

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра _____ «Прикладная математика и информатика» _____
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки / специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка программного обеспечения для рационального размещения товаров на складе на основе ABC-анализа»

Обучающийся	_____ А.А. Арефьев _____ (Инициалы Фамилия) (личная подпись)
Руководитель	_____ к. т. н., доцент, Н.А. Сосина _____ (ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)
Консультант	_____ к. п. н., доцент, С.А. Гудкова _____ (ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема бакалаврской работы – «Разработка программного обеспечения для рационального размещения товаров на складе на основе ABC-анализа».

Целью работы является разработка программного обеспечения, в котором реализованы алгоритмы классификации ABC-анализа и XYZ-анализа, а также алгоритм распределения товара на складе.

Бакалаврская работа состоит из введения, трёх разделов, заключения и списка литературы.

В первом разделе описано исследование предметной области.

Во втором разделе рассматриваются используемые для разработки алгоритма инструменты и описывается алгоритм работы программы.

В третьем разделе описывается разработанное программное обеспечение и производится его тестирование.

В заключении представлены выводы по проделанной работе.

В работе использованы 19 рисунков, список литературы содержит 25 источников. Общий объем выпускной квалификационной работы составляет 42 страницы.

Abstract

The title of the graduation work is «Software development for the rational placement of goods in a warehouse based on ABC analysis».

The paper provides software that implements algorithms for the classification of ABC analysis and XYZ analysis, as well as an algorithm for the distribution of goods in a warehouse.

The senior paper consists of an introduction, three parts, a conclusion, tables and list of references including foreign sources.

The first part describes the research of the subject area.

The second part discusses the tools used to develop the algorithm and describes the algorithm of the program.

The third part describes the developed software and tests it.

In conclusion, the conclusions on the work done are presented.

The work uses 19 images the list of references contains 25 sources. The total volume of the final qualifying work is 42 pages.

Содержание

Введение.....	5
1 Анализ методов оптимизации управления запасами	7
1.1 Описание ABC-анализа	7
1.2 Применение ABC-анализа в различных сферах	9
1.3 Преимущества и ограничения ABC-анализа.....	11
1.4 Описание дополнений метода ABC-анализа	12
1.5. Описание XYZ-анализа	13
1.6. Применение XYZ-анализа.....	15
2 Разработка алгоритма	18
2.1 Выбор языка программирования и среды разработки	18
2.2 Библиотеки для работы с Excel файлами	19
2.3 Определение требований к алгоритму.....	20
2.4 Создание алгоритма	21
3 Реализация и тестирование программного обеспечения	24
3.1 Реализация ПО.....	24
3.2 Тестирование ПО	35
Заключение	39
Список используемой литературы	40

Введение

В условиях современного рынка, характеризующегося высокой конкуренцией и динамичностью, эффективность управления складскими запасами приобретает особое значение. Компании, стремящиеся к максимизации прибыли и удовлетворению потребностей клиентов, сталкиваются с рядом вызовов, связанных с оптимизацией хранения товаров, минимизацией затрат на их содержание и своевременным обеспечением поставок. В этих условиях особенно актуальны методы и инструменты, позволяющие эффективно управлять запасами и их размещением на складе.

Цель данной работы заключается в разработке программного обеспечения для рационального размещения товаров на складе на основе ABC-анализа. Данный метод позволяет классифицировать товары по их значимости для бизнеса, что, в свою очередь, способствует оптимизации складских процессов. Программное обеспечение, разработанное в рамках этой работы, реализует алгоритмы классификации ABC-анализа и XYZ-анализа, а также алгоритм распределения товаров по складу, что обеспечивает более эффективное использование складского пространства.

Актуальность данной темы обусловлена несколькими факторами. Во-первых, правильное управление складскими запасами позволяет снизить издержки, связанные с их хранением, и уменьшить количество неликвидных товаров. Во-вторых, своевременное и точное размещение товаров на складе способствует ускорению процессов комплектации заказов и повышению уровня обслуживания клиентов. В-третьих, автоматизация процессов управления запасами с использованием современных программных решений позволяет значительно повысить эффективность работы склада и сократить затраты времени на выполнение рутинных операций.

В рамках данной работы будут выполнены следующие задачи.

Проведено исследование предметной области, включающее анализ методов оптимизации управления запасами и их применения в различных сферах бизнеса.

Разработан алгоритм классификации товаров с использованием методов ABC-анализа и XYZ-анализа.

Подобраны и изучены инструменты для реализации алгоритма, в том числе язык программирования C++ и библиотека xlnt для работы с Excel-файлами.

Разработано программное обеспечение, реализующее алгоритмы классификации и распределения товаров по складу.

Проведено тестирование разработанного программного обеспечения на примере реальных данных, оценена его эффективность и выявлены области для дальнейшего совершенствования.

1 Анализ оптимизации управления запасами

1.1 Описание ABC-анализа

Чтобы определить наиболее важные компоненты инвентаря компании, многими используется ABC-анализ, метод классификации и управления запасами. Он основан на «правиле 80/20», или принципе Парето, который утверждает, что 20% причин приводит к 80% результатов [15]. Основная цель ABC-анализа состоит в том, чтобы группировать товары или услуги в соответствии с тем, насколько они важны для бизнеса. Термин «ABC» происходит от английских слов A (Absolute), B (Better) и C (Common), которые означают категории, в которые происходит классификация предметов [1].

В рамках ABC-анализа осуществляется разделение всех товаров на три группы с учетом их стоимости, объема продаж или других факторов, зависящих от конкретной ситуации [11].

Группа "А" включает в себя элементы с наибольшей значимостью, которые обычно составляют относительно небольшой процент от общего числа, но приносят основную часть выручки или имеют высокую важность. Это могут быть, например, высоко-маржинальные, быстро реализуемые или используемые в производстве ключевых продуктов товары.

Группа "В" включает в себя товары средней важности, которые занимают промежуточное положение между группами "А" и "С". Они имеют более низкую значимость по сравнению с товарами из группы "А", но все еще играют важную роль в деятельности организации. Товары этой группы могут включать в себя продукцию, которая регулярно используется, но не являются наиболее важными для бизнеса.

Группа "С" состоит из наименее важных элементов, которые вносят наименьший вклад в общий доход или имеют наименьшее значение. Эти товары обычно составляют большую часть ассортимента, но могут приносить небольшую прибыль или даже иметь отрицательную маржу.

Для начала классификации необходимо вычислить важность каждого товара или услуги. Обычно это делается на основе стоимости (цены) каждого элемента. Пусть C_i будет стоимостью товара I , а T – общей стоимостью всех товаров. Тогда доля стоимости товара i от общей стоимости будет равна P_i по формуле (1).

$$P_i = \frac{C_i}{T}. \quad (1)$$

После того, как все товары отсортированы по убыванию их доли прибыли, происходит их разделение на группы "А", "В" и "С" на основе кумулятивной доли значимости товара [17]. Кумулятивная доля значимости товара подсчитывается по формуле (2).

$$K_i = \sum_{j=1}^i \left(\left(\frac{Q_j * P_j}{\sum_{k=1}^n (Q_k * P_k)} \right) * 100\% \right). \quad (2)$$

Некоторые изменяют способы распределения товаров по группам, но стандартный способ распределения выглядит следующим образом.

Группа "А" – Товары составляющие 80% значимости, обычно их 20%.

Группа "В" – Товары составляющие 15% значимости, обычно их 30%.

Группа "С" – Товары составляющие 5% значимости, обычно их 50%.

Пример.

Представим, что у нас есть 100 различных товаров на складе.

Отсортируем их по общей полученной от них выгоде в порядке убывания.

Товар-1 – \$100, Товар 2 – \$80, Товар 3 – \$60 ... Товар 100 – \$5.

Тогда кумулятивный процент важности для каждого товара можно подсчитать и получить такие результаты.

Товар-1 – 20%, Товар 2- 36%, Товар 3- 48% ... Товар 100 – 100%.

Товары 1 и 2 войдут в категорию "А", товар 3 в группу "В", а все остальные товары – в категорию "С".

На рисунке 1 изображена кривая показывающая наиболее распространенное распределение товаров по категориям А, В и С. В дальнейшем это распределение можно будет использовать для различных действий по повышению эффективности предприятия [16].

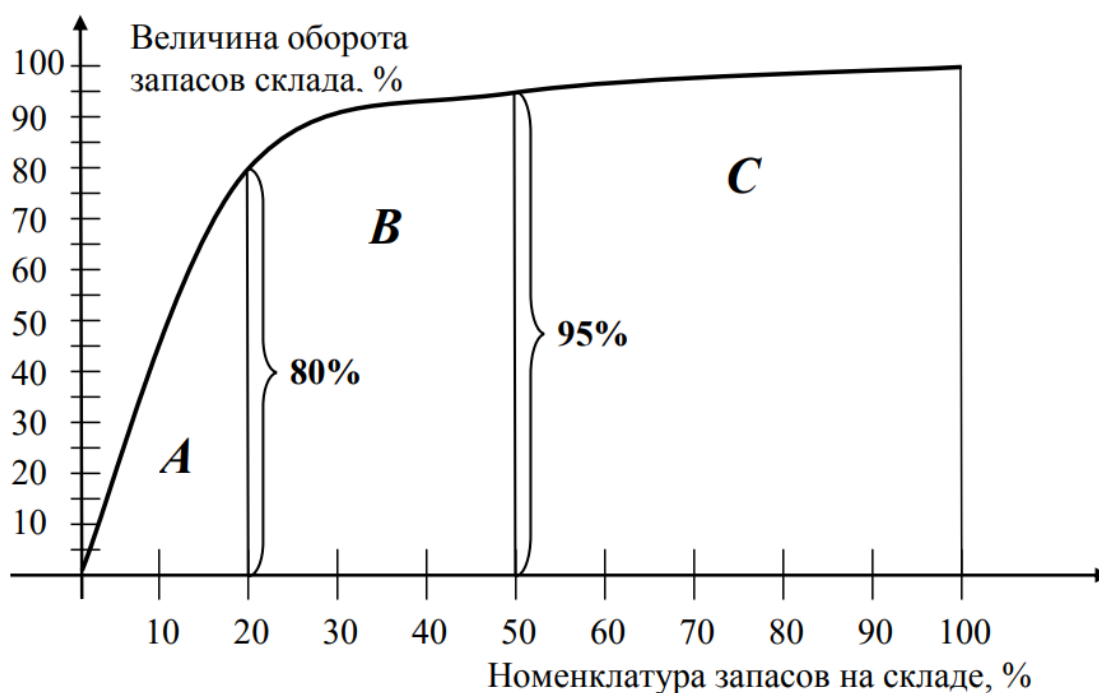


Рисунок 1 – Кривая, отображающая отношение оборота товаров

1.2 Применение ABC-анализа в различных сферах

ABC-анализ – отличное средство управления запасами используемое во многих областях. Используется оно широко из-за своей гибкости и простоты в реализации. Рассмотрим некоторые примеры.

«Производственная промышленность использует ABC-анализ для оптимизации управления запасами сырья, деталей и готовой продукции. Планирование закупок, производственных процессов и распределения ресурсов может быть более эффективным, если знать, какие материалы и

компоненты наиболее важны [14]. Например, наиболее важные компоненты могут подвергаться более тщательному контролю и управлению, чтобы не допустить их недостатка или избытка» [21].

В розничной торговле ABC-анализ используется для контроля ассортимента продукции на полках магазинов. Знание того какие товары приносят наибольший доход позволяет оптимизировать их размещение в магазине, поддерживать оптимальный уровень их запасов и предлагать покупателям наиболее прибыльные товары [9].

«В области услуг ABC-анализ может применяться для управления номенклатурой контрагентов или проектов. Зная, какие клиенты приносят наибольший доход или какие проекты наиболее прибыльны, можно сосредоточить усилия и ресурсы на наиболее важных областях. Это помогает выявить клиентов или проекты, которым требуется дополнительное внимание, или скорректировать стратегию работы с ними» [25].

ABC-анализ может применяться и в медицине для управления запасами медицинских товаров и лекарств. Знание того, какие материалы или лекарства используются чаще всего позволяет оптимизировать их приобретение и хранение. Это помогает предотвратить ситуации нехватки важных или избытка ненужных медицинских ресурсов [12].

«В области логистики и транспортировки ABC-анализ используется для управления запасами и распределением товаров. Знание того, какие грузы являются наиболее важными или приносят наибольший доход, позволяет оптимизировать их транспортировку, хранение и обработку [18]. Это помогает снизить затраты на хранение и транспортировку, а также повысить скорость работы» [22].

Эти примеры демонстрируют широкие возможности применения ABC-анализа в различных сферах бизнеса и деятельности. Этот инструмент помогает организациям лучше понимать и оптимизировать свои процессы, повышая тем самым эффективность управления запасами и общую производительность.

1.3 Преимущества и ограничения ABC-анализа

ABC-анализ является отличным инструментом управления запасами. Далее описание некоторых его характеристик.

Основными преимуществами ABC-анализа являются.

Оптимизация ресурсов – при использовании данного метода компании могут оптимизировать управление ресурсами сосредоточив свои усилия и инвестиции на наиболее прибыльных и важных товарах и услугах, что в свою очередь позволит улучшить эффективность работы и повысить прибыльность [13].

Простота и ясность – ABC-анализ относительно простой и понятный метод, который легко применять и использовать в самых разных областях. Основные инструменты данного метода понятны даже непрофессионалам что делает его широко используемым [2].

Гибкость и адаптивность – поскольку ABC-анализ легок в модифицировании, ведь его можно усложнять и модифицировать дополнительными факторами чуть ли не до бесконечности, он позволяет компаниям адаптировать свою стратегию управления запасами к меняющимся рыночным условиям, потребностям клиентов и внутренним факторам.

Основными ограничениями ABC-анализа являются.

Упрощенный подход – одним из основных ограничений ABC-анализа является упрощенный подход к классификации товаров и услуг. Сезонность, период спроса, технологические изменения не принимаются стандартным ABC-анализом во внимание, для них нужно дополнять этот метод другими или модифицировать его, что подходит не всем [4].

Статичность анализа – ABC-анализ применяет данные за определенное время без приоритизации определенной информации, что может сделать его использование неподходящим для условий крайне быстрого изменения рынка. В качестве примера можно привести игрушки, детей интересуют в основном игрушки популярные в текущем информационном поле (популярные

мультфильмы и игры), поэтому для них невозможно будет предугадать спрос и сделать правильное распределение по группам А, В и С [3].

Несмотря на ограничения ABC-анализ остается одним из самых популярных и широко используемых инструментов управления запасами благодаря своей простоте, гибкости и способности приносить значительные выгоды бизнесу при относительно легком процессе внедрения.

1.4 Описание дополнений метода ABC-анализа

ABC – анализ имеет ряд методов дополнений, расширяющих возможности анализа, они широко используются в различных сферах. Рассмотрим некоторые.

XYZ-анализ – метод классификации товаров в соответствии с предсказуемостью спроса. Товары делятся на три категории по степени изменчивости.

Группа X (Высокая предсказуемость) – товары с высокой степенью предсказуемости спроса. Это товары, спрос на которые стабилен и предсказуем.

Группа Y (Средняя предсказуемость) – товары, спрос на которые умеренно предсказуем, но есть некоторые колебания.

Группа Z (Низкая предсказуемость) – товары с низкой степенью предсказуемости спроса. Для этих товаров характерны значительные колебания продаж.

XYZ-анализ помогает оптимизировать управление запасами, прогнозировать спрос на товары и снижать риски связанные с нехваткой или избыточными запасами.

VED-анализ классифицирует продукты в соответствии с их жизненным циклом и бизнес-критичностью. Этот метод делит товары на три категории V, E и D.

Vital (Критические) – продукты, критически важные для бизнеса. Их отсутствие может привести к серьезным последствиям, таким как остановка производственных процессов или потеря клиентов.

Essential (Существенные) – товары, которые важны для бизнеса, но их отсутствие не приведет к критическим последствиям. Они необходимы для нормальной работы, но их можно заменить или временно обойтись без них.

Desirable (Желательные) – товары, которые желательны, но не критичны для бизнеса. Их отсутствие не оказывает существенного влияния на бизнес-процессы и результаты.

VED-анализ помогает организациям определить стратегию управления запасами и расставить приоритеты в отношении важности товаров.

Каждый из этих методов имеет свои преимущества, поэтому их выбор зависит от конкретных потребностей и особенностей работы. Далее мы рассмотрим более подробно метод XYZ-анализа поскольку он является отличным дополнением к методу ABC-анализа [5].

1.5 Описание XYZ-анализа

XYZ-анализ – метод управления запасами, который является дополнением к методу ABC-анализа, он используется для классификации товаров в соответствии с предсказуемостью их спроса. Суть метода заключается в разделении товаров на три категории X, Y и Z в зависимости от степени изменчивости спроса [10].

Для классификации товаров по категориям XYZ используются статистические показатели, такие как среднеквадратичное отклонение и коэффициент вариации.

Начнем со среднеквадратичного отклонения, проведения XYZ-анализа нужно подсчитать его для каждого товара.

Среднеквадратичное отклонение (σ) – мера изменчивости данных относительно их среднего значения. Для расчета среднеквадратичного отклонения воспользуемся формулой (3).

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}, \quad (3)$$

где x_i – значения в выборке;

\bar{x} – среднее значение выборки;

n – количество элементов в выборке.

Коэффициент вариации (V) измеряет степень изменчивости данных. Формула для вычисления коэффициента вариации выглядит следующим образом.

$$V = \frac{\sigma}{\bar{x}}, \quad (4)$$

где σ – среднеквадратичное отклонение;

\bar{x} – среднее значение выборки.

После вычисления коэффициента вариации товары классифицируются по следующему принципу.

Если $V < 10\%$, то товар относится к категории X.

Если $10\% \leq V < 30\%$, то товар относится к категории Y.

Если $V \geq 30\%$, то товар относится к категории Z.

Вот так XYZ-анализ позволяет распределить товары по степени колебания спроса на них.

1.6 Применение XYZ-анализа

XYZ-анализ находит широкое применение в различных отраслях и сферах деятельности. Рассмотрим некоторые из них.

«В производственных компаниях XYZ-анализ используется для оптимизации производственных процессов и управления запасами сырья и готовой продукции. Это помогает минимизировать производственные издержки и улучшить качество продукции» [19].

В ритейле XYZ-анализ применяется для управления ассортиментом и запасами товаров. Компании могут более точно прогнозировать спрос на различные товары и оптимизировать закупки, что способствует снижению затрат и улучшению обслуживания клиентов [20].

«Логистические компании и операторы цепочек поставок используют XYZ-анализ для оптимизации маршрутов доставки, управления складскими запасами и планирования поставок. Это помогает снижать транспортные расходы и улучшать оперативную эффективность» [23].

«В сфере здравоохранения XYZ-анализ может быть применен для управления запасами медикаментов и медицинского оборудования. Это позволяет медицинским учреждениям обеспечивать постоянную готовность к оказанию помощи и снижать затраты на хранение и закупки» [24].

В финансовом секторе XYZ-анализ используется для управления портфелями активов и анализа риска. Финансовые учреждения могут классифицировать активы по степени стабильности их доходности и риска, что помогает в разработке инвестиционных стратегий и управлении рисками [6].

В образовательных учреждениях XYZ-анализ может применяться для управления ресурсами и планирования учебного процесса. Например, школы и университеты могут классифицировать учебные материалы и оборудование по степени их использования и необходимости, что помогает оптимизировать расходы и улучшить качество образования.

1.7 Постановка задачи

Для разработки программного обеспечения для рационального размещения товаров на складе на основе ABC-анализа, необходимо сформулировать математическую модель для задачи оптимизации. Для начала следует описать условия.

Есть товары, у каждого товара есть характеристики – коэффициент прибыли от товара и коэффициент нестабильности спроса. Также есть склад с неким количеством мест. Поскольку разместить нужно все товары, а складское пространство полностью занимать не обязательно, представим его как количество мест для размещения товара (равное количеству товара) по удаленности от входа. Необходимо разместить товар по принципу прибыльный и стабильный у входа. Поскольку это делается для сокращения общего времени обработки всего товара делаем вывод что нам нужно минимизировать время. Опишем целевую функцию (5).

$$\text{Min } T = \sum_{i=1}^n \sum_{j=1}^n P_i * D_j * U_i * x_{ij}, \quad (5)$$

где n – количество товаров;

P_i – прибыль от товара i , где $i \in \{1, 2, \dots, n\}$;

D_j – расстояние позиции j от входа на склад, $j \in \{1, 2, \dots, n\}$;

U_i – коэффициент нестабильности спроса для товара i ;

x_{ij} – переменная отвечающая за размещение товара i на позиции j , она равна 1 если товар размещен и 0 если нет.

Опишем ограничения модели.

Поскольку каждый товар должен быть размещен на какой-либо позиции, ограничение (6).

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in \{1, 2, \dots, n\}. \quad (6)$$

Поскольку каждая позиция должна содержать только один товар, ограничение (7).

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in \{1, 2, \dots, n\}. \quad (7)$$

Итоговая модель (8).

$$\begin{aligned} \text{Min } T &= \sum_{i=1}^n \sum_{j=1}^n P_i * D_j * U_i * x_{ij}, \\ \sum_{j=1}^n x_{ij} &= 1 \quad \forall i \in \{1, 2, \dots, n\}, \\ \sum_{i=1}^n x_{ij} &= 1 \quad \forall j \in \{1, 2, \dots, n\}, \\ x_{ij} &\in \{0, 1\}. \end{aligned} \quad (8)$$

Итоговая модель учитывает параметры прибыльности и стабильности спроса товаров. Далее будет разработка алгоритма для решения и программная реализация.

2 Разработка алгоритма

2.1 Выбор языка программирования и среды разработки

Для реализации программного обеспечения мною был выбран язык программирования C++. Для разработки на данном языке мною была выбрана среда разработки Visual Studio.

C++ – язык программирования который обеспечивает быструю работу. Это особенно важно для анализа больших объемов данных которые могут возникнуть при применении ABC-XYZ-анализа в крупных компаниях. Благодаря возможностям C++ можно оптимизировать производительность программы для эффективной обработки поистине колоссальных объемов данных, что определенно пригодится при последующем расширении алгоритма. К тому же C++ предоставляет широкий спектр инструментов и стандартных библиотек которые будут крайне полезны в дальнейшем.

C++ имеет большое сообщество разработчиков из-за чего в интернете находится множество библиотек и инструментов, разработанных посторонними людьми, которые будут полезны при реализации любых проектов. Они позволяют не тратить время на создание своего решения уже возникшей проблемы и к тому-же большинство их находятся в свободном доступе.

Visual Studio – это интегрированная среда разработки (IDE) с широким набором инструментов для разработки на C++. Она предоставляет удобный интерфейс для написания, отладки и тестирования программ что делает процесс разработки удобным и быстрым. Кроме того, Visual Studio обеспечивает интеграцию с другими инструментами разработки, такими как плагин vskrp, что упрощает работу над проектами [7].

Подводя итог, C++ и Visual Studio оправдан для реализации алгоритмов ABC-анализа эффективной работой с большими объемами данных, наличием

широкого набора инструментов и возможностью удобной интеграции плагинов и библиотек при разработке.

2.2 Библиотеки для работы с Excel файлами

Рассмотрим библиотеки для работы с Excel на C++ и выделим их особенности и недостатки.

Libxl – одна из самых популярных библиотек для работы с Excel на языке C++. Обеспечивает хорошую поддержку форматов xls иxlsx и предлагает широкий спектр функций для чтения и записи данных в Excel. Однако libxl требует много времени для освоения и использования.

XlsxWriter – библиотека для создания файлов Excel на языке программирования C. Она предоставляет простой интерфейс для создания и форматирования таблиц, добавления данных и создания диаграмм. Однако XlsxWriter не может читать файлы Excel.

C++ Excel Writer – крайне схожая с XlsxWriter библиотека, но для C++. И неподходящая по той же причине.

Я выбрал библиотеку xlnt для реализации, потому что это отличный вариант для реализации алгоритма ABC-анализа через Excel по многим причинам.

Библиотека xlnt специально разработана исключительно для работы с файлами формата Excel (.xlsx). Это означает, что она позволяет легко читать данные из существующих документов Excel и создавать новые документы Excel с результатами анализа без необходимости использования дополнительных инструментов или преобразования данных и не будет перегружена лишним, ненужным функционалом.

Библиотека xlnt имеет простой и интуитивно понятный интерфейс программирования, с xlnt можно легко создавать, изменять и анализировать данные в Excel используя всего несколько строк кода что упрощает различные операции с данными и не ограничивает в разработке.

xlnt высокой производительности, что позволяет управлять большими наборами данных в Excel с минимальными задержками и затратами времени, а это особенно важно при выполнении сложных многокритериальных задач как ABC-XYZ-анализ [8].

В целом, библиотека xlnt является удобным инструментом обработки данных в Excel и идеальным выбором для реализации моего будущего алгоритма.

2.3 Определение требований к алгоритму

Далее будет описано создание алгоритма для программного обеспечения, предназначенного для рационального размещения товаров на складе на основе ABC-анализа. Итоговый алгоритм будет представлять последовательность шагов и процедур программы благодаря которой будет обеспечено эффективное управление запасами для оптимизации складских операций и улучшения логистических процессов.

Будущая программа будет выполнять указанные далее функции.

Сбор и обработка данных о товарах – в начале работы программа будет загружать данные о товарах из указанного Excel файла формата .xlsx, содержащего информацию о: продажах, стоимости, количестве, габаритах и других параметрах товара.

Выполнение ABC-анализа – на основе данных о продажах и стоимости программа будет классифицировать товары на категории А, В и С, определяя их относительную важность.

Выполнение XYZ-анализа – программа также будет выполнять анализ стабильности спроса на товары, разделяя их на категории X, Y и Z.

Сортировка товаров – после выполнения анализа товары будут сортироваться по категориям ABC и XYZ.

Размещение товаров на складе – программа будет распределять товары по складу с учетом их важности и габаритов, оптимизируя использование складского пространства.

Запись результатов – все результаты анализа и размещения будут сохраняться в новом Excel-файле для дальнейшего использования и анализа.

2.4 Создание алгоритма

Порядок работы программы.

Загрузка данных. Программа запрашивает у пользователя имя файла, содержащего данные о товарах. Загружает данные из указанного Excel-файла с помощью библиотеки xlnt.

Преобразует данные в объекты структуры, включающие название, цену, количество проданных товаров по кварталам, идентификатор, размеры и другие параметры.

Выполнение ABC-анализа. Вычисляется общая выгода от продажи каждого товара. Товары сортируются по убыванию их вклада в общую прибыль. Затем идет распределение товаров по категориям А, В и С на основе кумулятивного процента прибыли: товары, составляющие 80% прибыли попадают в категорию А, следующие 15% – в категорию В, оставшиеся 5% – в категорию С.

Выполнение XYZ-анализа. Для каждого товара вычисляется коэффициент вариации (CV) на основе стабильности продаж по кварталам. Товары классифицируются по категориям X, Y и Z в зависимости от значения CV: товары с $CV < 10\%$ попадают в категорию X, с CV от 10% до 30% – в категорию Y, с $CV > 30\%$ – в категорию Z.

Сортировка товаров. Товары повторно сортируются по категориям ABC и XYZ, а затем по прибыли. Программа использует специальные функции сравнения для упорядочивания объектов в соответствии с заданными критериями.

Запись результатов.

Создается новый Excel файл, в который записываются все данные о товарах, включая результаты ABC и XYZ-анализов. В файл storage.txt записывается информация о распределении товаров по складским зонам, включая занятые и свободные площади.

Практическая польза алгоритма.

Оптимизация ресурсов – ABC-анализ позволяет сосредоточить усилия и ресурсы на наиболее важных товарах, что снижает затраты на хранение и управление запасами.

Улучшение обслуживания клиентов – правильное размещение товаров на складе ускоряет процессы поиска и доставки, что улучшает скорость обслуживания клиентов.

Снижение затрат – XYZ-анализ помогает прогнозировать спрос на товары и сокращать расходы, связанные с избыточными запасами или нехваткой товаров.

Эффективное использование складского пространства – алгоритм позволяет максимально эффективно использовать доступное складское пространство, размещая товары таким образом, чтобы минимизировать количество пространств с пустыми областями и обеспечить быстрый доступ к наиболее важным товарам.

Разработанный алгоритм представляет собой комплексную систему управления запасами и оптимизации складских операций. Она позволяет эффективно распределять товар по складу на основе анализа его значимости и стабильности спроса, что способствует повышению общей эффективности и рентабельности деятельности предприятия.

3 Реализация и тестирование программного обеспечения

3.1 Реализация ПО

Разберем программную реализацию алгоритма.

В начале описываю раздел отвечающий за подключение библиотек и пространств имён, необходимых для работы программы, их можно наблюдать на рисунке 3.

```
2  ✓ #include <xlnt/xlnt.hpp>
3      #include <vector>
4      #include <algorithm>
5      #include <iostream>
6      #include <string>
7      #include <fstream>
8      #include <cmath>
9      #include <conio.h>
10
11  using namespace std;
```

Рисунок 3 – Пространство имен и библиотеки используемые в программе

`xlnt/xlnt.hpp` – библиотека для работы с Excel файлами [8].

`vector` – библиотека для хранения данных в виде динамических массивов.

`algorithm` – библиотека, содержащая функции для работы с данными, например, сортировка.

`iostream`, `fstream` – библиотеки для ввода/вывода данных через стандартные потоки и файлы.

`string` – библиотека для работы со строками.

`cmath` – библиотека для математических функций, например, вычисление квадратного корня.

`conio.h` – библиотека для работы с консольным вводом/выводом.

using namespace std – указывает компилятору использовать стандартное пространство имен, чтобы не писать std:: перед типами данных и функциями из стандартной библиотеки.

Следом идет структура, которая предназначена для хранения информации о товарах, ее можно наблюдать на рисунке 4.

```
13 // Структура для хранения информации о товаре
14 struct Subject {
15     string name; // Название товара
16     int price_sold; // Цена продажи
17     int price_purchase; // Цена покупки
18     int quantity_1; // Количество продаж по кварталам
19     int quantity_2;
20     int quantity_3;
21     int quantity_4;
22     int id; // Идентификатор товара
23     int count; // Общее количество
24     int width; // Размеры товара
25     int height;
26     int benefit; // Выгода от продажи
27     string ABC_category; // Категория ABC-анализа
28     string XYZ_category; // Категория XYZ-анализа
29
30     int getId() const {
31         return id;
32     }
33
34     int getCount() const {
35         return count;
36     }
37
38     int getWidth() const {
39         return width;
40     }
41
42     int getHeight() const {
43         return height;
44     }
45 };
```

Рисунок 4 – Структура Subject

В данной структуре хранятся следующие данные о товаре: название, цена, количество проданных товаров по кварталам, идентификатор, общее количество, размеры, выгода от продажи и категории ABC и XYZ.

Еще там присутствуют функции для доступа к некоторым полям структуры, что упрощает получение данных без прямого доступа к полям структуры.

Далее на рисунке 5 представлена функция XYZ_analysis, она анализирует стабильность продаж товара по кварталам.

```
47 // Анализ XYZ для определения стабильности продаж товара
48 string XYZ_analysis(int q1, int q2, int q3, int q4) {
49     vector<int> data = { q1, q2, q3, q4 };
50     double sum = 0;
51     for (int i = 0; i < data.size(); i++) {
52         sum += data[i];
53     }
54     double mean = sum / data.size();
55
56     double sq_diff_sum = 0;
57     for (int i = 0; i < data.size(); i++) {
58         sq_diff_sum += (data[i] - mean) * (data[i] - mean);
59     }
60     double variance = sq_diff_sum / data.size();
61     double std_dev = sqrt(variance);
62
63     double cv = std_dev / mean;
64
65     if (cv < 0.1) {
66         return "X";
67     }
68     else if (cv >= 0.1 && cv <= 0.3) {
69         return "Y";
70     }
71     else {
72         return "Z";
73     }
74 }
```

Рисунок 5 – Функция для определения XYZ-категории

Сначала в данной функции рассчитывается среднее значение продаж. Затем вычисляется стандартное отклонение. Коэффициент вариации (CV) используется для классификации товара в одну из трех категорий: X (самый стабильный), Y (со средней стабильностью), Z (наименее стабильный).

Следом на рисунке 6 идет класс Space.

```
76 // Класс для управления пространством склада
77 class Space {
78 private:
79     int width;// Размеры пространства
80     int height;
81     vector<vector<int>> space;// Матрица занятости
82     vector<int> subjectIds;// Список ID размещённых товаров
83
84 public:
85     Space(int w, int h) : width(w), height(h) {
86         space = vector<vector<int>>(width, vector<int>(height, 0));
87     }
88     // Функция проверки возможности размещения товара
89     bool canFit(int x, int y, int a, int b) const {
90         if (x + a > width || y + b > height)
91             return false;
92
93         for (int i = x; i < x + a; ++i) {
94             for (int j = y; j < y + b; ++j) {
95                 if (space[i][j] == 1)
96                     return false;
97             }
98         }
99
100         return true;
101     }
102     // Функция размещения товара
103     void fillSpace(int x, int y, int a, int b, int subjectId) {
104         for (int i = x; i < x + a; ++i) {
105             for (int j = y; j < y + b; ++j) {
106                 space[i][j] = 1;
107             }
108         }
109         subjectIds.push_back(subjectId);
110     }
111
112     vector<vector<int>> getSpace() const {
113         return space;
114     }
115
116     vector<int> getSubjectIds() const {
117         return subjectIds;
118     }
119 };
```

Рисунок 6 – Класс Space

Класс Space управляет пространством склада, проверяет возможность размещения товара на заданных координатах и размерах. Размещает товар в пространстве, отмечая его как занятое. Возвращает текущее состояние занятости пространства и ID размещенных товаров.

На рис. 7 изображены функции сравнения классов используемые в дальнейшем для сортировки объектов.

```
bool compareSubjects_benefit(const Subject& subject1, const Subject& subject2) {
    return subject1.benefit > subject2.benefit;
}

bool compareSubjects(const Subject& subject1, const Subject& subject2) {
    if (subject1.ABC_category != subject2.ABC_category) {
        return subject1.ABC_category < subject2.ABC_category;
    }
    else if (subject1.XYZ_category != subject2.XYZ_category) {
        return subject1.XYZ_category < subject2.XYZ_category;
    }
    else {
        return subject1.benefit > subject2.benefit;
    }
}
```

Рисунок 7 – Функции сравнения объектов по параметрам

Далее опишем все составляющие main, ее начало изображено на рисунке 8.

```
139 int main() {
140     setlocale(LC_ALL, "Russian");
141     cout << "Здравствуйте, введите название файла к которому нужно применить обработку " << endl;
142     cout << "или введите 1 для стандартного названия subjects.xlsx:";
143     string file_read, file_write = "sorted.xlsx";
144     int benefit_sum = 0, tovar_sum = 0;
145     string buf = "1";
146     cin >> buf;
147     if (buf == "1") { file_read = "subjects.xlsx"; }
148     else { file_read = buf; }
```

Рисунок 8 – Начало main

Программа устанавливает русскую локаль для корректного отображения затем пользователю предлагается ввести имя файла, который нужно обработать. Введенное имя файла сохраняется в переменной buf. Если пользователь вводит "1", используется стандартное имя файла (subjects.xlsx).

Далее загружаются данные из Excel файла с помощью библиотеки xInt, как это сделано видно на рисунке 9. Данные каждой строки файла читаются и преобразуются в объекты структуры Subject. Это позволяет далее удобно работать с данными товаров в программе.

```
150     xInt::workbook wb;//открытие Excel файла
151     wb.load(file_read);
152     auto ws = wb.active_sheet();
153
154     vector<Subject> subjects;
155
156     auto rows = ws.rows(false);
157     auto row_it = std::next(rows.begin());//Переход к второй строчке
158     for (; row_it != rows.end(); ++row_it) {
159         auto row = *row_it;
160         Subject subjectV;
161         //Забор данных
162         subjectV.name = row[0].to_string();
163         subjectV.price_sold = row[1].value<int>();
164         subjectV.price_purchase = row[2].value<int>();
165         subjectV.quantity_1 = row[3].value<int>();
166         subjectV.quantity_2 = row[4].value<int>();
167         subjectV.quantity_3 = row[5].value<int>();
168         subjectV.quantity_4 = row[6].value<int>();
169         subjectV.id = row[7].value<int>();
170         subjectV.count = row[8].value<int>();
171         subjectV.width = row[9].value<int>();
172         subjectV.height = row[10].value<int>();
173         subjectV.benefit = (subjectV.price_sold - subjectV.price_purchase) *
174         (subjectV.quantity_1+ subjectV.quantity_2 + subjectV.quantity_3 + subjectV.quantity_4);
175         benefit_sum += subjectV.benefit;
176         //Запись в объект
177         subjects.push_back(subjectV);
178         tovar_sum++;
179     }
180 }
```

Рисунок 9 – Загрузка данных из Excel файла

Важные поля каждого товара, такие как цены, количества по кварталам, размеры и ID, заполняются из Excel. Выгода от продажи каждого товара рассчитывается как разница между ценой продажи и покупки, умноженная на общее количество продаж.

Переменная `benefit_sum` накапливает общую выгоду от всех товаров, что важно для последующего ABC-анализа.

Теперь сортируем объекты по прибыли. После чего каждый товар классифицируется по категориям А, В, и С на основе его кумулятивной доли в общей выгоде. В товары категории А записываются объекты кумулятивная доля прибыли которых менее 80%, в товары категории В записываются объекты кумулятивная доля прибыли которых менее 95%, остальным присваивается категория С.

Параллельно вызывается функция в которой проводится XYZ-анализ для каждого товара, что помогает определить стабильность его продаж по кварталам.

После чего объекты еще раз сортируются, но уже не по прибыли. Сначала по ABC критерию, затем по XYZ критерию и в конце по прибыли, с этим помогает указанная ранее функция `compareSubjects`.

Как происходят эти действия можно наблюдать на рисунке 10.

```
181 //Сортировка по столбцу "прибыль"
182 sort(subjects.begin(), subjects.end(), compareSubjects_benefit);
183 //ABC анализ в группы
184 int j=0,benefit_stack=0;
185 for (j = 0; j < tovar_sum;j++) {
186     if (benefit_stack <= (benefit_sum*0.8)) {
187         subjects[j].ABC_category = "A";
188     }
189     else if (benefit_stack <= (benefit_sum * 0.95)) {
190         subjects[j].ABC_category = "B";
191     }
192     else {
193         subjects[j].ABC_category = "C";
194     }
195     benefit_stack += subjects[j].benefit;
196     subjects[j].XYZ_category = XYZ_analysis(subjects[j].quantity_1,
197     subjects[j].quantity_2, subjects[j].quantity_3, subjects[j].quantity_4);
198 }
199
200 //Сортировка объектов
201 sort(subjects.begin(), subjects.end(), compareSubjects);
```

Рисунок 10 – Сортировка и присвоение товарам категорий ABC и XYZ

Затем создается новая Excel таблица в которую заносится все данные из той, которую мы обрабатывали и новые значения которые мы получили, такие как ABC категория, XYZ категория и общая выгода. Исполнение этого на рисунке 11.

```
203     xlsx::workbook new_wb; //Создание новой Excel таблицы
204     auto new_ws = new_wb.active_sheet();
205
206     int row = 2;
207
208     for (const auto& subject : subjects) { //Заполнение Excel файла
209         new_ws.cell("A" + std::to_string(row)).value(subject.name);
210         new_ws.cell("B" + std::to_string(row)).value(subject.price_sold);
211         new_ws.cell("C" + std::to_string(row)).value(subject.price_purchase);
212         new_ws.cell("D" + std::to_string(row)).value(subject.quantity_1);
213         new_ws.cell("E" + std::to_string(row)).value(subject.quantity_2);
214         new_ws.cell("F" + std::to_string(row)).value(subject.quantity_3);
215         new_ws.cell("G" + std::to_string(row)).value(subject.quantity_4);
216         new_ws.cell("H" + std::to_string(row)).value(subject.id);
217         new_ws.cell("I" + std::to_string(row)).value(subject.count);
218         new_ws.cell("J" + std::to_string(row)).value(subject.width);
219         new_ws.cell("K" + std::to_string(row)).value(subject.height);
220         new_ws.cell("L" + std::to_string(row)).value(subject.ABC_category);
221         new_ws.cell("M" + std::to_string(row)).value(subject.XYZ_category);
222         new_ws.cell("N" + std::to_string(row)).value(subject.benefit);
223         row++;
224     }
225     new_ws.cell("A1").value("Name");
226     new_ws.cell("B1").value("Price_sold");
227     new_ws.cell("C1").value("Price_purchase");
228     new_ws.cell("D1").value("Quantity_1");
229     new_ws.cell("E1").value("Quantity_2");
230     new_ws.cell("F1").value("Quantity_3");
231     new_ws.cell("G1").value("Quantity_4");
232     new_ws.cell("H1").value("ID");
233     new_ws.cell("I1").value("Count");
234     new_ws.cell("J1").value("Width");
235     new_ws.cell("K1").value("Height");
236     new_ws.cell("L1").value("ABC_Category");
237     new_ws.cell("M1").value("XYZ_Category");
238     new_ws.cell("N1").value("Benefit");
239
240     new_wb.save(file_write);
```

Рисунок 11 – Запись отредактированных данных в новую таблицу

Затем, используя ifstream, программа пытается открыть файл storage_properties.txt, который содержит параметры складских пространств. Если файл не удастся открыть программа выводит сообщение об ошибке.

Программа считывает из файла количество пространств (spaceCount), которое определяет, сколько раз ей нужно будет выполнить следующий шаг.

В цикле программа читает координаты каждого пространства (ширину и высоту) и создает объекты Space, которые она сохраняет в векторе spaces. Эти пространства будут использоваться для распределения по ним товаров на складе.

После чтения всех данных файл закрывается для освобождения системных ресурсов. Весь процесс считывания изображен на рисунке 12.

```
242 //Открытие файла с параметрами
243 ifstream inFile("storage_properties.txt");
244 if (!inFile) {
245     cerr << "Error: Unable to open file." << endl;
246     return 1;
247 }
248
249 int spaceCount;
250 inFile >> spaceCount;
251
252 vector<Space> spaces;
253 for (int i = 0; i < spaceCount; ++i) {
254     int spaceX, spaceY;
255     inFile >> spaceX >> spaceY;
256     spaces.push_back(Space(spaceX, spaceY));
257 }
258
259 inFile.close();
```

Рисунок 12 – Считывание данных о складе

После считывания программа распределяет товары по пространствам на складе, реализация на рисунке 13.


```

261 //начало распределения на складе
262 for (const auto& subject : subjects) {
263     for (int count = 0; count < subject.getCount(); ++count) {
264         bool subjectAdded = false;
265         for (auto& space : spaces) {
266             for (int i = 0; i < space.getSpace().size(); ++i) {
267                 for (int j = 0; j < space.getSpace()[0].size(); ++j) {
268                     if (space.canFit(i, j, subject.getWidth(), subject.getHeight())) {
269                         space.fillSpace(i, j, subject.getWidth(), subject.getHeight(), subject.getId());
270                         subjectAdded = true;
271                         break;
272                     }
273                     if (space.canFit(i, j, subject.getHeight(), subject.getWidth())) {
274                         space.fillSpace(i, j, subject.getHeight(), subject.getWidth(), subject.getId());
275                         subjectAdded = true;
276                         break;
277                     }
278                 }
279                 if (subjectAdded)
280                     break;
281             }
282             if (subjectAdded)
283                 break;
284         }
285     }
286     if (!subjectAdded) {
287         break;
288     }
289 }
290 }

```

Рисунок 13 – Распределение товаров на складе

Для каждого товара (subject) программа пытается разместить его на складе в зависимости от его количества (getCount()).

Для каждого товара программа проверяет каждую возможную позицию в каждом складском пространстве, чтобы увидеть, можно ли разместить товар там, не нарушая уже занятые площади.

Если находится подходящее место, товар добавляется в это место (fillSpace()), и цикл прерывается для этого товара. Если товар можно разместить повернув (поменяв местами ширину и длину), товар помещается повернутым.

Если для товара не нашлось места, цикл распределения для данного товара прекращается, предотвращая дальнейшие попытки.

После это происходит запись результатов распределения в текстовый файл, как это сделано можно увидеть на рисунке 14.

```

292     ofstream outFile("storage.txt");
293     if (!outFile) {
294         cerr << "Error: Unable to open file." << endl;
295         return 1;
296     }
297
298     outFile << "Final space status:" << endl;
299     for (int i = 0; i < spaces.size(); ++i) {
300         outFile << "Space " << i + 1 << ":" << endl;
301         vector<vector<int>> currentSpace = spaces[i].getSpace();
302         for (const auto& row : currentSpace) {
303             for (int val : row) {
304                 outFile << val << " ";
305             }
306             outFile << endl;
307         }
308         outFile << "Subject IDs: ";
309         vector<int> subjectIds = spaces[i].getSubjectIds();
310         for (int id : subjectIds) {
311             outFile << id << " ";
312         }
313         outFile << endl;
314         outFile << endl;
315     }
316
317     outFile.close();
318
319     cout << "Обработка успешно применена\n";
320     system("pause");
321
322     return 0;
323 }

```

Рисунок 14 – Запись результата распределения товаров в текстовый файл

Используя `ofstream`, программа открывает файл `storage.txt` для записи результатов распределения товаров.

В файл записывается состояние каждого складского пространства, включая занятые и свободные площади, а также ID товаров, размещенных в каждом пространстве.

По завершении записи файл закрывается для сохранения данных.

После завершения всех операций, программа информирует пользователя о том, что все задачи были успешно выполнены.

Команда `system("pause")` используется для задержки закрытия консольного окна, позволяя пользователю увидеть сообщение о завершении.

Эта программа на основе файла `storage_properties.txt` с данными зон склада и файла `subjects.xlsx` с данными о товарах предприятия производит выполнение ABC-анализа, XYZ-анализа, последующую сортировку товаров по этим критериям, запись отсортированных значений в новую таблицу вместе с новыми подсчитанными параметрами и в конце проводит распределения товаров по складским зонам в зависимости от их важности и габаритов, на выходе имеем файл `sorted_subjects.xlsx` с данными товаров и файл `storage.txt` с данными о распределении этих товаров.

Далее рассмотрим работу программы на примере.

3.2 Тестирование ПО

Для тестирования программы как ранее было сказано нам понадобится два дополнительных файла.

Первый файл `storage_properties.txt`. В нём указывается информация о складе. На первой строчке указывается количество зон для расположения товаров. Затем с каждой следующей строки считывается через пробел длина и ширина зоны для погрузки товаров, считывается столько раз сколько существующих зон было указано на первой строке файла. Файл `storage_properties.txt` который мы будем использовать для тестирования изображен на рисунке 15.

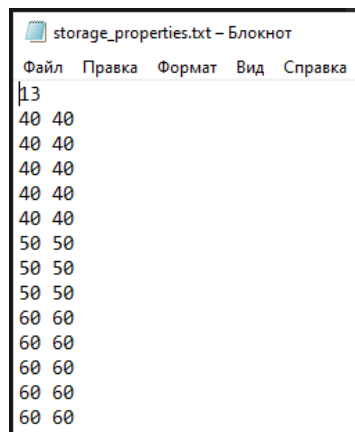


Рисунок 15 – Файл с данными о зонах для складирования

Затем нам нужен файл subjects.xlsx, в котором указаны данные о товарах, такие как продажи, остаток на складе, габариты, цена продажи и расходы на реализацию. Файл subjects.xlsx который мы будем использовать для тестирования изображен на рисунке 16.

	A	B	C	D	E	F	G	H	I	J	K
	Наименование товара	Цена продажи одной единицы товара	Расходы на хранение и реализацию одной единицы товара	Количество реализованных товаров в 1-ом квартале	Количество реализованных товаров в 2-ом квартале	Количество реализованных товаров в 3-ем квартале	Количество реализованных товаров в 4-ом квартале	ID товара	Количество товара на складе	Длина товара	Ширина товара
1											
2	Стул	50	10	22	30	24	32	1	24	4	4
3	Стол(мал.)	100	25	23	35	18	19	2	6	6	8
4	Стол(бол.)	150	37	27	22	16	18	3	17	8	12
5	Тумба	75	15	25	31	19	34	4	6	4	4
6	Люстра	75	17	18	15	30	14	5	17	4	5
7	Кровать(мал.)	200	43	25	28	34	20	6	18	18	10
8	Кровать(бол.)	300	78	31	26	13	24	7	5	22	18
9	Шкаф напольный	110	25	10	23	19	31	8	23	4	4
10	Шкаф настенный	120	35	13	16	12	27	9	10	6	3
11	Шкаф мебельный(мал.)	180	38	34	26	15	28	10	8	5	12
12	Шкаф мебельный(бол.)	220	64	23	24	29	33	11	15	8	16
13	Телевизор(мал.)	240	57	19	30	15	13	12	8	1	12
14	Телевизор(бол.)	360	96	10	34	34	29	13	6	2	18
15	Полка	50	15	20	13	13	17	14	22	2	4
16	Кресло	130	26	33	20	33	25	15	13	5	5
17	Диван(мал.)	220	49	27	11	23	12	16	9	18	10
18	Диван(бол.)	350	84	16	20	34	15	17	7	24	10
19	Шведская стенка	180	36	25	17	13	21	18	18	5	2
20	Обои	20	4	10	26	35	16	19	6	1	1
21	Линолеум	30	7	18	30	17	21	20	7	2	2

Рисунок 16 – Excel таблица с данными о товарах

Запустим программу. Результат работы можно увидеть на рисунке 17.

```

Консоль отладки Microsoft Visual Studio
Здравствуйтe, введите название файла к которому нужно применить обработку
или введите 1 для стандартного названия subjects.xlsx:1
Обработка успешно применена
Для продолжения нажмите любую клавишу . . .

C:\Users\hpc\Desktop\DeepLom\enforce\x64\Release\enforce.exe (процесс 22316) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:
    
```

Рисунок 17 – Результат работы программы

После запуска программа просит ввести название Excel файла или ввести «1» для стандартного имени файла subjects.xlsx. После успешной обработки выводится сообщение “Обработка успешно применена”. Затем в той же папке появятся файлы sorted_subjects.xlsx с информацией о товарах и storage.txt с информацией о распределении товаров на складе.

Проверим эти файлы, их можно увидеть на рисунках 18 и 19.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Name	Price_sold	Price_purc	Quantity_1	Quantity_2	Quantity_3	Quantity_4	ID	Count	Width	Height	ABC_Categ	XYZ_Categ	Benefit
2	Кровать(б	300	78	31	26	13	24	7	5	22	18	A	Y	20868
3	Шкаф меб	220	64	23	24	29	33	11	15	8	16	A	Y	17004
4	Кровать(м	200	43	25	28	34	20	6	18	18	10	A	Y	16799
5	Шкаф меб	180	38	34	26	15	28	10	8	5	12	A	Y	14626
6	Кресло	130	26	33	20	33	25	15	13	5	5	A	Y	11544
7	Шведская	180	36	25	17	13	21	18	18	5	2	A	Y	10944
8	Стол(бол.)	150	37	27	22	16	18	3	17	8	12	A	Y	9379
9	Телевизор	360	96	10	34	34	29	13	6	2	18	A	Z	28248
10	Диван(бол	350	84	16	20	34	15	17	7	24	10	A	Z	22610
11	Телевизор	240	57	19	30	15	13	12	8	1	12	A	Z	14091
12	Диван(ма	220	49	27	11	23	12	16	9	18	10	A	Z	12483
13	Стол(мал.	100	25	23	35	18	19	2	6	6	8	B	Y	7125
14	Тумба	75	15	25	31	19	34	4	6	4	4	B	Y	6540
15	Шкаф нап	110	25	10	23	19	31	8	23	4	4	B	Z	7055
16	Шкаф нас	120	35	13	16	12	27	9	10	6	3	B	Z	5780
17	Люстра	75	17	18	15	30	14	5	17	4	5	B	Z	4466
18	Стул	50	10	22	30	24	32	1	24	4	4	C	Y	4320
19	Полка	50	15	20	13	13	17	14	22	2	4	C	Y	2205
20	Линолеум	30	7	18	30	17	21	20	7	2	2	C	Y	1978
21	Обои	20	4	10	26	35	16	19	6	1	1	C	Z	1392

Рисунок 18 – Созданный после работы программы файл с данными о товарах

Программа исправно подсчитывает категории ABC и XYZ. Затем она сортирует товары по категориям и записывает в таблицу.

Заключение

В данной выпускной квалификационной работе был изучен ABC-анализ и на его основе написано программное обеспечение реализующее на его основе распределение товаров по указанным зонам хранения.

В процессе исследования предметной области были изучены материалы, содержащие информацию о теоретических основах и практическом применении методов оптимизации управления запасами, включая ABC-анализ и XYZ-анализ. Были подробно рассмотрены их преимущества, ограничения и области применения, что позволило создать надежную теоретическую базу для последующей разработки алгоритмов.

При описании программной реализации были исследованы и применены современные инструменты и технологии. В частности, для разработки алгоритмов использовался язык программирования C++, а для работы с данными в формате Excel применялась библиотека xlnl. Программное обеспечение было создано в интегрированной среде разработки Visual Studio.

В процессе тестирования разработанного программного обеспечения были предоставлены примеры входных данных и результаты их обработки, которые демонстрировали правильность работы всех алгоритмов. Программа успешно выполняла задачи по классификации товаров и их распределению по складу с учетом значимости и стабильности спроса. В результате тестирования не было обнаружено ошибок, а полученные результаты соответствовали ожиданиям.

Исходя из всей предоставленной информации, можно заключить, что разработанное программное обеспечение эффективно справляется с задачей распределения товаров. Оно позволяет автоматизировать процесс классификации товаров и повысить эффективность использования складского пространства.

Список используемой литературы

1. ABC и прочие методы анализа [Электронный ресурс] // Olegon URL: <https://olegon.ru/showthread.php?t=22121> (дата обращения 03.05.2024).
2. Best practice in inventory management by Wild Antony [Электронный ресурс] // Internet archive URL: <https://archive.org/details/bestpracticeinin0000wild/page/n3/mode/2up> (дата обращения 14.04.2024).
3. Computed ABC Analysis for Rational Selection of Most Informative Variables in Multivariate Data [Электронный ресурс] // ResearchGate URL: https://www.researchgate.net/publication/278044519_Computed_ABC_Analysis_for_Rational_Selection_of_Most_Informative_Variables_in_Multivariate_Data (дата обращения 10.04.2024).
4. Manufacturing planning and control systems for supply chain management [Электронный ресурс] // Mexico Documents URL: <https://vdocuments.mx/manufacturing-planning-and-control-systems-for-supply-chain-management.html?page=1> (дата обращения 12.04.2024).
5. Purchasing and Supply Chain Management By Kenneth Lysons, Brian Farrington [Электронный ресурс] // Internet archive URL: <https://archive.org/details/purchasingsupply06edlyso> (дата обращения 08.04.2024).
6. Shipping and Logistics Management – Y.H.V. Lun, K.-H. Lai, T.C.E. Cheng 2010 [Электронный ресурс] // Морской Трекер URL: <https://seatracker.ru/viewtopic.php?t=24006> (дата обращения 08.04.2024).
7. Vcprg [Электронный ресурс] // GitHub URL: <https://github.com/microsoft/vcprg> (дата обращения 01.04.2024).
8. Xlnt [Электронный ресурс] // GitHub URL: <https://github.com/tfussell/xlnt> (дата обращения 25.03.2024).
9. Анурин В. Маркетинговые исследования потребительского рынка // В. Анурин, И. Муромкина, Е. Евтушенко. – СПб.: Питер, 2006.

10. Буйлин А. ABC-XYZ анализ ассортимента выпускаемой продукции как элемент стратегического маркетинга // Ремедиум. – №3. – 2005.
11. В. В. Ефимов, Н. В. Паймушкина метод авс (управление непрямыми производственными затратами) Учебно-методическое пособие [Электронный ресурс] // Ulstu URL: <http://lib.ulstu.ru/venec/v8/efimov.pdf> (дата обращения 28.04.2024).
12. Книга об управлении цепями поставок [Электронный ресурс] // SCM URL: <https://scm.center/book/export/html/4?page=1> (дата обращения 06.04.2024).
13. Константинов В. А. Управленческий учет и анализ с практическими примерами. Метод ABC. Учебник [Электронный ресурс] // Лабиринт URL: <https://www.labyrinth.ru/books/580171/> – (дата обращения 20.03.2024).
14. Лукинский В.С., Цвиринько И.А., Малевич Ю.В. Логистика. // СПб: Издательство СПбГИЭУ, 2000.
15. Множество и принцип Парето [Электронный ресурс] // SPBU URL: https://pureportal.spbu.ru/files/99337701/_2022.pdf (дата обращения 08.05.2024).
16. Новый метод многокритериальной оптимизации, основанный на локальной геометрии множества Парето [Электронный ресурс] // pseven URL: https://www.pseven.io/assets/files/publications/Perestoronin_thesis.pdf (дата обращения 10.05.2024).
17. О применении метода Парето-оптимальности при оценке эффективности функционирования организационных структур материально-технического обеспечения [Электронный ресурс] // Молодой ученый URL: <https://moluch.ru/archive/79/13951/> (дата обращения 14.05.2024).
18. Основы логистики. Функциональные области логистического управления [Электронный ресурс] // AUP URL: http://www.aup.ru/books/m192/3_4.htm (дата обращения 30.04.2024).
19. Построение компромиссных решений и определение эффективности Парето в многокритериальных системах [Электронный

ресурс] // HABR URL: <https://habr.com/ru/companies/otus/articles/750038/> (дата обращения 12.05.2024).

20. Применение методов ABC и XYZ анализа в системе бюджетных закупок лекарственных средств [Электронный ресурс] // SciUp URL: <https://sciup.org/140104743> (дата обращения 16.04.2024).

21. Скородумов А.А. КРАТКИЙ КУРС молодого менеджера по закупкам & ЛИКБЕЗ маркетолога-аналитика [Электронный ресурс] // MatrixCheck URL: https://matrixcheck.ru/wp-content/uploads/2015/01/Aleksey_Skorodumov_Kratky_Kurs_Analitika.pdf (дата обращения 03.04.2024).

22. Управление запасами широкой номенклатуры: с чего начать? [Электронный ресурс] // HSE URL: <https://www.hse.ru/data/391/032/1239/ABC-XYZ%20-%20для%20Логинфо.pdf> (дата обращения 18.04.2024).

23. Фишер А., «Методы выделения групп в ABC-анализе» // журнал «Логистика и Управление», № 1-2008.

24. Шведова И.А. ABC-анализ: полный гид [Электронный ресурс] // Литрес URL: <https://www.litres.ru/book/irina-aleksandrovna-shvedova/avs-analiz-polnyy-gid-66677318/chitat-onlayn/> – (дата обращения 15.03.2024).

25. Экономический риск и методы его измерения Ч2 [Электронный ресурс] // Studfile URL: <https://studfile.net/preview/1100371/page:12/> (дата обращения 06.05.2024).