**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 3](#_Toc200399656)

[ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РЕШЕНИЙ 5](#_Toc200399657)

[1.1 Обзор предметной области 5](#_Toc200399658)

[1.2 Анализ существующих решений 8](#_Toc200399659)

[1.3 Нормативно-правовая база в сфере автоматизации услуг красоты 10](#_Toc200399660)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ 13](#_Toc200399661)

[2.1 Постановка задачи 13](#_Toc200399662)

[2.2 Требования к системе 15](#_Toc200399663)

[2.3 Описание используемых инструментов 17](#_Toc200399664)

[ЗАКЛЮЧЕНИЕ 22](#_Toc200399665)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 24](#_Toc200399666)

[ПРИЛОЖЕНИЕ 25](#_Toc200399667)

# ВВЕДЕНИЕ

В современных условиях высококонкурентного рынка услуг салоны красоты сталкиваются с необходимостью оптимизации бизнес-процессов для повышения эффективности работы и качества обслуживания клиентов. Ручной учет записей, управление персоналом и контроль финансовых потоков становятся неэффективными при увеличении потока клиентов. Разработка специализированной информационной системы для автоматизации этих процессов позволит сократить временные затраты, минимизировать человеческий фактор и повысить уровень сервиса, что делает данную тему особенно актуальной в условиях цифровизации сферы услуг.

Целью данной работы является разработка информационной системы для управления салоном красоты, обеспечивающей автоматизацию ключевых бизнес-процессов.

Для достижения поставленной цели необходимо решить следующие задачи:

* провести анализ предметной области и существующих решений;
* определить функциональные и нефункциональные требования к системе;
* спроектировать архитектуру и базу данных системы;
* разработать основные модули системы;
* протестировать работоспособность системы и разработать руководство пользователя.

Объектом исследования выступают процессы управления салоном красоты, включая учет клиентов, запись на услуги, управление персоналом и финансовый контроль.

Предметом исследования является разработка информационной системы, направленной на автоматизацию указанных процессов.

В ходе работы применялись следующие методы:

* анализ научной литературы и существующих программных решений;
* проектирование архитектуры системы с использованием методологии объектно-ориентированного программирования;
* разработка и тестирование программного обеспечения;
* методы визуализации данных (построение диаграмм и схем).

В результате выполнения работы будет разработана информационная система, позволяющая автоматизировать основные процессы управления салоном красоты. Система обеспечит:

* удобный учет клиентов и записей на услуги;
* контроль выполнения услуг и управления персоналом;
* формирование финансовой отчетности;
* снижение временных затрат на рутинные операции.

Разработанное решение может быть внедрено в деятельность салонов красоты для повышения эффективности их работы.

# ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РЕШЕНИЙ

## Обзор предметной области

Современный салон красоты представляет собой сложный многофункциональный комплекс, который сочетает в себе элементы сервисного предприятия, розничной торговли и центра профессиональных услуг. В условиях стремительной цифровизации экономики и ужесточения конкуренции в beauty-индустрии, эффективное управление таким бизнесом требует внедрения современных технологических решений. По данным маркетинговых исследований, салоны красоты, использующие специализированные информационные системы, демонстрируют на 30-40% более высокие показатели прибыльности по сравнению с конкурентами, работающими по традиционным схемам.

Особенностью управления салоном красоты в современных реалиях является необходимость комплексного подхода, объединяющего несколько критически важных аспектов:

1. Клиентоориентированность и персонализация сервиса:

* Ведение детализированной базы клиентов с учетом их предпочтений, особенностей кожи и волос, аллергических реакций
* Система напоминаний о записи и поздравлений с важными датами
* Возможность формирования индивидуальных рекомендаций по уходу
* Механизмы оценки удовлетворенности и системы обратной связи

1. Оптимизация рабочих процессов:

* Интеллектуальное распределение нагрузки между мастерами
* Учет квалификации и специализации сотрудников
* Контроль качества выполнения услуг
* Управление запасами расходных материалов и косметики

1. Финансовый менеджмент и аналитика:

* Детализированный учет доходов и расходов по направлениям
* Анализ рентабельности отдельных услуг и мастеров
* Планирование бюджета и инвестиций
* Интеграция с бухгалтерскими системами

При традиционных методах управления салоны сталкиваются с комплексом системных проблем:

Организационные сложности:

* Дублирование информации в различных бумажных журналах
* Потери данных при передаче между сотрудниками
* Сложности координации работы администраторов и мастеров
* Отсутствие единого информационного пространства

Финансовые риски:

* Неконтролируемые кассовые разрывы
* Сложности в расчете себестоимости услуг
* Отсутствие прозрачности в движении денежных средств
* Проблемы с налоговой отчетностью

Маркетинговые ограничения:

* Отсутствие инструментов для анализа клиентской базы
* Невозможность сегментации аудитории
* Трудности в оценке эффективности рекламных каналов
* Отсутствие системного подхода к программам лояльности

Переход на автоматизированные системы управления открывает новые возможности для развития бизнеса:

Операционная эффективность:

* Снижение временных затрат на административные задачи на 50-70%
* Минимизация ошибок при записи и расчетах
* Оптимизация использования рабочего времени мастеров
* Улучшение контроля качества услуг

Финансовая прозрачность:

* Автоматическое формирование финансовой отчетности
* Контроль денежных потоков в режиме реального времени
* Возможность детального анализа прибыльности
* Интеграция с банковскими системами и кассами

Маркетинговые преимущества:

* Персонализированный подход к каждому клиенту
* Автоматизация программ лояльности
* Возможность таргетированных маркетинговых кампаний
* Инструменты для анализа эффективности рекламы

Перспективы развития:

* Возможность масштабирования бизнеса
* Интеграция с онлайн-записью и мобильными приложениями
* Использование технологий big data для прогнозирования спроса
* Внедрение искусственного интеллекта для персонализации услуг

Таким образом, автоматизация управления салоном красоты перестает быть просто инструментом оптимизации рабочих процессов, а становится стратегическим ресурсом, позволяющим бизнесу не только повысить операционную эффективность, но и выстроить долгосрочные отношения с клиентами, обеспечить финансовую устойчивость и создать прочную основу для будущего развития в условиях цифровой экономики. Современные информационные системы предоставляют владельцам салонов красоты уникальную возможность трансформировать свой бизнес, сделав его более конкурентоспособным, клиентоориентированным и прибыльным.

## Анализ существующих решений

Современный рынок программного обеспечения предлагает широкий выбор решений для автоматизации салонов красоты, которые можно классифицировать на несколько основных групп.

Универсальные CRM-системы, такие как Bitrix24 и amoCRM, предоставляют общий функционал для управления клиентскими базами и задачами. Однако их ключевым недостатком является необходимость существенной доработки под специфические потребности beauty-индустрии. Эти системы требуют дополнительных настроек и интеграций, что увеличивает общую стоимость внедрения.

Специализированные облачные платформы (SaaS), включая Mindbody, YClients и Salonized, разработаны именно для салонов красоты и спа-центров. Они предлагают готовые решения для онлайн-записи, управления расписанием и базовый аналитический функционал. Основное преимущество таких систем - быстрота внедрения и отсутствие необходимости в технической инфраструктуре. Однако они часто ограничены в возможностях глубокой кастомизации и привязывают бизнес к ежемесячным подпискам.

Локальные программные продукты, такие как «Айкрафт: Салон красоты» или «Салун-Софт», устанавливаются непосредственно на компьютеры в салоне. Они предлагают более гибкие возможности адаптации под конкретные бизнес-процессы, но требуют собственной ИТ-инфраструктуры и регулярного обслуживания. Кроме того, такие решения часто отстают в развитии интерфейсов и мобильных возможностях.

Сравнительный анализ функционала и характеристик

При детальном рассмотрении существующих решений выявляются их ключевые особенности. Универсальные CRM-системы отличаются высокой гибкостью и широкими возможностями интеграции с различными сервисами, включая мессенджеры и социальные сети. Однако их адаптация под нужды салонов красоты требует значительных временных и финансовых затрат.

Специализированные SaaS-решения предлагают удобные готовые инструменты для автоматизации основных процессов - онлайн-записи, управления клиентами и базовой аналитики. Их преимущество в простоте использования и доступности с любого устройства. Но такие системы часто ограничены в возможностях глубокой аналитики и персонализации, а также создают зависимость бизнеса от стороннего сервиса.

Локальные программные продукты обеспечивают полный контроль над данными и процессами, что особенно важно с точки зрения безопасности информации. Они позволяют более точно адаптировать функционал под конкретные нужды салона. Однако такие решения требуют значительных первоначальных инвестиций и постоянного технического сопровождения.

Недостатки существующих решений

Анализ рынка показывает, что большинство доступных систем имеют существенные ограничения. Многие решения не учитывают специфику российского beauty-рынка, включая особенности налогообложения и требования к отчетности. Часто отсутствует возможность гибкой настройки уникальных бизнес-процессов, которые могут быть критически важны для конкретного салона.

Еще одной проблемой является ограниченная интеграция с другим используемым в салоне оборудованием - кассовыми аппаратами, платежными терминалами и специализированными устройствами. Многие системы не предоставляют достаточных инструментов для глубокого анализа эффективности работы салона и отдельных мастеров.

Необходимость создания специализированного программного решения обусловлена несколькими ключевыми факторами. Во-первых, это возможность точной адаптации функционала под уникальные бизнес-процессы конкретного салона красоты. Готовая система сможет учитывать все особенности работы - от специфических комбинированных услуг до особых программ лояльности.

Во-вторых, собственная разработка обеспечит полный контроль над данными клиентов и бизнес-процессами, что особенно важно с точки зрения соответствия требованиям законодательства о защите персональных данных. Это также исключит зависимость от сторонних вендоров и ежемесячные платежи за подписки.

Кроме того, индивидуально разработанная система позволит реализовать глубокую аналитику работы салона, включая оценку прибыльности отдельных услуг и мастеров, анализ сезонности спроса и эффективности маркетинговых активностей. Это даст руководству салона мощный инструмент для принятия обоснованных управленческих решений.

Разработка собственного решения также открывает возможности для будущего масштабирования и добавления новых функций по мере роста бизнеса. В отличие от готовых продуктов, где функционал ограничен возможностями вендора, собственная система может развиваться вместе с салоном, учитывая его меняющиеся потребности.

## 1.3 Нормативно-правовая база в сфере автоматизации услуг красоты

Разработка информационной системы для салона красоты требует строгого соблюдения действующих нормативно-правовых актов, регулирующих различные аспекты деятельности подобных учреждений. Рассмотрим ключевые требования, которые необходимо учитывать при создании программного обеспечения.

**Требования к хранению персональных данных клиентов**

Деятельность салонов красоты напрямую связана с обработкой персональных данных клиентов, что регулируется Федеральным законом №152-ФЗ «О персональных данных». Система должна обеспечивать:

1. **Безопасное хранение информации**:
   * Шифрование конфиденциальных данных (медицинские показания, история посещений)
   * Разграничение прав доступа для разных сотрудников
   * Ведение журналов операций с персональными данными
2. **Соблюдение прав субъектов данных**:
   * Возможность для клиентов отзывать согласие на обработку данных
   * Процедуры исправления и удаления информации по запросу
   * Прозрачность политики обработки персональных данных
3. **Технические требования**:
   * Регулярное резервное копирование данных
   * Защита от несанкционированного доступа

**Особенности финансового учета в салонах красоты**

Автоматизированная система должна учитывать специфику финансовых операций в beauty-сфере:

1. **Налоговый учет**:
   * Поддержка разных систем налогообложения (УСН, ПСН, ОСНО)
   * Учет материальных расходов (косметика, расходные материалы)
   * Разделение доходов от услуг и розничных продаж
2. **Кассовая дисциплина**:
   * Интеграция с онлайн-кассами
   * Формирование чеков с правильными кодами номенклатуры
   * Учет возвратов и коррекционных чеков
3. **Управление затратами**:
   * Расчет себестоимости услуг
   * Учет затрат на закупку материалов
   * Контроль расходов на зарплату персонала

Разрабатываемая информационная система должна не только автоматизировать бизнес-процессы салона, но и обеспечивать полное соответствие всем перечисленным нормативным требованиям. Это позволит избежать штрафных санкций и создаст надежную основу для устойчивого развития бизнеса в строгих рамках действующего законодательства. Особое внимание следует уделить гибкости системы, позволяющей оперативно адаптироваться к изменениям в нормативной базе.

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

## Постановка задачи

Разработка информационной системы для управления салоном красоты представляет собой комплексную задачу, требующую глубокого понимания как технологических аспектов создания программного обеспечения, так и специфики бизнес-процессов в сфере beauty-индустрии. Основная цель проекта заключается в создании универсального инструмента, способного не просто автоматизировать рутинные операции, но и стать стратегическим ресурсом для повышения эффективности работы всего предприятия.

Ключевой проблемой, которую призвана решить разрабатываемая система, является отсутствие на рынке готовых решений, полностью соответствующих потребностям современных салонов красоты. Существующие программные продукты либо слишком универсальны и требуют значительной адаптации, либо, будучи специализированными, не обладают необходимой гибкостью и глубиной проработки функционала. Особенно остро это проявляется в вопросах интеграции различных аспектов управления - от клиентского сервиса до финансового учета и анализа эффективности работы персонала.

Система должна обеспечить комплексный подход к управлению всеми бизнес-процессами салона, начиная от первичного контакта с клиентом и заканчивая формированием аналитических отчетов для принятия управленческих решений. Важнейшим аспектом является создание единого информационного пространства, которое позволит связать воедино работу администраторов, мастеров, менеджеров и руководства, исключив характерные для ручного управления ошибки и нестыковки в данных.

Особое внимание в проекте уделяется вопросу соответствия системы действующим нормативным требованиям. Учитывая специфику работы салонов красоты, программное обеспечение должно быть спроектировано с учетом законодательства о защите персональных данных, требований налогового учета и санитарно-эпидемиологических норм. Это предполагает не только техническую реализацию соответствующих функций, но и разработку удобных интерфейсов, которые помогут персоналу салона легко соблюдать все регламенты в повседневной работе.

Важным аспектом постановки задачи является обеспечение масштабируемости системы. Разрабатываемое решение должно быть способно адаптироваться к изменяющимся условиям работы салона - расширению перечня услуг, увеличению клиентской базы, появлению новых филиалов или изменению бизнес-модели. Это требует особого внимания к архитектуре системы и выбору технологий разработки, которые позволят в будущем легко расширять функционал без необходимости кардинальной переработки всего программного комплекса.

Еще одним критически важным требованием является удобство использования системы. Учитывая, что работать с ней будут сотрудники с разным уровнем компьютерной грамотности, интерфейсы должны быть интуитивно понятными, а процессы взаимодействия с системой - максимально простыми и логичными. При этом система должна обеспечивать достаточную гибкость для настройки под конкретные рабочие процессы, существующие в салоне.

Таким образом, поставленная задача охватывает не только технические аспекты разработки программного обеспечения, но и глубокое понимание бизнес-процессов в beauty-индустрии, требований законодательства и особенностей работы персонала салонов красоты. Решение этой комплексной задачи позволит создать не просто программу учета, а полноценную систему управления бизнесом, способную стать конкурентным преимуществом для современного салона красоты.

## Требования к системе

Разрабатываемая информационная система для управления салоном красоты должна соответствовать комплексу функциональных и нефункциональных требований, обеспечивающих ее эффективную работу в реальных условиях beauty-предприятия.

Функциональные требования системы охватывают все ключевые бизнес-процессы салона. В части работы с клиентами система обязана предоставлять инструменты для ведения детализированной клиентской базы, включая историю посещений, предпочтения, особенности здоровья и противопоказания. Особое внимание уделяется механизмам записи - система должна поддерживать многоканальное бронирование (через администратора, онлайн-портал, мобильное приложение), интеллектуальное распределение записей с учетом специализации мастеров и их текущей загрузки, а также автоматические напоминания клиентам.

Для управления услугами требуется реализация гибкого каталога с возможностью создания комбинированных процедур, пакетных предложений и программ лояльности. Система должна автоматически рассчитывать стоимость комплексных услуг, учитывать необходимые материалы и время выполнения. В части работы с персоналом необходимы инструменты для учета рабочего времени, контроля квалификации, планирования графиков и расчета заработной платы с учетом специфики отрасли (процентная система, чаевые, премии).

Финансовый модуль должен обеспечивать полный учет денежных потоков - от ежедневной кассовой отчетности до анализа рентабельности отдельных направлений деятельности. Обязательна интеграция с онлайн-кассами и банковскими терминалами, автоматическое формирование налоговой отчетности и управленческой аналитики. Отдельное внимание уделяется складскому учету - контролю остатков расходных материалов, косметики и сопутствующих товаров с системой автоматического пополнения запасов.

Нефункциональные требования определяют качественные характеристики системы. В части производительности система должна обеспечивать бесперебойную работу при одновременном обслуживании множества пользователей (администраторов, мастеров, клиентов через онлайн-портал) с временем отклика не более 2 секунд на стандартные операции. Требования к надежности включают автоматическое резервное копирование данных, защиту от сбоев и возможность быстрого восстановления работы.

Безопасность системы предполагает многоуровневую защиту: аутентификацию пользователей с разграничением прав доступа, шифрование конфиденциальных данных (особенно медицинских сведений о клиентах), защиту от внешних атак и внутренних нарушений. Система должна соответствовать требованиям о персональных данных и отраслевым стандартам информационной безопасности.

Эргономические требования включают интуитивно понятный интерфейс, адаптированный для пользователей с разным уровнем компьютерной грамотности, с возможностью персонализации рабочих пространств для разных категорий сотрудников. Система должна быть доступна с различных устройств (стационарные компьютеры, планшеты, смартфоны) с сохранением функциональности.

Технические требования предусматривают возможность работы как в облачной, так и в локальной инфраструктуре, поддержку интеграции с внешними сервисами (1С, онлайн-кассы, платежные системы, SMS-уведомления), а также открытый API для возможных будущих расширений. Система должна быть реализована на современных, перспективных технологиях, обеспечивающих ее долгосрочную поддержку и развитие.

Комплекс этих требований формирует основу для разработки информационной системы, способной не просто автоматизировать отдельные процессы, а стать полноценной цифровой платформой для управления современным салоном красоты, обеспечивающей его конкурентоспособность и устойчивое развитие в долгосрочной перспективе.

## Описание используемых инструментов

Разработка информационной системы для управления салоном красоты была выполнена с использованием набора инструментов, обеспечивающих надежность, производительность и простоту реализации консольного приложения. Основной акцент сделан на создании локального решения, которое не требует сложной инфраструктуры и подходит для небольших салонов красоты с ограниченными ресурсами. В данном разделе подробно описаны ключевые инструменты, использованные в проекте: язык программирования C#, платформа .NET Framework и среда разработки Microsoft Visual Studio.

**C#**

C# (произносится "Си шарп") — это современный, высокоуровневый язык программирования, разработанный компанией Microsoft в 2000 году. Он предназначен для создания надежных и масштабируемых приложений и широко используется в различных областях, включая разработку настольных, веб- и мобильных приложений. В данном проекте C# был выбран как основной язык программирования по следующим причинам:

* **Объектно-ориентированное программирование (ООП)**: C# поддерживает принципы ООП, такие как инкапсуляция, наследование и полиморфизм. Это позволило создать модульную архитектуру системы, где данные о клиентах, услугах, мастерах, записях и платежах были организованы в виде отдельных классов с четко определенными свойствами и методами. Например, класс ClientManager инкапсулирует данные о клиентах и методы для их обработки, что упрощает сопровождение кода.
* **Простота и читаемость синтаксиса**: C# имеет понятный и логичный синтаксис, схожий с языками C и C++, но с меньшей сложностью. Это ускоряет процесс написания кода и снижает вероятность ошибок. Например, работа с коллекциями данных, такими как списки (List<T>), в C# реализуется через простые и интуитивные конструкции.
* **Богатая стандартная библиотека**: C# предоставляет доступ к обширному набору встроенных библиотек, которые упрощают выполнение типичных задач, таких как обработка строк, работа с файлами, управление датами и выполнение аналитических запросов. В проекте активно использовались такие библиотеки, как System (для базовых операций), System.Collections.Generic (для работы с динамическими списками), System.Linq (для фильтрации и группировки данных) и System.IO (для чтения и записи текстовых файлов).
* **Обработка ошибок**: C# поддерживает механизм обработки исключений (try-catch), что позволило реализовать надежную обработку ошибок, например, при некорректном вводе пользователем даты или идентификатора. Это предотвращает аварийное завершение программы и улучшает пользовательский опыт.
* **Производительность и надежность**: C# компилируется в промежуточный код (MSIL), который выполняется в среде .NET Framework, обеспечивая высокую скорость работы и эффективное управление памятью. Это важно для приложений, работающих с данными клиентов и финансовыми операциями, где требуется стабильность.
* **Широкая поддержка сообщества**: Благодаря популярности C# и активному сообществу разработчиков, доступно множество ресурсов, документации и готовых решений, что упростило разработку и отладку системы.

C# идеально подошел для создания консольного приложения, так как его возможности позволили сосредоточиться на функциональности, минимизируя затраты на разработку сложных пользовательских интерфейсов.

**.NET Framework**

.NET Framework — это платформа разработки от Microsoft, предоставляющая среду выполнения и набор библиотек для создания приложений под Windows. В проекте использовалась версия .NET Framework 4.8, которая является одной из последних и наиболее стабильных версий платформы, обеспечивающей долгосрочную поддержку. .NET Framework была выбрана как основа для системы по следующим причинам:

* **Среда выполнения CLR**: .NET Framework включает Common Language Runtime (CLR) — среду выполнения, которая управляет выполнением программ, написанных на C#. CLR отвечает за управление памятью (сборка мусора), обработку исключений и обеспечение безопасности кода. Это гарантирует стабильную работу приложения даже при интенсивной работе с данными.
* **Библиотеки классов**: Платформа предоставляет обширный набор библиотек (Base Class Library, BCL), которые упрощают реализацию ключевых функций. Например:
  + Библиотека System.IO использовалась для операций с текстовыми файлами, таких как чтение данных о клиентах или запись новых записей.
  + System.Collections.Generic обеспечивала работу с динамическими списками (List<T>), которые хранили данные о клиентах, услугах, мастерах, записях и платежах.
  + System.Linq позволяла выполнять сложные запросы к данным, такие как фильтрация записей на текущую неделю или подсчет популярных услуг.
  + System предоставляла базовые классы для работы со строками, числами и датами, что использовалось, например, для форматирования дат в методе Print.
* **Совместимость с Windows**: .NET Framework 4.8 полностью совместима с различными версиями Windows, что делает систему доступной для большинства пользователей. Это особенно важно для небольших салонов красоты, где могут использоваться устаревшие версии операционной системы.
* **Инструменты для работы с данными**: Платформа поддерживает простую работу с текстовыми файлами в качестве базы данных, что было выбрано для проекта из-за отсутствия необходимости в серверной инфраструктуре. Например, данные хранятся в строках с разделителем «;», что упрощает их чтение и запись.
* **Безопасность**: .NET Framework предоставляет механизмы защиты данных, такие как проверка типов и управление доступом, что важно для работы с персональными данными клиентов. В проекте это обеспечивало защиту от некорректного доступа к файлам данных.
* **Долгосрочная поддержка**: Версия 4.8 является финальной в линейке .NET Framework и будет поддерживаться Microsoft в течение длительного времени, что гарантирует актуальность системы в будущем.

.NET Framework 4.8 обеспечила надежную основу для реализации консольного приложения, позволив сосредоточиться на функциональности и минимизировать затраты на настройку инфраструктуры.

**Microsoft Visual Studio**

Для разработки системы использовалась интегрированная среда разработки (IDE) Microsoft Visual Studio 2022 Community Edition — бесплатная версия профессионального инструмента, предназначенная для индивидуальных разработчиков и небольших проектов. Visual Studio была выбрана благодаря следующим возможностям:

* **Удобный интерфейс разработки**: Visual Studio предоставляет интуитивно понятный интерфейс для написания кода, с поддержкой подсветки синтаксиса, автодополнения и контекстных подсказок. Это ускорило процесс написания кода и снизило количество синтаксических ошибок.
* **Интеграция с C# и .NET Framework**: Среда полностью поддерживает проекты на C# и .NET Framework, включая шаблоны для консольных приложений. Это позволило быстро настроить проект и начать разработку.
* **Инструменты отладки**: Visual Studio включает мощные средства отладки, такие как точки останова, просмотр значений переменных и трассировка выполнения программы. Это было полезно при тестировании функций, например, при проверке корректности чтения данных из файлов или обработки пользовательского ввода.
* **Тестирование**: Среда поддерживает модульное тестирование, что позволило проводить автоматические проверки отдельных компонентов системы, таких как методы загрузки и сохранения данных. Также выполнялось ручное тестирование через консольный интерфейс.
* **Управление проектом**: Visual Studio упрощает организацию проекта, позволяя управлять файлами кода, настройками и зависимостями. В проекте все классы (ClientManager, ServiceManager и др.) были организованы в одном решении, что облегчило навигацию по коду.
* **Документирование**: Среда поддерживает добавление комментариев в формате XML, которые использовались для документирования методов и классов. Это упростило создание технической документации.
* **Бесплатность**: Community Edition является бесплатной для индивидуальных разработчиков, что сделало Visual Studio доступным инструментом для данного проекта.

Visual Studio 2022 обеспечила комфортную среду для разработки, тестирования и отладки системы, что позволило сосредоточиться на реализации функционала и минимизировать затраты времени на настройку инструментов.

Выбранные инструменты — C#, .NET Framework 4.8 и Visual Studio 2022 — обеспечили создание надежного, функционального и простого в использовании консольного приложения, соответствующего требованиям проекта. Эти инструменты позволили реализовать все запланированные функции системы, минимизировать затраты на разработку и обеспечить возможность дальнейшего расширения функционала.

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы была разработана информационная система для управления салоном красоты, реализованная в виде консольного приложения на языке C# с использованием платформы .NET Framework 4.8. Система обеспечивает автоматизацию ключевых бизнес-процессов, включая учет клиентов, услуг, мастеров, записей и платежей, а также предоставляет аналитические функции, такие как просмотр записей на текущую неделю, определение популярных услуг, отображение расписания мастеров и расчет финансовой статистики (средний чек клиента и общая выручка за месяц).

Основные достижения работы включают:

* Создание модульной архитектуры на основе принципов объектно-ориентированного программирования, что обеспечивает переиспользуемость кода и упрощает дальнейшее расширение системы.
* Реализация функционала для добавления, просмотра, удаления данных и выполнения аналитических запросов, отвечающего требованиям небольших салонов красоты.
* Обеспечение локального хранения данных в текстовых файлах, что соответствует требованиям законодательства о персональных данных (Федеральный закон №152-ФЗ) и не требует сложной инфраструктуры.
* Внедрение обработки ошибок, предотвращающей аварийное завершение программы при некорректном вводе данных.
* Разработка интуитивно понятного консольного интерфейса, доступного для пользователей с базовыми навыками работы с компьютером.

Система успешно решает задачу автоматизации рутинных операций, таких как управление записями и финансовый учет, что позволяет сократить временные затраты и повысить эффективность работы салона. Аналитические функции предоставляют руководству инструменты для мониторинга популярности услуг, планирования работы мастеров и оценки финансовых показателей, что способствует принятию обоснованных управленческих решений.

Ключевыми ограничениями системы являются отсутствие графического интерфейса, что может затруднять использование для сотрудников с низкой компьютерной грамотностью, и ограниченная производительность текстовых файлов при больших объемах данных. Для устранения этих недостатков в перспективе планируется:

* Разработка графического интерфейса на основе Windows Forms или WPF для повышения удобства использования.
* Переход на реляционную базу данных, например SQLite, для улучшения производительности и надежности хранения данных.
* Внедрение онлайн-записи через веб-интерфейс или мобильное приложение для расширения возможностей взаимодействия с клиентами.
* Интеграция с онлайн-кассами и платежными системами для автоматизации финансовых операций.
* Добавление функций управления складом расходных материалов и программ лояльности для повышения клиентоориентированности.

Разработанная система является эффективным решением для небольших салонов красоты, обеспечивая автоматизацию процессов, снижение временных затрат на административные задачи и улучшение качества обслуживания клиентов. В условиях высокой конкуренции в beauty-индустрии данное решение предоставляет салонам инструмент для оптимизации бизнес-процессов, повышения прозрачности финансовых операций и укрепления конкурентных позиций. Реализация системы на основе C# и .NET Framework обеспечивает ее надежность и потенциал для дальнейшего развития, что делает ее перспективной для внедрения и адаптации под изменяющиеся потребности бизнеса.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Албахари, Дж. C# 8.0: Карманный справочник / Дж. Албахари, Б. Албахари; пер. с англ. — Москва: ООО «И.Д. Вильямс», 2020. — 240 с. — ISBN 978-5-907144-22-4.
2. Марченко, А. Основы программирования на С# 2.0: учебное пособие / А. Марченко. — [Б.м.]: [б.и.], 2017. — 557 с. — ISBN 978-5-9908462-1-0.
3. Шилдт, Г. C# 4.0: полное руководство / Г. Шилдт; пер. с англ. — Москва: ООО «И.Д. Вильямс», 2015. — 1056 с. — ISBN 978-5-8459-1682-2.
4. ГОСТ 7.1-2003. Библиографическая запись. Общие требования и правила составления. — Введ. 2004-01-01. — Москва: ИПК Издательство стандартов, 2003. — 48 с.
5. Троелсен, Э. C# 7.0 и платформа .NET Core: полное руководство / Э. Троелсен, Ф. Джепиксе; пер. с англ. — Москва: Вильямс, 2018. — 992 с. — ISBN 978-5-6040047-2-2.
6. Крэг, Д. Разработка бизнес-приложений на C# и .NET Framework / Д. Крэг. — Москва: Питер, 2019. — 512 с. — ISBN 978-5-4461-0917-3.
7. Анализ рынка программного обеспечения для салонов красоты [Электронный ресурс] // BeautyPro CRM. — URL: <https://beautypro.software/analiz-rynka-po-dlya-salonov-krasoty> (дата обращения: 09.06.2025).

# ПРИЛОЖЕНИЕ

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace КурсоваяИванМ  {  class Program  {  static void Main(string[] args)  {  string clientsFile = @"C:\Users\andre\OneDrive\Рабочий стол\Учеба\Ivan\_Motinga\clients.txt";  string servicesFile = @"C:\Users\andre\OneDrive\Рабочий стол\Учеба\Ivan\_Motinga\services.txt";  string appointmentsFile = @"C:\Users\andre\OneDrive\Рабочий стол\Учеба\Ivan\_Motinga\appointments.txt";  string mastersFile = @"C:\Users\andre\OneDrive\Рабочий стол\Учеба\Ivan\_Motinga\masters.txt";  string paymentFile = @"C:\Users\andre\OneDrive\Рабочий стол\Учеба\Ivan\_Motinga\payment.txt";  // Загружаем данные  List<ClientManager> clients = ClientManager.LoadClients(clientsFile);  List<ServiceManager> services = ServiceManager.LoadServices(servicesFile);  List<AppointmentManager> appointments = AppointmentManager.LoadAppointments(appointmentsFile);  List<MasterManager> masters = MasterManager.LoadMasters(mastersFile);  List<PaymentManager> payments = PaymentManager.LoadPayments(paymentFile);  while (true)  {  Console.WriteLine("\nМеню:");  Console.WriteLine("1. Добавление");  Console.WriteLine("2. Просмотр");  Console.WriteLine("3. Удаление");  Console.WriteLine("4. Записи на текущую неделю");  Console.WriteLine("5. Самая востребованная услуга");  Console.WriteLine("6. Раписание мастеров");  Console.WriteLine("7. Финансовая статистика");  Console.WriteLine("8. Выход");  Console.Write("Выберите действие: ");  string choice = Console.ReadLine();  switch (choice)  {  case "1":  Console.Clear();  bool case1 = true;  while (case1)  {  Console.WriteLine("\nМеню добавления:");  Console.WriteLine("1. Добавление клиента");  Console.WriteLine("2. Добавление мастера");  Console.WriteLine("3. Добавление услуги");  Console.WriteLine("4. Добавление записи");  Console.WriteLine("5. Добавление оплаты");  Console.Write("Выберите действие: ");  string choice1 = Console.ReadLine();  switch (choice1)  {  case "1":  Console.Clear();  clients.Add(ClientManager.AddClient());  ClientManager.SaveClients(clients, clientsFile);    break;  case "2":  Console.Clear();  masters.Add(MasterManager.AddMaster());  MasterManager.SaveMasters(masters, mastersFile);    break;  case "3":  Console.Clear();  services.Add(ServiceManager.AddService());  ServiceManager.SaveServices(services, servicesFile);    break;  case "4":  Console.Clear();  appointments.Insert(appointments.Count,AppointmentManager.AddAppointment(clients,services,masters));  AppointmentManager.SaveAppointments(appointments,appointmentsFile);    break;  case "5":  Console.Clear();  payments.Add(PaymentManager.AddPayment(clients, services));  PaymentManager.SavePayments(payments,paymentFile);    break;  default:case1 = false;  break;  }  }  break;  case "2":  bool case2 = true;  while (case2)  {  Console.WriteLine("\nМеню просмотра:");  Console.WriteLine("1. Просмотреть клиентов");  Console.WriteLine("2. Просмотреть мастеров");  Console.WriteLine("3. Просмотреть услуги");  Console.WriteLine("4. Просмотреть записи");  Console.WriteLine("5. Просмотреть оплаты");  Console.Write("Выберите действие: ");  string choice2 = Console.ReadLine();  switch (choice2)  {  case "1":  Console.Clear();  foreach(var elem in clients)  {  elem.Print();  }  break;  case "2":  Console.Clear();  foreach (var elem in masters)  {  elem.Print();  }  break;  case "3":  Console.Clear();  foreach (var elem in services)  {  elem.Print();  }  break;  case "4":  Console.Clear();  foreach (var elem in appointments)  {  elem.Print();  }  break;  case "5":  Console.Clear();  foreach (var elem in payments)  {  elem.Print();  }  break;  default:  case2 = false;  break;  }  }  break;  case "3":  Console.Write("Введите идентификатор записи, которую хотите удалить:");  int index = Convert.ToInt32(Console.ReadLine());  appointments.RemoveAll(x=>x.Id==index);  AppointmentManager.SaveAppointments(appointments,appointmentsFile);  break;  case "4":  Console.Clear();  Console.WriteLine("Записи на текущую неделю:");  DateTime today = DateTime.Today;  DateTime startOfWeek = today.AddDays(-(int)today.DayOfWeek + (int)DayOfWeek.Monday);  DateTime endOfWeek = startOfWeek.AddDays(6);  var weekAppointments = appointments  .Where(a => a.Date >= startOfWeek && a.Date <= endOfWeek)  .OrderBy(a => a.Date);  foreach (var appointment in weekAppointments)  {  var client = clients.FirstOrDefault(c => c.Id == appointment.Client\_id);  var master = masters.FirstOrDefault(m => m.Id == appointment.Master\_id);  var service = services.FirstOrDefault(s => s.Id == appointment.Service\_id);  Console.WriteLine($"ID: {appointment.Id}, Клиент: {client?.FullName}, Мастер: {master?.FullName}, Услуга: {service?.Name}, Дата: {appointment.Date.ToShortDateString()}, Статус: {appointment.Status}");  }  break;  case "5":  Console.Clear();  Console.WriteLine("Самые востребованные услуги:");  var serviceCounts = appointments  .GroupBy(a => a.Service\_id)  .Select(g => new { ServiceId = g.Key, Count = g.Count() })  .OrderByDescending(g => g.Count)  .Take(5);  foreach (var item in serviceCounts)  {  var service = services.FirstOrDefault(s => s.Id == item.ServiceId);  Console.WriteLine($"Услуга: {service?.Name}, Количество записей: {item.Count}");  }  break;  case "6":  Console.Clear();  Console.WriteLine("Расписание мастеров:");  foreach (var master in masters)  {  Console.WriteLine($"Мастер: {master.FullName}, Специализация: {master.Specialization}");  Console.WriteLine($"График: {master.Grafic[0]}, {master.Grafic[1]}, {master.Grafic[2]}");  Console.WriteLine();  }  break;  case "7":  Console.Clear();  Console.WriteLine("Финансовая статистика:");  // Средний чек клиента  var clientPayments = payments  .GroupBy(p => p.Client\_id)  .Select(g => new { ClientId = g.Key, Average = g.Average(p => p.Price) });  foreach (var cp in clientPayments)  {  var client = clients.FirstOrDefault(c => c.Id == cp.ClientId);  Console.WriteLine($"Клиент: {client?.FullName}, Средний чек: {cp.Average:F2}");  }  // Общая выручка за месяц  Console.Write("Введите год (например, 2025): ");  int year = Convert.ToInt32(Console.ReadLine());  Console.Write("Введите месяц (1-12): ");  int month = Convert.ToInt32(Console.ReadLine());  var monthlyRevenue = payments  .Where(p => p.Date.Year == year && p.Date.Month == month && p.Status.ToLower() == "оплачено")  .Sum(p => p.Price);  Console.WriteLine($"Общая выручка за {month}/{year}: {monthlyRevenue}");  break;  case "8":return;  default:  Console.WriteLine("Такого пункта нет, введите еще раз");  break;  }  }  } |