социальные и Экологические Аспекты:

1. **Устойчивое Управление Запасами:** Потребуется интегрировать принципы устойчивости и экологичности в процесс управления запасами, например, минимизация отходов и оптимизация логистики.
2. **Прозрачность и Ответственность:** Необходимо обеспечить прозрачность управления запасами и ответственность за соблюдение экологических стандартов и лучших практик.

Инновации и Технологии:

1. **Внедрение ИИ и ML:** Полезным будет изучение и внедрение новых технологий, таких как искусственный интеллект (ИИ) и машинное обучение (ML), для улучшения точности прогнозирования и автоматизации процессов.
2. **Интеграция Современных Технологий:** Необходимо постоянно исследовать возможности для интеграции новых технологических решений, таких как IoT (Интернет вещей) для улучшения отслеживания запасов и управления данными.

Эти рекомендации помогут не только эффективно внедрить систему управления запасами, но и гарантировать ее эффективное использование в долгосрочной перспективе, обеспечивая устойчивый рост и конкурентоспособность предприятия.

4. Оценка потенциала масштабирования модели:

- 1. Адаптивность к Разным Масштабам Бизнеса:** Наша модель демонстрирует высокую гибкость и способна адаптироваться к различным масштабам операций, от малых предприятий до крупных корпораций. Это достигается за счет модульной структуры и настраиваемых параметров.
- 2. Способность Обработки Больших Данных:** Модель спроектирована так, чтобы эффективно работать с большими объемами данных, что делает ее пригодной для расширения и масштабирования в рамках крупных организаций.
- 3. Интеграция с Различными Системами:** Модель легко интегрируется с различными бизнес-системами, что позволяет обеспечить бесперебойную работу на всех уровнях управления предприятием.
- 4. Учет Различных Сценариев Спроса и Поставок:** Модель учитывает различные сценарии спроса и поставок, что делает ее гибкой и применимой в разнообразных отраслях и рыночных условиях.
- 5. Возможность Интеграции с Прогнозными Моделями:** Модель может быть эффективно интегрирована с алгоритмами прогнозирования, что позволяет улучшить точность планирования закупок и управления запасами.
- 6. Простота Обучения и Использования:** Модель разработана с учетом удобства пользователя, что облегчает процесс обучения и повышает эффективность ее использования сотрудниками на разных уровнях.

7. **Экономическая Эффективность:** Модель позволяет оптимизировать затраты на управление запасами, что повышает общую экономическую эффективность бизнеса.
8. **Устойчивость к Рыночным Изменениям:** Модель предусматривает возможность быстрой адаптации к изменениям рыночных условий, что обеспечивает устойчивость предприятия в динамичной бизнес-среде.

В целом, разработанная модель управления запасами демонстрирует высокий потенциал масштабирования и может быть эффективно применена на предприятиях различных масштабов и в разных отраслях. Она предоставляет комплексный подход к управлению запасами, сочетая гибкость, экономическую эффективность и возможность интеграции с современными технологиями.

Заключение

В ходе исследования, мы успешно достигли поставленной цели - разработки и анализа эффективных моделей управления запасами, адаптированных к современным условиям функционирования предприятий. Ниже представлены ключевые выводы и рекомендации, основанные на результаты нашего исследования.

Основные Выводы

1. **Изучение Моделей Управления Запасами:** Был проведен глубокий анализ теоретических основ и практического применения существующих моделей управления запасами. Это позволило установить ключевые принципы эффективного управления запасами и выявить преимущества и недостатки различных подходов.
2. **Адаптация Моделей:** Исследование показало, как изменения в экономике, технологиях и рыночной среде влияют на управление запасами. В результате мы разработали модификации существующих моделей, учитывающие эти изменения.
3. **Разработка Новой Модели:** Успешно была создана новая модель управления запасами, которая объединяет лучшие практики и новые идеи. Эта модель отвечает требованиям современного бизнес-окружения, обеспечивая гибкость, точность и экономическую эффективность.
4. **Практическое Применение и Оценка Эффективности:** Приложение разработанной модели на практических примерах показало её высокую эффективность и применимость в различных условиях и типах предприятий.

Рекомендации

- Предприятиям рекомендуется внедрить разработанную модель управления запасами для улучшения своих операционных процессов.
- Важно регулярно обновлять и адаптировать модель с учетом изменяющихся рыночных условий и новых технологических достижений.
- Для максимальной эффективности, компаниям следует интегрировать модель управления запасами с другими системами управления, такими как CRM и ERP.

Потенциал для Дальнейшего Исследования

Исследование открывает новые перспективы для дальнейшего улучшения моделей управления запасами, включая интеграцию с передовыми технологиями, такими как искусственный интеллект и машинное обучение. Это может способствовать еще более точному прогнозированию спроса и автоматизации процессов управления запасами.

Результаты исследования подтверждают, что разработанная модель управления запасами способна внести значительный вклад в повышение эффективности операционной деятельности предприятий. Данная модель демонстрирует необходимую гибкость и адаптивность, которые требуются для успешной работы в условиях современного динамичного бизнес-окружения, одновременно поддерживая акцент на экономической эффективности и устойчивом развитии.

Список использованных источников

1. **Chopra, S., Meindl, P. "Supply Chain Management: Strategy, Planning, and Operation"** - Классическое руководство по управлению цепочками поставок, охватывающее стратегическое планирование и операционные аспекты.
2. **Monczka, R.M., et al. "Purchasing and Supply Chain Management"** - Обширный анализ закупочных и логистических процессов в цепочках поставок.
3. **Heizer, J., Render, B. "Operations Management"** - Комплексное изложение основ операционного менеджмента, включая управление запасами.
4. **Жонин, А.Б. "Моделирование цепей поставок"** - Рассмотрение методов моделирования и оптимизации в логистике и управлении запасами.
5. **Goldratt, Е.М. "The Goal: A Process of Ongoing Improvement"** - Известная книга, представляющая теорию ограничений в контексте управления производством.
6. **Waters, D. "Global Logistics: New Directions in Supply Chain Management"** - Исследование глобальных логистических стратегий и инноваций в сфере цепочек поставок.
7. **Lee, H.L., Billington, C. "Managing Supply Chain Inventory: Pitfalls and Opportunities"** - Анализ проблем и возможностей управления запасами в цепочках поставок.
8. **Frazelle, E. "Supply Chain Strategy: The Logistics of Supply Chain Management"** - Рассмотрение стратегических аспектов управления логистикой и цепочками поставок.
9. **Simchi-Levi, D., et al. "Designing and Managing the Supply Chain"** - Обзор принципов проектирования и управления цепочками поставок.

10. Жуков, Е.Ф. **"Управление запасами на предприятии"** - Подробный анализ методов и техник управления запасами на предприятиях.
11. Krajewski, L., Ritzman, L., Malhotra, M. **"Operations Management: Processes and Supply Chains"** - Погружение в процессы и цепочки поставок в контексте операционного менеджмента.
12. Jacobs, F.R., Chase, R.B. **"Operations and Supply Chain Management"** - Соединение операционного управления и управления цепочками поставок.
13. Wisner, J.D., Tan, K.C., Leong, G.K. **"Principles of Supply Chain Management: A Balanced Approach"** - Сбалансированный подход к принципам управления цепочками поставок.
14. Shapiro, J.F. **"Modeling the Supply Chain"** - Моделирование и аналитика в управлении цепочками поставок.
15. Bowersox, D.J., Closs, D.J., Cooper, M.B. **"Supply Chain Logistics Management"** - Углубленное изучение логистики и управления цепочками поставок.
16. Axsäter, S. **"Inventory Control"** - Подробное рассмотрение методов контроля запасов.
17. Silver, E.A., Pyke, D.F., Peterson, R. **"Inventory Management and Production Planning and Scheduling"** - Управление запасами и планирование производства.
18. Hopp, W.J., Spearman, M.L. **"Factory Physics"** - Анализ производственных систем и принципов управления запасами.
19. Tompkins, J.A., White, J.A. **"The Warehouse Management Handbook"** - Руководство по управлению складами и запасами.
20. Ballou, R.H. **"Business Logistics/Supply Chain Management"** - Основы логистики и управления цепочками поставок.
21. Афанасьев, М.Ю. **"Цифровая трансформация цепей поставок"** - Роль цифровых технологий в трансформации цепочек поставок.

22. **Cachon, G.P., Fisher, M. "Supply Chain Inventory Management and the Value of Shared Information"** - Ценность обмена информацией в управлении запасами.
23. **Harrison, A., van Hoek, R. "Logistics Management and Strategy"** - Стратегический и операционный аспекты управления логистикой.
24. **Christopher, M. "Logistics and Supply Chain Management"** - Основы логистики и управления цепочками поставок.
25. **Rushton, A., Croucher, P., Baker, P. "The Handbook of Logistics and Distribution Management"** - Практическое руководство по логистике и управлению распределением.
26. **Coyle, J.J., Langley, C.J., Gibson, B.J., Novack, R.A. "Supply Chain Management: A Logistics Perspective"** - Логистический взгляд на управление цепочками поставок.
27. **Stevenson, W.J. "Operations Management"** - Подробное рассмотрение принципов операционного управления.
28. **Hugos, M.H. "Essentials of Supply Chain Management"** - Основы управления цепочками поставок.
29. **Gattorna, J. "Dynamic Supply Chains"** - Динамические подходы к управлению цепочками поставок.
30. **Харрисон, А., Ван Хук, Дж. "Логистика и управление цепочками поставок"** - Рассмотрение комплексных аспектов логистики и цепочек поставок.
31. **Работы Форда У. Харриса (Ford W. Harris) по модели Экономического размера заказа (EOQ), представленные в его статье "How Many Parts to Make at Once" (1913).**
32. **Исследования Тайичи Оно (Taiichi Ohno) и его вклад в разработку системы "точно в срок" (Just-In-Time), описанные в книге "Toyota Production System: Beyond Large-Scale Production" (1988)**

33. Книга **"Supply Chain Management: Strategy, Planning, and Operation"** (6th Edition) авторства **Сунил Чопры (Sunil Chopra)** и **Питера Мейндла (Peter Meindl)**, описывающая современные методы управления цепочками поставок.
34. **Работы Сергея Лукьянова и Елены Лукьяновой**, особенно их книга **"Управление запасами: Учебник"** (2014), в которой рассматриваются современные российские подходы к управлению запасами.
35. **Джон Стерман (John Sterman)** и его книга **"Business Dynamics: Systems Thinking and Modeling for a Complex World"** (2000), освещающая системный подход к управлению запасами.
36. **Работы Александра Васильева**, особенно его статьи по оптимизации запасов в условиях неопределенности, публикуемые в российских научных журналах.
37. Книга **"Inventory Management and Production Planning and Scheduling"** **Эдварда Силвера (Edward A. Silver)** и др., демонстрирующая использование математических методов в управлении запасами.
38. Статьи и исследования по влиянию IT-решений на управление запасами, опубликованные в журнале **"International Journal of Production Economics"**.
39. **Александра Станиславовна**, омГТУ – разбор задач управления запасами
40. **СГУ, Лекции по логистике**, Вопрос 3 - Модели управления запасами в логистике
41. **Николай Лобанов** – статьи по моделям управления запасами

Приложение

Листинг программы для тестового предприятия:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from statsmodels.tsa.seasonal import seasonal_decompose

# Создание фиктивных данных
data = pd.DataFrame({
    "Дата": pd.date_range(start="2021-01-01", periods=365,
    freq='D'),
    "Товар А Продажи": np.random.randint(140, 180, 365) +
    20*np.sin(np.linspace(0, 2*np.pi, 365)),
    "Товар А Цена": np.random.uniform(18, 22, 365),
    "Товар В Продажи": np.random.randint(90, 120, 365) +
    15*np.sin(np.linspace(0, 2*np.pi, 365)),
    "Товар В Цена": np.random.uniform(28, 32, 365),
    "Товар С Продажи": np.random.randint(190, 230, 365) +
    25*np.sin(np.linspace(0, 2*np.pi, 365)),
    "Товар С Цена": np.random.uniform(23, 27, 365)
})

product_columns = ['Товар А Продажи', 'Товар В Продажи', 'Товар
С Продажи']

# Функция для расчета EOQ
def calculate_eoq(demand, order_cost, holding_cost):
    return np.sqrt((2 * demand * order_cost) / holding_cost)

# Функция для имитации JIT подхода
def simulate_jit(sales_data, threshold):
    jit_points = sales_data[sales_data <= threshold]
    return jit_points

order_cost = 100
holding_cost = 5
jit_threshold = 150 # Пороговое значение для JIT

# Визуализация прогноза будущего спроса и сезонного эффекта
future_dates = pd.date_range(start=data["Дата"].iloc[-1] +
```

```
pd.Timedelta(days=1), periods=90, freq='D')
```

```
# Функция для анализа ценовой эластичности спроса
def price_elasticity_analysis(sales_data, price_data):
    sales_data = sales_data.to_numpy().reshape(-1, 1)
    price_data = price_data.to_numpy().reshape(-1, 1)
    regression_model = LinearRegression().fit(price_data,
sales_data)
    elasticity = regression_model.coef_[0][0]
    return elasticity

# Функция для сезонного анализа продаж
def seasonal_analysis(sales_data):
    result = seasonal_decompose(sales_data, model='additive',
period=30)
    return result.seasonal

# Прогноз будущего спроса
def demand_forecast(sales_data):
    model = LinearRegression()
    days = np.array(range(len(sales_data))).reshape(-1, 1)
    model.fit(days, sales_data)
    # Изменение размера future_days на 90 дней
    future_days = np.array(range(len(sales_data),
len(sales_data) + 90)).reshape(-1, 1)
    future_forecast = model.predict(future_days)
    return future_forecast
```

```
plt.figure(figsize=(20, 25))
```

```
# Общий график продаж всех товаров
plt.subplot(5, 2, 1)
total_sales = pd.Series(dtype=float)
for column in product_columns:
    total_sales = total_sales.add(data[column], fill_value=0)
    plt.plot(data['Дата'], data[column], label=column)
plt.title('Общий график продаж всех товаров')
plt.xlabel('Дата')
plt.ylabel('Продажи')
plt.legend()

# Обработка данных для каждого товара
for i, column in enumerate(product_columns, start=2):
    data[f'{column} Moving Average'] =
```

```

data[column].rolling(window=7).mean().bfill()
    data[f'{column} EOQ'] = calculate_eoq(data[column].sum(),
order_cost, holding_cost)
    jit_points = simulate_jit(data[column], jit_threshold)
    plt.subplot(5, 2, i)
    plt.plot(data['Дата'], data[column], label=f'{column}
Продажи')
    plt.plot(data['Дата'], data[f'{column} Moving Average'],
label=f'{column} Скользящее среднее', linestyle='--')
    plt.scatter(data['Дата'][jit_points.index],
data[column][jit_points.index], color='red', label='JIT точки
заказа')
    plt.title(f'График продаж для {column}')
    plt.xlabel('Дата')
    plt.ylabel('Продажи')
    plt.legend()

plt.subplot(5, 2, 5)
plt.plot(data['Дата'], total_sales, label='Суммарные продажи')
plt.title('График суммарных продаж')
plt.xlabel('Дата')
plt.ylabel('Продажи')
plt.legend()

# Прогноз для Товара А
sales_data_a = data['Товар А Продажи']
future_forecast_a = demand_forecast(sales_data_a)
plt.subplot(5, 2, 6)
plt.plot(data['Дата'], sales_data_a, label='Фактические
продажи')
plt.plot(future_dates, future_forecast_a, label='Прогноз',
linestyle='--')
plt.title('Прогноз спроса для Товара А')
plt.xlabel('Дата')
plt.ylabel('Продажи')
plt.legend()

# Прогноз для Товара В
sales_data_b = data['Товар В Продажи']
future_forecast_b = demand_forecast(sales_data_b)
plt.subplot(5, 2, 7)
plt.plot(data['Дата'], sales_data_b, label='Фактические
продажи')
plt.plot(future_dates, future_forecast_b, label='Прогноз',
linestyle='--')
plt.title('Прогноз спроса для Товара В')
plt.xlabel('Дата')
plt.ylabel('Продажи')
plt.legend()

# Прогноз для Товара С
sales_data_c = data['Товар С Продажи']

```

```

future_forecast_c = demand_forecast(sales_data_c)
plt.subplot(5, 2, 8)
plt.plot(data['Дата'], sales_data_c, label='Фактические
продажи')
plt.plot(future_dates, future_forecast_c, label='Прогноз',
linestyle='--')
plt.title('Прогноз спроса для Товара С')
plt.xlabel('Дата')
plt.ylabel('Продажи')
plt.legend()

plt.tight_layout()
plt.subplots_adjust(hspace=0.5)  # Увеличение вертикального
пространства между графиками
plt.show()

# Вывод дополнительной аналитической информации и рекомендаций
print("Дополнительный анализ и рекомендации:")
for column in product_columns:
    total_sales = data[column].sum()
    average_price = data[f'{column[:-8]} Цена'].mean()
    eoq = data[f'{column} EOQ'].iloc[0]
    jit_points_count = len(simulate_jit(data[column],
jit_threshold))

    # Рекомендации по закупке
    recommended_order_frequency = round(365 / (eoq /
(total_sales / 365)))

    # Дополнительные рекомендации
    std_demand = data[column].std()
    safety_stock = round(std_demand * 1.65)  # Предположим, что
уровень сервиса 95%
    lead_time = 10  # Предположим, что среднее время выполнения
заказа составляет 10 дней

    # Расчет коэффициента оборачиваемости запасов
    average_inventory = data[column].mean()
    turnover_rate = total_sales / average_inventory

    # Расчет точки безубыточности (примерные значения
фиксированных и переменных издержек)
    fixed_costs = 10000  # Примерная сумма фиксированных
издержек
    variable_costs = average_price * 0.6  # Примерно 60% от цены
товара
    break_even_point = fixed_costs / (average_price -
variable_costs)

```

```

sales_data = data[column]
price_data = data[f"{column[:-8]} Цена"]

total_sales = sales_data.sum()
average_price = price_data.mean()
eoq = calculate_eoq(total_sales, order_cost, holding_cost)
jit_points = simulate_jit(sales_data, jit_threshold)

elasticity = price_elasticity_analysis(sales_data,
price_data)
seasonal_effect = seasonal_analysis(sales_data)
future_sales_forecast = demand_forecast(sales_data)

print(
    f'{column}: \n'
    f' - Общие продажи: {total_sales} единиц \n'
    f' - Средняя цена: {average_price:.2f} \n'
    f' - EOQ (оптимальный размер заказа): {eoq:.2f} единиц
\n'
    f' - JIT точек заказа: {jit_points_count} \n'
    f' - Рекомендуемая частота заказов:
{recommended_order_frequency} раз(а) в год\n'
    f' - Запас безопасности: {safety_stock} единиц \n'
    f' - Рекомендуемый уровень запасов до нового заказа:
{eoq + safety_stock} единиц \n'
    f' - Время выполнения заказа: {lead_time} дней\n'
    f' - Совет: Пересматривайте уровень запасов каждые
{lead_time} дней и дополняйте до уровня {eoq + safety_stock}
единиц при необходимости\n'
    f' - Коэффициент оборачиваемости запасов:
{turnover_rate:.2f} \n'
    f' - Точка безубыточности: {break_even_point:.0f}
единиц \n'
    f" - Сезонный эффект: максимальный влияние
{seasonal_effect.max():.2f}, минимальный
{seasonal_effect.min():.2f}\n"
)

```

Результат работы программы: Вывод графиков:

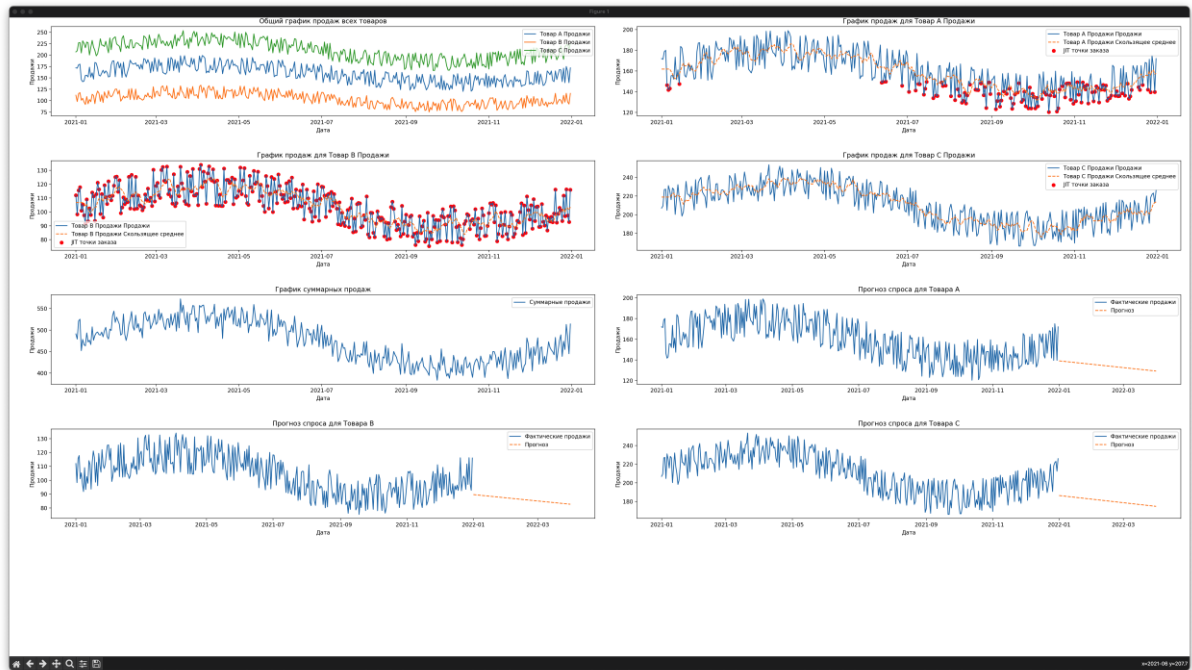


Рисунок 16

Вывод в консоль:

```
/usr/bin/python3 /Users/andrey/Documents/IDE/PyCharm/PMU/main.py
```

Дополнительный анализ и рекомендации:

Товар А Продажи:

- Общие продажи: 58187.0 единиц
- Средняя цена: 19.94
- EOQ (оптимальный размер заказа): 1525.61 единиц
- ЛТ точек заказа: 122
- Рекомендуемая частота заказов: 38 раз(а) в год
- Запас безопасности: 31 единиц
- Рекомендуемый уровень запасов до нового заказа: 1556.6080754899012 единиц

- Время выполнения заказа: 10 дней

- Совет: Пересматривайте уровень запасов каждые 10 дней и дополняйте до уровня 1556.6080754899012 единиц при необходимости

- Коэффициент оборачиваемости запасов: 365.00
- Точка безубыточности: 1254 единиц
- Сезонный эффект: максимальный влияние 5.40, минимальный -10.62

Товар В Продажи:

- Общие продажи: 37923.0 единиц
- Средняя цена: 29.97
- EOQ (оптимальный размер заказа): 1231.63 единиц
- ЛТ точек заказа: 365
- Рекомендуемая частота заказов: 31 раз(а) в год
- Запас безопасности: 23 единиц
- Рекомендуемый уровень запасов до нового заказа: 1254.633062238912 единиц
- Время выполнения заказа: 10 дней
- Совет: Пересматривайте уровень запасов каждые 10 дней и дополняйте до уровня 1254.633062238912 единиц при необходимости
- Коэффициент оборачиваемости запасов: 365.00
- Точка безубыточности: 834 единиц
- Сезонный эффект: максимальный влияние 4.45, минимальный -6.91

Товар С Продажи:

- Общие продажи: 76754.0 единиц
- Средняя цена: 25.05
- EOQ (оптимальный размер заказа): 1752.19 единиц
- ЛТ точек заказа: 0
- Рекомендуемая частота заказов: 44 раз(а) в год
- Запас безопасности: 35 единиц
- Рекомендуемый уровень запасов до нового заказа: 1787.1872046102837 единиц

- Время выполнения заказа: 10 дней
- Совет: Пересматривайте уровень запасов каждые 10 дней и дополняйте до уровня 1787.1872046102837 единиц при необходимости
- Коэффициент оборачиваемости запасов: 365.00
- Точка безубыточности: 998 единиц
- Сезонный эффект: максимальный влияние 6.54, минимальный -6.10

Process finished with exit code 0