

Специальность 09.02.07 Информационные системы и программирование

(квалификация: специалист по информационным ресурсам)

КУРСОВАЯ РАБОТА

по МДК 3.2. «Управление проектами»

Студента: Коготков Кристиан Юрьевич

(фамилия, имя, отчество)

Группы: 3-ИСП9-23

на тему: «Разработка программы для учёта операций в кассе столовой»

Руководитель: преподаватель Голиков Дмитрий Алексеевич

(должность ФИО)

Оценка \_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(Подпись преподавателя)

Дата «\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_2023 г.

Москва, 2023

Оглавление

[Введение 5](#_Toc152266481)

[Глава 1. Теоретические основы разработки программы учёта операций в кассе столовой 7](#_Toc152266482)

[1.1 Описание предметной области 7](#_Toc152266483)

[1.2 Архитектура программы 9](#_Toc152266484)

[1.3 Функциональные требования 12](#_Toc152266485)

[1.4 Проектирование пользовательского интерфейса 13](#_Toc152266486)

[1.5 Описание алгоритмов работы программы 15](#_Toc152266487)

[Выводы по теоретической части 16](#_Toc152266488)

[Глава 2 Моделирование и разработка программного решения для расчета заработной платы для сотрудников. 18](#_Toc152266489)

[2.1 Выбор и создание программного решения: 18](#_Toc152266490)

[2.2 Добавление и удаление блюд из системы 18](#_Toc152266491)

[2.3 Управление добавлением в корзину 24](#_Toc152266492)

[2.4 Уровни доступа для пользователей 27](#_Toc152266493)

[2.5 Регистрация новых пользователей 29](#_Toc152266494)

[2.6 Тестирование программных модулей 30](#_Toc152266495)

[Выводы практической части 33](#_Toc152266496)

[Заключение 35](#_Toc152266497)

[Список использованной литературы 37](#_Toc152266498)

# Введение

Тема кассовых аппаратов столовой остаётся актуальной в современном мире, особенно в контексте общественного питания. Стремительное развитие технологий и повышение ожиданий со стороны клиентов создают потребность в более эффективном управлении бизнесом. Автоматизация учёта операций в столовых через разработку специализированной программы позволяет упростить процессы заказа, оплаты, учёта ингредиентов и составления отчётов.

Современные столовые и рестораны являются частью культурного и общественного пейзажа. Они предоставляют студентам не только возможность насладиться разнообразными блюдами, но и оформить уникальные места для общения, отдыха и проведения времени. Однако успешное ведение столового или ресторанного бизнеса требует не только высокого качества продуктов и обслуживания, но и точного учета всех финансовых операций.

Учет кассовых операций играет ключевую роль в обеспечении финансовой прозрачности и эффективности управления предприятием. Соблюдение бухгалтерских правил, финансовый учет доходов и расходов, а также детальные финансовые отчеты становятся важной частью процесса управления столовой или рестораном. Для автоматизации и упрощения этого процесса широко применяются специализированные программные решения.

Разработка подобной программы, использующей WinForms, становится значимой для столовых и ресторанов, так как она может помочь снизить издержки, упростить управление и улучшить обслуживание клиентов, что важно для сохранения конкурентоспособности и эффективности бизнеса в сфере общественного питания.

Объектом исследования в курсовой работе является приложение для учёта операций в кассе столовой. Это включает в себя все аспекты, связанные с учётом операций, заказами, блюдами и чеками в контексте работы столовой.

Предметом исследования является разработка программы с использованием технологии WinForms и языка программирования C# для автоматизации учёта операций в кассе столовой. Это включает в себя функции добавления, удаления и добавления в корзину блюд, а также генерацию чеков.

Методами исследования являются анализ, моделирование, эксперимент (программирование), тестирование.

Целью курсовой работы является разработка программы для учёта операций в кассе столовой с использованием WinForms в среде C#.

Исходя из цели курсовой работы были поставлены следующие задачи:

1. Проектирование программы для учёта операций в кассе столовой, включая разработку архитектуры приложения и интерфейса.
2. Разработка функциональности программы, включая возможность добавления, удаления и добавления в корзину блюд, а также вывод чека.
3. Разработка механизма записи и чтения данных из txt-файлов для учёта операций.
4. Проведение тестирования и отладки программы для обеспечения её корректной работы.
5. Анализ результатов и формулирование заключения, включая сводные выводы.

Курсовая работа состоит из 37 листов, а также содержит введение, две главы, заключения, список литературы.

# Глава 1. Теоретические основы разработки программы учёта операций в кассе столовой

## 1.1 Описание предметной области

Учет операций в столовой является важным фактором ее успешной деятельности. Независимо от размеров и типа учреждения и систематический учет операций в кассе стола играют решающую роль в обеспечении финансовой прозрачности и эффективности управления бизнесом. Рассмотрим более подробно различные аспекты в кассе столовой. Можно выделить шесть основных учетных операций, проводимых в столовых[7].

Одной из таких операций является прием заказов и обслуживание клиентов. Центральным элементом деятельности столовой является прием заказов и обслуживание клиентов. Учет операций в этой сфере включает в себя регистрацию заказов, фиксацию выбранных клиентами блюд и напитков, а также обработку заказов. Этот процесс также включает в себя проведение расчетов с клиентами, включая оплату заказов наличными, кредитными картами и другими доступными методами оплаты.

Так же можно выделить финансовые операции, которые играют центральную роль в учете столовой. Это включает оплату поставщиков сырья и материалов, выплату заработной платы персоналу, а также учет расходов на столовое обслуживание. Точный учет финансовых операций помогает контролировать бюджет, снижать затраты и максимизировать прибыль.

Создание отчетов и анализ данных также является важным аспектом. Они являются неотъемлемой частью учета операций в столовой. Это включает в себя генерацию финансовых отчетов, таких как балансы, отчеты о прибылях и убытках, а также отчеты о продажах и анализ популярности блюд. Эти отчеты помогают владельцам и управляющим принимать обоснованные решения, улучшать меню и повышать прибыль.

Не стоит забывать про соблюдение бухгалтерских стандартов и законодательства. Для успешного учета столовой деятельности необходимо строго соблюдать бухгалтерские стандарты и законодательные требования, особенно в отношении финансовой отчетности. Нарушение этих норм может привести к правовым и финансовым проблемам, поэтому соблюдение норм и стандартов является обязательной частью процесса учета операций в столовой.

Для упрощения и совершенствования процесса учета операций в столовой необходимо разработать программное решение на языке программирования C#. Это решение будет способствовать более эффективному управлению столовой и повышению качества обслуживания клиентов.

И последней важной операцией является анализ существующих программных решений: сегодня существует множество программных решений и инструментов для учета операций в кассе стола, таких как «Restik», «R\_keeper», «JOWi». Анализ этих решений играет важную роль при разработке нового программного продукта. Этот анализ помогает выявить сильные и слабые стороны существующих систем, а также определить, какие функциональные возможности можно интегрировать в разрабатываемую программу на C#.

На сегодняшний день существует множество программных решений и инструментов для учета операций в кассе столов и ресторанов, включая небольшие локальные приложения и комплексные облачные системы. Эти решения доступны как в форме коммерческих продуктов, так и в виде бесплатного и открытого программного обеспечения.

Преимущества существующих решений включают их надежность, готовность к использованию и поддержку, а также возможность регулярных обновлений. Однако коммерческие продукты могут быть дорогими, а некоторые решения могут иметь ограничения в настройке под конкретные потребности столовой.

Анализ этих существующих решений поможет выявить возможности для улучшения и оптимизации. Разрабатываемая программа на C# должна учесть эти возможности и предложить инновационные решения, обеспечивая особенности, такие как интеграция, настройка и доступная стоимость.

Таким образом, анализ существующих систем учета операций в кассе столовой является важным этапом в разработке нового программного продукта, который будет способствовать более эффективному управлению и повышению качества обслуживания клиентов в столовых и ресторанах[5].

## 1.2 Архитектура программы

Первым и наиболее важным этапом при разработке программного обеспечения для учета операций в кассе столовой является детальный анализ требований и предметной области. Это включает в себя изучение всех аспектов столовой работы, начиная с приема заказов и заканчивая составлением отчетов. Понимание настроений клиентов и учетных записей каждого стола – ключевой фактор для определения требований к программному продукту.

Следующим этапом будет выбор внешнего режима и режима ожидания. На этом этапе определяется, какая архитектура и шаблоны программирования будут использоваться при создании программы. Правильный выбор конфигурации и подхода к обеспечению масштабируемости, устойчивости и удобства сопровождения программы. Разработчики должны учитывать особенности учета операций, такие как обработка транзакций, безопасность данных и высокая производительность.

Следующим шагом будет определение требований к программе. Это включает в себя описание всех основных функций, которые должны быть реализованы в программе, начиная с регистрации заказов и заканчивая генерацией финансовых отчетов. Определение требований также включает в себя работу с данными, определение структуры базы данных и определение методов взаимодействия с пользователем.

Для обеспечения удобства использования программы важно разработать понятный и удобный пользовательский интерфейс. Дизайн играет важную роль в достижении бизнес-целей, поскольку он улучшает опыт взаимодействия с пользователем и удовлетворенность клиентов. Пользовательский интерфейс должен учитывать использование настольного персонала и обеспечивать простоту и эффективность взаимодействия с программой. Проектирование интерфейса также включает в себя разработку макетов, создание элементов управления и тестирование пользовательского опыта.

Исходя из пользовательского интерфейса нужно приступить к разработке программного обеспечения для учета операций в кассе стола включает в себя создание алгоритмов для выполнения учетных операций и обработки данных. Это включает в себя разработку логики обработки заказов, расчет стоимости блюд и многое другое. Также необходимо разработать алгоритмы для получения отчетов и статистики.

После разработки программы, необходимо провести тестирование и отладку, чтобы обеспечить ее надежность и безошибочную работу. Это включает в себя проверку всех функций и алгоритмов, а также проверку соблюдения требований безопасности и производительности.

Изучение разработки программного обеспечения для учета операций в кассе столовой – это сложный и ответственный процесс, который требует систематического и глубокого подхода. Важно учитывать все аспекты работы столовой и клиентов, чтобы создать программу, которая будет полезна для учета операций и управления бизнесом[4].

Архитектурный шаблон – это общая структура и организация программы, определяющая, как компоненты программы взаимодействуют между собой. Для ведения учета операций в кассе стола можно настроить следующие архитектурные шаблоны:

Шаблон «модель – представление – контроллер» (MVC) разделяет функциональность приложения на компоненты трех видов:

* модель – содержит данные приложения;
* представление – отображает некоторую часть базовых данных и взаимодействует с пользователем;
* контроллер – действует в качестве посредника между моделью и представлением и управляет уведомлениями об изменении состояния[10].

Микро сервисная организация: данный шаблон предлагает разделение приложений на небольшие независимые микросервисы, каждый из которых выполняет свою конкретную функцию. Этот подход обеспечивает гибкость и масштабируемость приложения. Например, можно реализовать микросервисы для учета заказов, управления ингредиентами и формирования отчетов.

Монолитный: в этой архитектуре все функциональные приложения объединены в единую систему. Это может быть удобно, если программа имеет небольшой масштаб и не требует трудоемких работ. Однако монолитные приложения могут быть менее масштабируемыми.

Помимо выбора архитектурного шаблона, необходимо определить подход к разработке программы.

Когда в проекте имеет множество высококвалифицированных специалистов может применяться разработка «с нуля». Полная разработка программы с нуля, позволяющая полностью контролировать каждый аспект приложения. Этот подход требует больше времени и ресурсов, но обеспечивает полную гибкость.

Проекты, которые не имеют достаточно средств, может подойти готовые решения: возможность использования дополнительных библиотек, фреймворков и компонентов для ускорения разработки. Это может уменьшить процесс, но потребует адаптации и настройки выбранных компонентов.

Для создание уникального и стабильного проекта используют гибридный подход: комбинация разработки «с нуля» и использование готовых решений. Например, можно создать собственный пользовательский интерфейс, но использовать готовую платформу для управления базой данных.

При выборе архитектурного шаблона и подход к разработке необходимо также учитывать особенности предметной области, в данном случае, операций в кассе столовой. Например, если столовая имеет хорошую загрузку и большое количество заказов, то необходимо обеспечить высокую производительность и масштабируемость. Если столовая обеспечивает доставку услуг, то интеграция с постоянными заказами и доставка становятся важным аспектом.

Осуществление анализа и сравнение различных архитектурных паттернов и подходов позволяет определить, какой из них наиболее подходит для конкретной задачи. Важно использовать настольный вариант, а также возможность будущего расширения и обновления программы.

В конечном итоге, выбор архитектурного решения – это важный шаг в разработке программы учета операций в кассе столовой на языке C#. Это влияет на конструкцию, производительность и гибкость программы и должно быть тщательно продумано и адаптировано с учетом конкретного вида и характеристик стола, который она будет обслуживать.

## 1.3 Функциональные требования

Одной из основных функций программы является возможность регистрации заказов от клиентов. Сотрудники столовой должны иметь доступ к программе, чтобы фиксировать выбранные клиентами блюда, напитки и другие услуги. Это включает в себя возможность быстрого внесения изменений в заказ, сохранения количества и соблюдения определенных требований.

Программа предоставляет необходимые функции для управления столовым меню. Это включает в себя добавление и удаление блюда, изменение цены, а также отслеживание ингредиентов и ингредиентов для приготовления блюда.

Программа автоматически определяет стоимость заказа на основе выбранных клиентом позиций и их количества. Это включает в себя применение цен из меню и учет скидок или налогов, которые могут повлиять на общую стоимость.

Программа также предоставляет необходимые функции для составления различных отчетов и анализа данных. Это включает в себя генерацию финансовых отчетов. Генерация отчетов помогает владельцам столовой принимать обоснованные решения и оптимизировать бизнес-процессы.

Программа обеспечивает безопасность данных и ограничение доступа только сотрудникам столовой с соблюдением правил. Это важно для защиты конфиденциальных данных клиентов и финансовой информации.

Общее описание основных функций программы для учета операций в кассе стола включает множество аспектов, которые должны быть учтены при ее разработке. Эти функции позволяют эффективно управлять бизнесом, обеспечивать высококачественное обслуживание клиентов и контролировать финансовую стабильность предприятия.

Данные о меню и блюдах: название блюда, описание блюд, Цена за порцию, категория блюд (например, супы, горячие блюда, напитки) и их количество.

Данные о клиентах: имя и фамилия клиента, Контактная информация (телефон, адрес, электронная почта).

Данные о заказах: номер заказан, Дата и время заказа, Список блюд и напитков, включая количество и стоимость каждого, Сумма заказа.

Данные о персонале: имя и фамилия сотрудника, Должность и роль в столовой, Учетные данные для входа в систему.

Данные о финансах: общая выручка и прибыль стола, Список всех транзакций и операций по оплате, Список налогов и скидок, применяемых к заказам, Финансовые отчеты (балансы, отчеты о прибылях и убытках).

Создание заказов: сотрудники столовой могут формировать новые заказы, выбирая блюда из меню, указывая количество порций и удовлетворяя потребности клиентов.

Редактирование заказа: возможность изменения состава заказа или его последствия.

Генерация отчетов: создание различных отчетов о финансах, продажах, статистике и других аспектах столовой работы.

Определение необходимого набора данных и операций над ними является фундаментальным шагом при разработке программы для учета операций в столовой кассе. Это обеспечивает полную функциональность системы, учитывает и анализирует все аспекты столовой работы и позволяет обеспечить качественное обслуживание клиентов[12].

## 1.4 Проектирование пользовательского интерфейса

Изучение проектирования удобного и внедрение понятного пользовательского интерфейса (UI) играет решающую роль в разработке программы для учета операций в кассе стола. Проектирование пользовательского интерфейса требует внимания к множеству аспектов, которые позволяют сделать пользовательский опыт удобным и эффективным.

Важно начать с глубокого понимания интерфейсов приложений. Знание настроения и ожиданий пользователей позволяет создать интерфейс, который будет наиболее соответствовать их требованиям. Интуитивность и простота пользовательского интерфейса играют важную роль. Уменьшение лишних элементов и простые инструкции помогают пользователям легко ориентироваться в приложении и выполнять задачи.

Согласованность пользовательского интерфейса дает ощущение единообразия и узнаваемости. Она включает в себя использование одинаковых стилей, цветовых палитр, шрифтов и элементов управления во всем мире. Это быстрая навигация и создает бренд-идентичность.

Необходимо выбрать организацию информации в самом предложении. Логическая и удобная структура данных, четкая иерархия, категоризация и метки помогают пользователям быстро находить необходимую информацию. Уведомления и обратная связь важны для информирования пользователей о нарушениях и ошибках.

Тестирование на реальных пользователях позволяет выявлять проблемы на ранних стадиях разработки и учитывать предпочтения пользователей. Адаптивность интерфейса обеспечивает хорошую работу приложений на разных устройствах. Соблюдение современных дизайн-трендов и лучших практик в UI-дизайне помогает приложению оставаться актуальным и привлекательным для пользователей.

Для создания актуального пользовательского интерфейса необходимо совместное участие дизайнеров и разработчиков в процессе программного продукта, учет мнений пользователей и актуальных требований для создания интерфейса, который соответствует этим требованиям и обеспечивает удовлетворительный пользовательский опыт[9].

## 1.5 Описание алгоритмов работы программы

Разработка алгоритмов для выполнения учетных операций и обработки данных – это один из наиболее важных этапов создания программы для учета операций в кассе столовой. Эти алгоритмы определяют фундаментальную функциональность приложения, определяя, как приложение будет взаимодействовать с данными и выполнять ключевую операцию.

Важно учитывать, что эти алгоритмы должны быть разработаны с учетом требований бизнес-процессов. Это включает в себя регистрацию заказов, управление меню и едой, расчет стоимости заказов, обработку платежей, управление столами и местами, составление отчетов и анализ данных, обработку заказов, учет ингредиентов и поставок, а также управление резервированием столов.

При разработке алгоритмов следует уделять внимание эффективности и оптимизации, чтобы обеспечить контроль обработки данных и операций. Также важно обеспечить безопасность и защиту данных, особенно при работе с наличными и личными данными.

Кроме того, алгоритмы должны быть легко расширяемыми и изменяемыми, чтобы учитывать будущие производители и изменения в настольных бизнес-процессах. Это позволит приложению адаптироваться к изменяющимся условиям и потребностям.

Общий успех программы учета операций в столовой кассе зависит от тщательной и грамотной разработки алгоритмов, которые обеспечивают надежную и эффективную работу системы печати.

Расчет и описание основных формул и алгоритмов для расчета статистики и отчетности являются частью разработки программы для учета операций в столовой кассе. Эти алгоритмы позволяют получать ценную информацию о финансовых и операционных аспектах столовой, что, в свою очередь, помогает найти важные управленческие решения и оптимизировать бизнес-процессы.

Для расчета общей выручки необходимо просуммировать стоимость всех выполненных заказов за текущий период. Это включает в себя подсчет стоимостей всех позиций заказов и учет скидок или налогов.

Прибыль рассчитывается как разница между выручкой и затратами, включая стоимость ингредиентов, затраты на персонал, аренду и другие расходы. Формула: прибыль = выручка - затраты.

Расчет среднего чека позволяет определить среднюю сумму, которую клиенты ведут в столовой за один прием. Для его расчета необходимо учитывать выручку от количества заказов за текущий период.

Анализ популярности блюд: этот метод помогает определить, какие блюда наиболее популярны среди клиентов, их популярность может быть вычислена на основе количества заказов каждого блюда.

Расчет налогов и скидок: для лога учетной отчетности необходимо учитывать налоги, которые принимаются с заказами, а также скидки, предоставляемые клиентам. Эти расчеты могут быть сложными в зависимости от законодательства и политики столовой.

Генерация финансовых отчетов: алгоритмы формирования данных в виде финансовых отчетов, включая баланс, отчет о прибылях и убытках, позволяют принять обоснованные решения и отслеживать финансовое состояние стола.

Важным моментом является то, что эти алгоритмы могут быть дополнены или модифицированы в зависимости от требований столовой и бизнес-процессов. Грамотная разработанная система расчетов и анализа данных предоставляет управляющим ценную информацию для принятия решений и оптимизации работы стола.

## Выводы по теоретической части

В первом главе курсовой работы было проведено масштабное внедрение в разработку программы для учета операций в кассе столовой. Начиная с обзора учетных операций, что необходимо понять в разработке, какие процессы и операции необходимо учитывать в разработанной программе. Затем провелся анализ существующих решений в этой области, который помог изучить успешные практики и избежать встречных ошибок.

Далее обратим внимание на архитектуру программы, определение принципов разработки и архитектурных шаблонов, а также выбор языка программирования. Эти данные создадут техническую основу для будущих разработок.

Также были описаны основные функции программы, задачи и цели, которые должны быть реализованы в приложении. Важным этапом изучения стал переход к проектированию удобного и понятного пользовательского интерфейса, который обеспечивает комфортное взаимодействие пользователей с программой.

Следующим шагом будет разработка алгоритмов для учета выполнения операций и обработки данных. Эти алгоритмы обеспечивают работоспособность системы, включая регистрацию заказов, управление меню, расчет стоимости заказов и другие важные аспекты.

Подводя итоги, первая глава привела к фундаментальному пониманию предметной области, требований и архитектурных решений, которые будут использоваться при разработке программы для учета операций в кассе стола. Дальнейшая работа будет включать более детальное проектирование и реализацию функциональности, а также тестирование и оптимизацию приложений с учетом знаний, полученных в главе.

# Глава 2 Моделирование и разработка программного решения для расчета заработной платы для сотрудников.

## 2.1 Выбор и создание программного решения:

Лучшим языком программирования для создания программного решения будет являться С#, потому что благодаря отсутствию заголовочных файлов, более простой системе пространств имен и меньшим проблемам с управлением памятью, C# похож на элегантного двоюродного брата C++, который проще для новичков, в то же время позволяя опытным разработчикам писать более понятный и легко поддерживаемый код[8].

Структура проекта будет организована с учетом архитектурного шаблона MVC. Это включает в себя разделение кода на модели, контроллеры и представления, что облегчит поддержку и расширение приложения в будущем.

Для начала проектирования программы и разработки кода была выбрана интегрированная среда разработки (IDE) Visual Studio, которая предоставляет мощные инструменты для создания .NET-приложений. Visual Studio обеспечивает комфортное окружение для разработчиков, включая редактор кода, отладчик, инструменты для тестирования и другие полезные функции.

Выбор архитектурного решения, языка программирования и инструмента разработки является первым и важным шагом в создании программы для учета операций в кассе столовой. Эти решения обеспечивают надежность, производительность и удобство разработки, что сделает программу эффективным инструментом для учета операций и управления столовой.

## 2.2 Добавление и удаление блюд из системы

В данном разделе будет изучен процесс добавления и удаления блюд в программе для учета операций в кассе столовой. Эти операции являются ключевыми, так как они позволяют актуализировать меню столовой, учитывать сезонные изменения и удовлетворять потребности клиентов.

Следующим этапом будет разработка модуля, который позволит добавлять новые блюда в базу данных системы. Для обеспечения возможности добавления новых блюд в корзину был создан соответствующий модуль. Предоставленный интерфейс сделан для ввода информации о новом блюде, такой как название, описание, вес и цена. Пример интерфейса для добавления и удаления блюда продемонстрирован на рисунке 2.1.

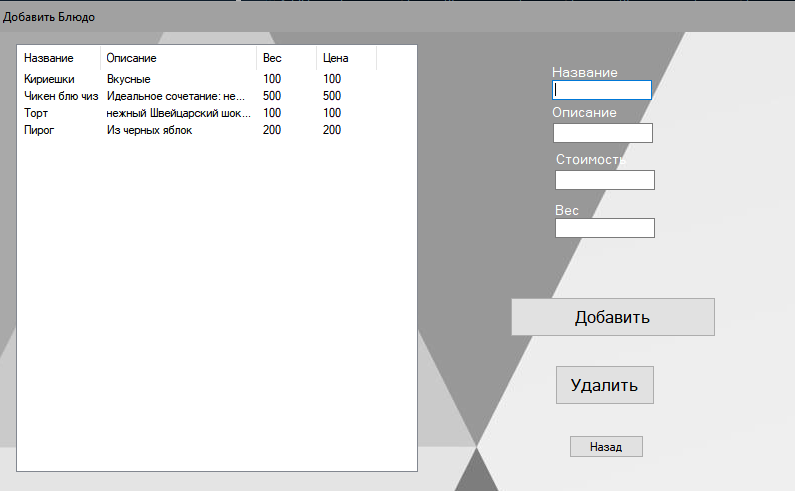


Рисунок 2.1 – Создание интерфейса формы добавления и удаления блюд

Интерфейс, изображенный на рисунке 2.1, предоставляет возможность администраторам или ответственным лицам (кассирам) вводить информацию о новых блюдах. Один из важных аспектов - обеспечение валидации данных, чтобы предотвратить возможные ошибки, такие как проверка правильности формата цены и обязательности заполнения названия блюда.

Помимо добавления блюд, система также предоставляет возможность удаления блюд из базы данных. Это важно в силу изменений в меню со временем, когда блюда могут устаревать или становиться недоступными. Создан соответствующий модуль, который позволяет выбирать блюда для удаления и подтверждать операцию.

Важным аспектом при добавлении и удалении блюд является проверка данных, включая правильность введенных значений, обязательных полей, и защиту от возможных атак или ошибок ввода. Все данные, введенные в интерфейс, проходят проверку перед сохранением в текстовых файлах.

Примеры валидации включают в себя:

* проверку правильности формата цены блюда;
* обязательное заполнение полей, таких как название блюда;
* обеспечение валидации и контроля данных при добавлении и удалении блюд помогает сохранить целостность и надежность информации в системе.

Таким образом, модуль добавления и удаления блюд играет важную роль в обеспечении актуальности и надежности данных в программе для учета операций в кассе столовой. Важно уделять особое внимание валидации и контролю данных, чтобы предотвратить возможные ошибки и обеспечить корректную работу системы.

Следующим этапом представлен код, состоящий из нескольких модулей.

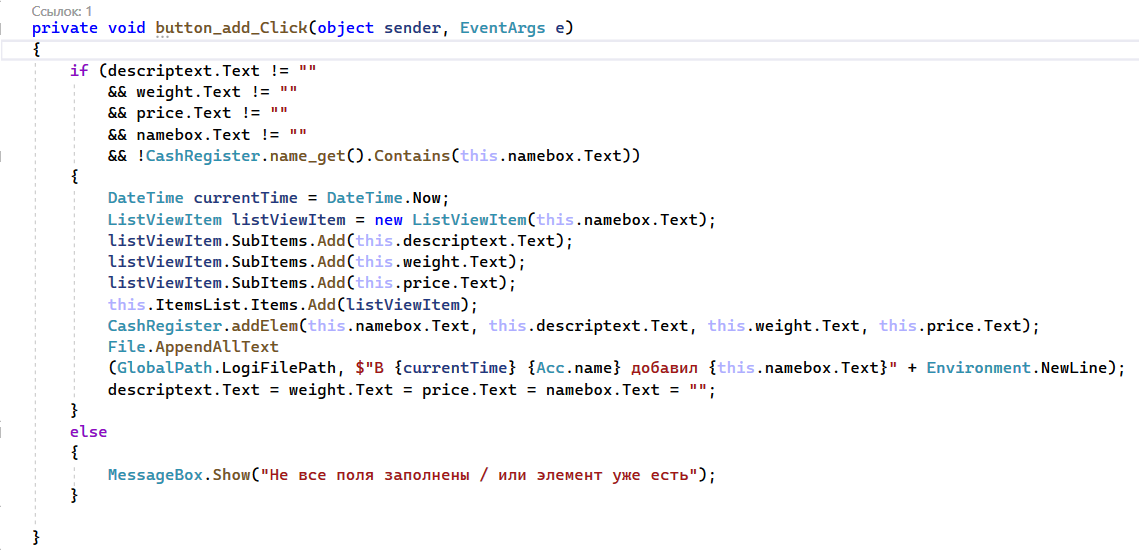


Рисунок 2.2 – Код модуля добавления блюд

Осуществление программы происходит через основной класс «CashRegister», в котором созданы методы работы с данными и основной функционал программы.

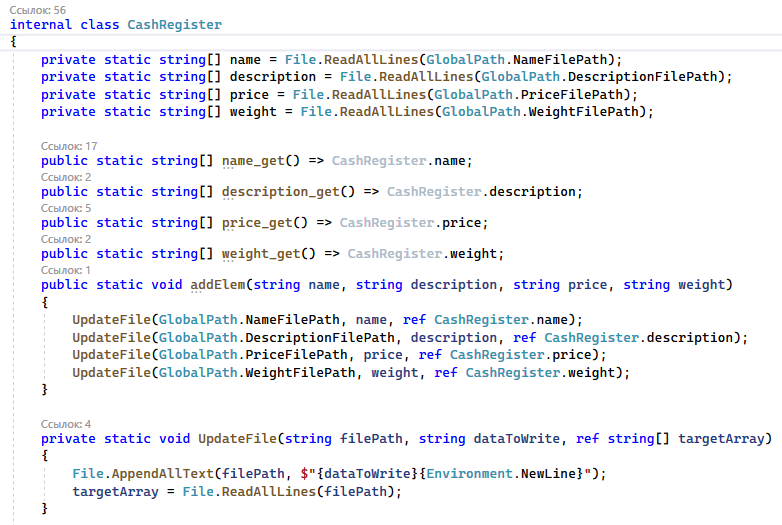


Рисунок 2.3 – Код основного класса

В модуле добавления блюд, изображенный на рисунке 2.2, используются label, textbox и list view[9]. Для начала работы добавления блюд необходимо заполнить все поля и после этого нажать на кнопку и после этого активируется метод «addElem», который принимает четыре строки и служит для добавления информации об элементе в четыре разных текстовых файла. Эта информация записывается в соответствующий файл, и затем эти данные также сохраняются в статических полях класса «CashRegister», который изображен на рисунке 2.3 и только после этого код читает данные из файла «name.txt» и сохраняет их в массив «arr».

Затем происходит проверка, что все текстовые поля «descriptext», «weight», «price» и «namebox» не пусты, и что значение в «namebox» не совпадает с каким-либо элементом в массиве «arr». Если это условие выполняется, то создается новый элемент для отображения данных в списке элементов и записывается информация в файл «logi.txt» о добавлении элемента с указанием времени и имени пользователя.

Если какое-либо из условий не выполняется (например, одно из текстовых полей пустое или элемент с таким именем уже существует), то выводится сообщение об ошибке с помощью «MessageBox».

В целом, этот код добавляет информацию о новом товаре в файлы данных, отображает этот товар в интерфейсе и регистрирует действие в журнале событий.

Для обеспечения актуальности меню и возможности удаления устаревших блюд был разработан метод удаления. Ниже на рисунке 2.4 представлен фрагмент кода на C#, демонстрирующий реализацию этого метода:

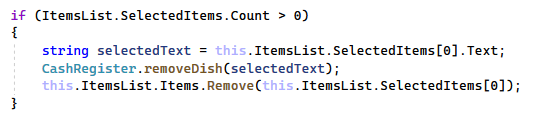


Рисунок 2.4 – Код удаления блюд

При нажатии на кнопку будет производиться код с рисунка 2.4 и вызываться метод с иллюстрации 2.5. Сначала происходит проверка на выделение блюда, затем запись в переменную «selectedText» и выбираться элемент, далее активируется метод «CashRegister.removeDish()», и после этого производится удаление из list view.

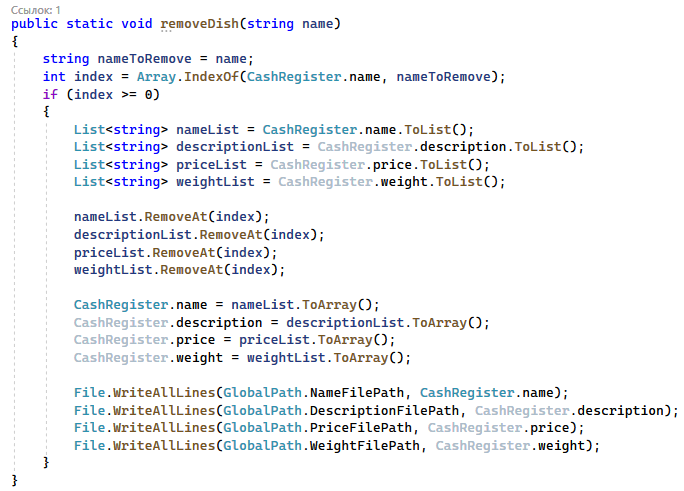


Рисунок 2.5 – Код метода удаления блюд

На рисунке 2.5 реализован метод «CashRegister.removeDish()», который предназначен для удаления информации о блюде из системы учета операций в кассе столовой. В данном коде начинаются указания путей к файлам, где хранится информация о блюдах. Затем необходимо проверить имя объекта с именем nameToRemove в переменной «CashRegister.name». Если блюдо найдено (проверка условий с использованием «Array.IndexOf», код продолжает выполнение. При успешном прохождении условий применяются временный простейший список однотипных объектов, основанные на массивах из «CashRegister». Заранее текущий индекс объектов, которые необходимо удалить, используется для удаления соответствующих записей из временных списков. Затем происходит обновление массивов в «CashRegister», и удаленное блюдо больше не присутствует в них. Наконец, метод «File.WriteAllLines» используется для перезаписи файлов «name.txt», «description.txt», «price.txt», «weight.txt» с обновленными данными после удаления блюд.

Этот метод обеспечивает функциональность удаления блюда из системы, обновляя соответствующие файлы с информацией о блюдах в процессе.

Важной частью разработки данных методов является валидация данных, чтобы предотвратить ошибки и поддерживать целостность данных. В коде приведены примеры проверок на уникальность блюда и наличие блюда перед удалением. Такие проверки могут включать в себя также проверку на корректность цены, наличие обязательных полей и другие аспекты, в зависимости от конкретных требований системы.

Эти методы обеспечивают надежный и безопасный способ добавления и удаления блюд в системе, а также включают проверки, которые способствуют поддержанию целостности данных в системе учета операций в кассе столовой.

## 2.3 Управление добавлением в корзину

В этом разделе описывается разработка методов для управления корзиной заказа в системе учета операций в кассе столовой. Эти методы позволяют добавлять и удалять блюда из корзины и выполнять расчет общей стоимости заказа.

Для начала работы товара необходимо выбрать элемент из списка продуктов как продемонстрированно на рисунке 2.6.

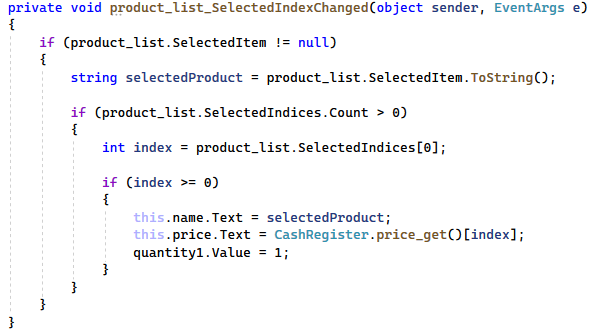


Рисунок 2.6 – Код проверки выбран ли элемент

На рассматриваемом рисунке продемонстрирован процесс выполнения алгоритма. Если выбранное не ровняется ничему, происходит запись переменной «selectedProduct», после этого проверяем находится ли индекс в листе, следующим действием производим запись в label данных о товарах

Пользовательский интерфейс, показанный на рисунке 2.7, реализован через «Tab Control»[2], в котором есть две вкладки: меню и корзина, мы должны выбрать блюдо, указать его количество и после этого добавить в корзину.

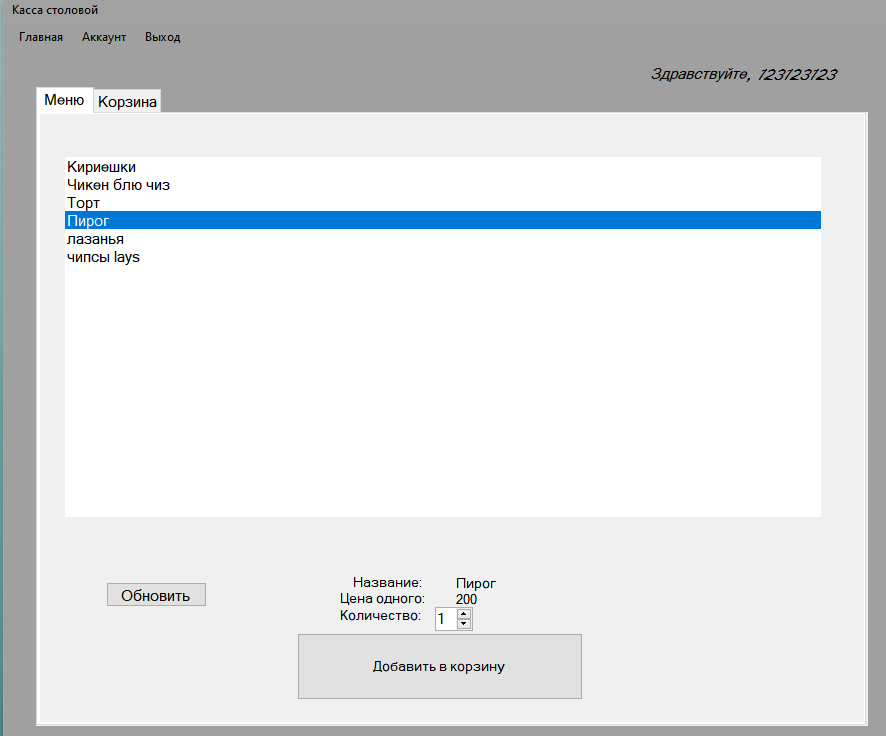


Рисунок 2.7 – Форма добавления блюд в корзину

Код с изображения 2.8 представляет собой обработчик события при нажатии на кнопку «to\_shopping\_basket» в пользовательском интерфейсе.

При выборе продукта из списка «product\_list» код проверяет, был ли выбран какой-либо продукт. Если ничего не было выбрано, выводится сообщение об ошибке. В противном случае, код продолжает выполнение.

Определяется индекс выбранного продукта в массиве «CashRegister.name\_get()».Затем он проверяет, находится ли выбранный продукт уже в корзине покупок («shopping\_basket»). Если да, то извлекает количество и цену выбранного продукта, обновляет их, а также записывает информацию о действии (добавлении продукта) в журнал («logi.txt»).

Если выбранный продукт не находится в корзине, код извлекает количество и цену, рассчитывает общую сумму, добавляет запись о продукте в корзину для отображения в интерфейсе, и также записывает информацию о действии в журнал логов.

Наконец, код обновляет общую сумму в интерфейсе на основе добавленного продукта и сбрасывает значение поля «quantity1.Value» на 1.

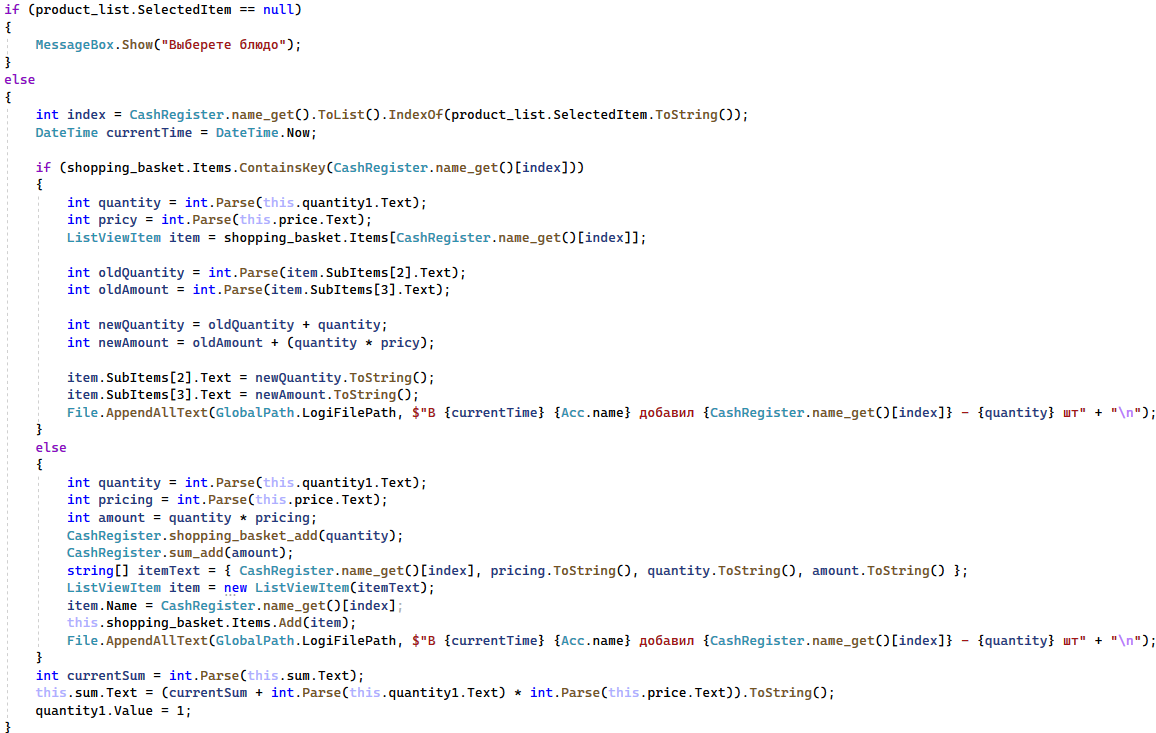


Рисунок 2.8 – Код добавления блюд в корзину

Этот код обеспечивает возможность пользователя добавлять продукты в корзину покупок, учитывать количество и цены, а также вести журнал операций.

В корзине пользователь может выбрать опцию экспорта итога покупки в текстовый файл (txt). Это позволяет сохранить информацию о продуктах в корзине, их количестве, сумме и общей сумме покупки для последующего использования или архивирования.

Сам интерфейс, продемонстрирован на рисунке 2.9, использует кнопки и listview[1].

Также пользователь может удалить выбранный продукт из корзины покупок, что особенно полезно, если он передумал покупать определенный товар. Также предусмотрена опция для удаления всех элементов из корзины покупок, что позволяет начать покупки заново или очистить корзину после завершения покупки.

Эти действия позволяют управлять содержимым корзины покупок и обеспечивают пользователю контроль над выбранными продуктами, а также возможность сохранить информацию о покупках для будущего использования или архивирования чеков.

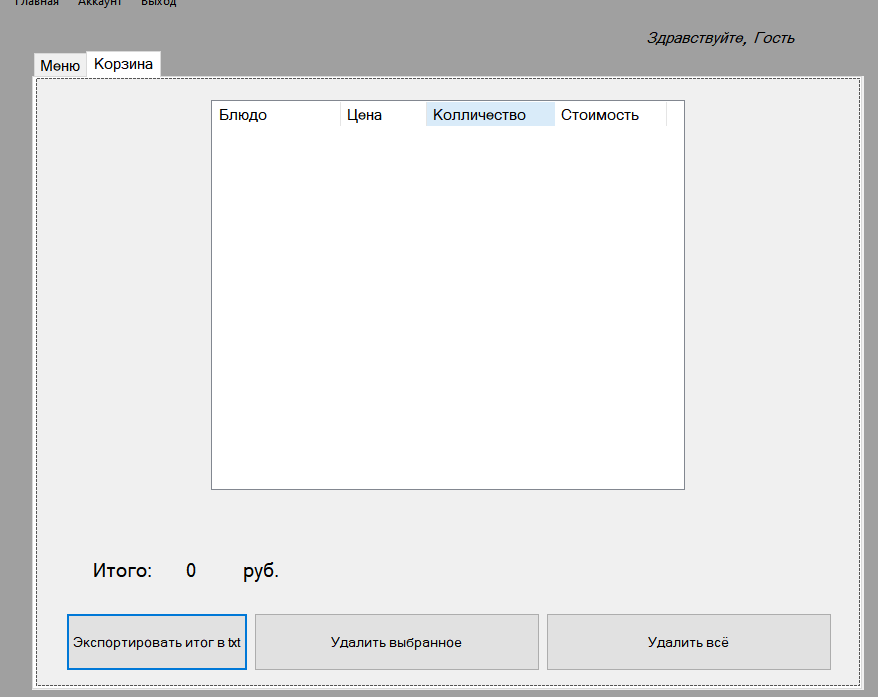


Рисунок 2.9 – Интерфейс добавления блюд в корзину

Возможность удалять элементы доступна только после того, как выбрали элемент, если элемент не был выбран, то программа не совершает никакие действия, так же существует возможность очищать корзину, при нажатии на данную кнопку все данные очищаются и список становится пустым

Так же в программном решении присутствует возможность экспорта. При экспорте всех элементов корзины необходимо выбрать название файла и после выбора места хранения чека появиться возможность его посмотреть, при этом корзина очищается и начинает новая история покупок.

## 2.4 Уровни доступа для пользователей

В программе существует несколько уровней доступа:

* 1. «Admin» – администратор программы, обладающий полными правами. Этот уровень доступа позволяет администратору редактировать меню, добавлять и удалять блюда, а также управлять настройками программы.
  2. «Cashier» – кассир, является пользователем программы с ограниченными правами. Кассиру разрешено добавлять, удалять и очищать корзину заказа, а также экспортировать чеки для клиентов.
  3. «None» – это уровень доступа для пользователей, у которых отсутствуют специальные права доступа. Они могут использовать программу только для просмотра меню, и не могут изменять его содержимое или осуществлять административные действия.

Проверка уровня доступа осуществляется через проверку соответствий условий продемонстрирована на рисунке 2.10.

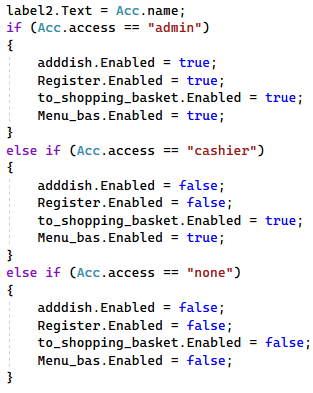


Рисунок 2.10 – Код проверки условий доступа

На рассматриваемом рисунке изображён код, который установлен в событие загрузки формы, что позволяет его активации при первом открытии формы. Также он активируется при наведении на «menuStrip». Это обеспечивает управление доступом в зависимости от уровня доступа пользователя при различных сценариях использования программы.

## 2.5 Регистрация новых пользователей

В программном решении предоставлена возможность регистрации только администратору. Это означает, что только пользователь с уровнем доступа «администратор» имеет право регистрировать новых пользователей в системе. Остальные пользователи, такие как кассиры или те, у которых уровень доступа «none» не имеют доступа к этой функции. Это предостерегает от несанкционированной регистрации и обеспечивает контроль доступа в системе.

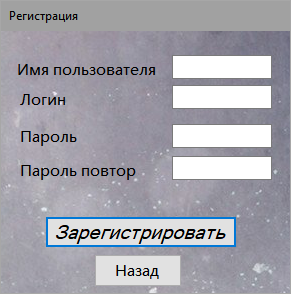


Рисунок 2.11 – Форма регистрации

При активации кнопки «Регистрация» на форме, как показано на рисунке 2.11, код изображения 2.12 осуществляет проверку заполненности текстовых полей с логином, паролем и подтверждением пароля. Если хотя бы одно из полей содержит менее двух символов, система выдаст сообщение об ошибке. Затем код проверяет доступность выбранного логина, считывая логины из файла.

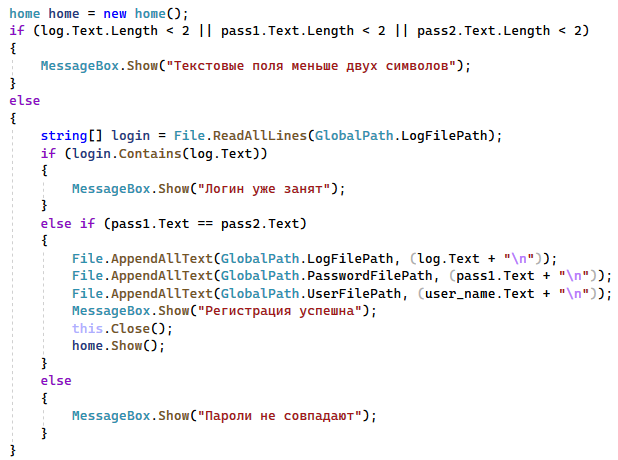


Рисунок 2.12 – Код модуля регистрации

При совпадении паролей, логин, пароль и имя пользователя добавляются в соответствующие файлы, а затем выводится сообщение об успешной регистрации. В случае несовпадения паролей система выдаст сообщение об ошибке. После успешной регистрации форма автоматически закрывается, а открывается главное окно.

## 2.6 Тестирование программных модулей

Тестирование программных модулей является важным этапом в разработке, так как оно позволяет выявить и исправить ошибки и недоработки, обеспечивая надежность и безошибочную работу программы. Тестирование гарантирует, что каждый модуль функционирует корректно, а также взаимодействие между модулями не вызывает конфликтов или сбоев. Это улучшает качество программы, повышает уровень доверия пользователей и снижает вероятность возникновения проблем в процессе эксплуатации.

Для обеспечения надежности и качества программного продукта было проведено тестирование, охватывающее различные аспекты функциональности. Созданы тестовые случаи для каждого метода и модуля, что позволило выявить и устранить возможные ошибки и недоразумения.

Было проведено модульное тестирование**,** в ходе которого каждый метод и функциональность тщательно проверялись в изоляции. Это позволило убедиться в корректности и надежности отдельных частей программы.

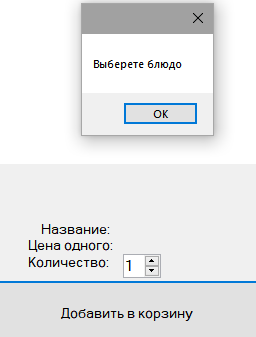


Рисунок 2.13 – Тестирование модуля добавления в корзину

На иллюстрации 2.13 демонстрируется интерфейс модуль для добавления блюда в корзину. При активации кнопки «Добавить» необходимо произвести выбор, иначе возникает исключительная ситуация.

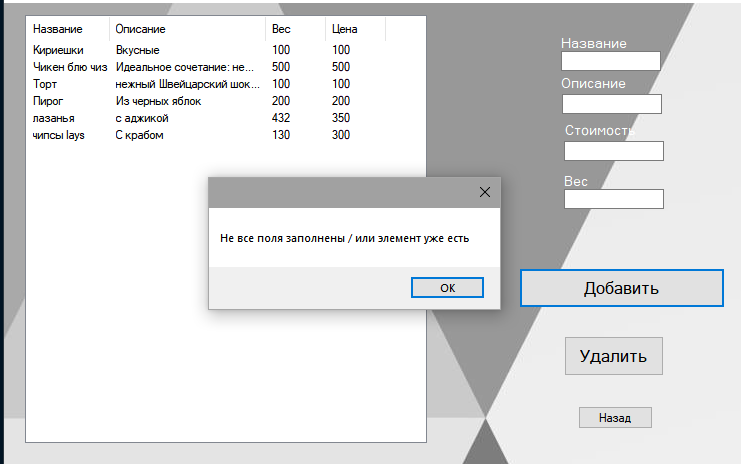


Рисунок 2.14 – Тестирование модуля добавления блюд в систему

При рассмотрении рисунка 2.14 можно выделить важный момент в модуле добавления блюд в систему. В данном контексте реализован механизм обработки исключений, который эффективно реагирует на две ситуации: существующее блюдо с таким же именем или неполноту ввода данных. В случае обнаружения дубликата или неполного ввода, система генерирует исключение, предотвращая выполнение операции и уведомляя пользователя о необходимости корректного ввода данных. Это обеспечивает более безопасную и надежную работу программы, предотвращая возможные ошибки в учете блюд.

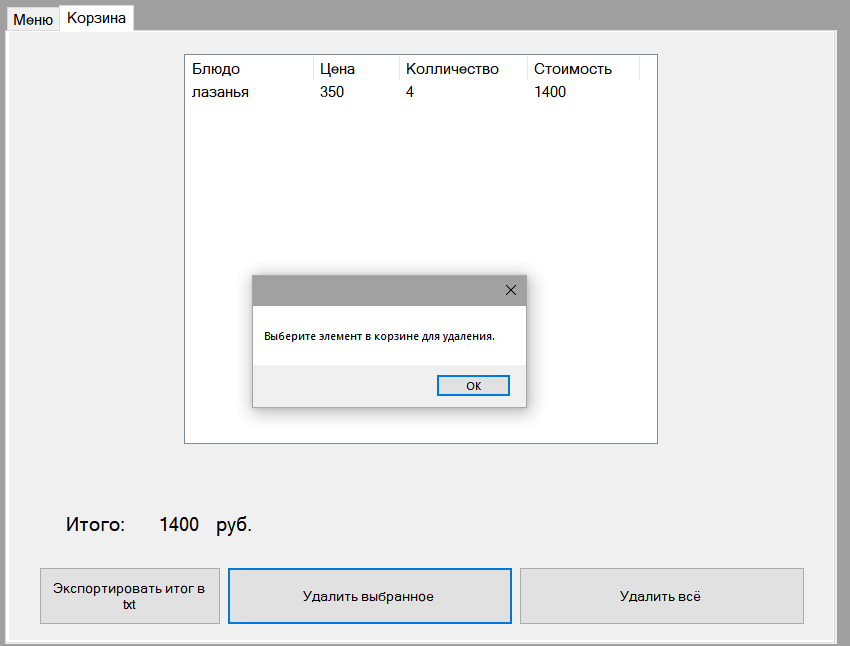


Рисунок 2.14 – Тестирование модуля удаления из корзины

В контексте удаления блюд из корзины, показанной на рисунке 2.14, важно отметить несколько ключевых моментов. Первоначально, перед выполнением операции удаления необходимо выбрать элемент в корзине заказа. В случае, если элемент не выбран, система предоставляет соответствующее уведомление, указывая на необходимость выбора конкретного блюда перед удалением.

Кроме того, стоит учитывать сценарий, при котором корзина заказа пуста. В такой ситуации операция удаления не имеет смысла, поэтому система должна предусмотреть соответствующее предупреждение или блокировку возможности удаления, чтобы избежать потенциальных ошибок.

Подводя итоги, результаты проведенного тестирования выявили и успешно устранили обнаруженные ошибки. Это существенно улучшило надежность программы. Кроме того, тестирование выявило моменты, где функциональность могла быть улучшена, и эти улучшения были успешно внесены. Таким образом, тестирование сыграло ключевую роль в обеспечении качества и надежности программного продукта.

## Выводы практической части

В практической части курсовой работы был представлен процесс разработки программы для учёта операций в кассе столовой. Основываясь на детальном анализе требований и предметной области, были выбраны соответствующие архитектурные шаблоны и технологии для создания программного продукта. В результате, была разработана система, способная учета заказов, управления меню, и обеспечивающая удобство взаимодействия с пользователем.

Каждый этап разработки, начиная с архитектурного проектирования и выбора языка программирования, заканчивая разработкой алгоритмов и методов взаимодействия с данными, был детально рассмотрен и реализован. Пользовательский интерфейс был спроектирован с учетом удобства использования настольного персонала.

Особое внимание было уделено тестированию, включая модульное, интеграционное, функциональное и системное тестирование[8]. Это позволило выявить и устранить ошибки, а также убедиться в надежной работе программы.

Важной составляющей работы стала регистрация пользователей, при этом уровень доступа предоставляется только администратору. Это обеспечивает безопасность и контроль над системой.

Таким образом, результатом практической части работы стала разработка программы для учета операций в кассе столовой с учетом всех требований и нормативов, обеспечивая надежность, безопасность и удобство использования.

# Заключение

В современном бизнесе автоматизация играет важную роль, и столовые не исключение. Внедрение кассовых аппаратов и программных решений для учета операций в столовой позволяет оптимизировать процессы и улучшить обслуживание клиентов. Основываясь на проведенных исследованиях и разработках, были сделаны следующие выводы.

Внедрение кассовых аппаратов и программного обеспечения в столовых позволяет ускорить обработку заказов, уменьшить вероятность ошибок и улучшить точность учета продукции. Это способствует повышению качества обслуживания клиентов и увеличению их удовлетворенности.

Кроме того, автоматизация учета операций в столовой позволяет эффективно управлять меню, добавлять и удалять блюда, контролировать доступ пользователей и регистрировать новых клиентов. Это снижает нагрузку на персонал и позволяет предоставлять более высокий уровень сервиса.

Оценка степени достижения поставленных целей и задач показывает, что внедрение автоматизации в столовых оправдывает себя. Быстрое и точное оформление заказов, учет продукции и управление меню обеспечивают эффективное функционирование предприятия.

Практическая значимость работы заключается в том, что разработанные программные модули и кассовые аппараты способствуют улучшению работы столовых, повышению их конкурентоспособности и удовлетворенности клиентов.

В перспективе предполагается углубленное развитие и усовершенствование системы учета операций в столовой. Планируется расширение функционала системы, включая более детальные отчеты, аналитические инструменты и статистические данные для более подробного анализа работы столовой.

Таким образом, все поставленные задачи курсовой работы были успешно выполнены. Был проведен полный цикл исследований и разработок, включая изучение требований к кассовым аппаратам в столовых, а также освоение основ C# и инструментов разработки в среде WinForms.

Основной результат работы заключается в разработке программы для учета операций в кассе столовой, включая архитектуру приложения и интерфейс.

Эта программа обладает функциональностью, позволяющей добавлять, удалять блюда и оформлять чеки, а также осуществлять учет операций с использованием txt-файлов.

Тестирование и отладка программы были проведены, что позволило обеспечить ее стабильную и безошибочную работу. Результаты исследований подтвердили важность внедрения автоматизации в столовые и рестораны, что способствует повышению качества обслуживания клиентов и оптимизации процессов.

В целом, проделанная работа позволяет сделать вывод о успешном выполнении поставленных целей и задач курсовой работы. Разработанное программное решение имеет практическую значимость и перспективы дальнейшего развития, что делает его актуальным в современном мире.

# Список использованной литературы

1. How to: Add and Remove Items with the Windows Forms ListView Control [Электронный ресурс] Режим доступа: – https://learn.microsoft.com/en-us/dotnet/desktop/winforms/controls/how-to-add-and-remove-items-with-the-windows-forms-listview-control?view=netframeworkdesktop-4.8
2. Tab Control Класс [Электронный ресурс] Режим доступа: – https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.forms.tabcontrol?view=windowsdesktop-7.0
3. Документация по C# [Электронный ресурс] Режим доступа: –https://learn.microsoft.com/ru-ru/dotnet/csharp/
4. Жизненный Цикл Разработки ПО [Электронный ресурс] Режим доступа: – https://vc.ru/u/700268-marketing-solveit/199225-chto-takoe-zhiznennyy-cikl-razrabotki-po-i-kakie-problemy-voznikayut-na-kazhdom-etape-sdlc
5. Какие программные решения существуют [Электронный ресурс] Режим доступа: – https://trends.rbc.ru/trends/education/63357fa49a79477a3c39f614
6. Кассовое программное обеспечение [Электронный ресурс] Режим доступа: – https://www.retail.ru/rbc/pressreleases/kassovoe-programmnoe-obesp-echenie-chto-eto-takoe/
7. Контрольно-кассовая машина [Электронный ресурс] – Режим доступа: https://w.wiki/88D6
8. Особенности тестирования «черного ящика» [Электронный ресурс] Режим доступа: – https://quality-lab.ru/blog/key-principles-of-black-box-testing/
9. Почему дизайн важен [Электронный ресурс] Режим доступа: – https://www.uprock.ru/articles/pochemu-ui-ux-dizayn-vazhen-dlya-vashego-biznesa
10. Самые важные архитектурные шаблоны [Электронный ресурс] – Режим доступа: https://habr.com/ru/companies/alconost/articles/522662/
11. Элементы управления C# [Электронный ресурс] Режим доступа: – https://metanit.com/sharp/windowsforms/4.1.php
12. Элементы управления Windows Forms [Электронный ресурс] Режим доступа: – https://learn.microsoft.com/ru-ru/dotnet/desktop /winforms/control-s/? view=netframeworkdesk