



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ  
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Экономики и управления на предприятии природопользования»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
(бакалаврская работа)  
по направлению подготовки 09.03.03 «Прикладная информатика»  
(квалификация – бакалавр)

На тему «Проектирование информационной системы для автоматизации деятельности предприятия сферы общественного питания»

Исполнитель Абян Алэн Артынович

Руководитель к.геогр.н, доцент по кафедре информатики и прикладной математике  
Полупанов Владимир Николаевич

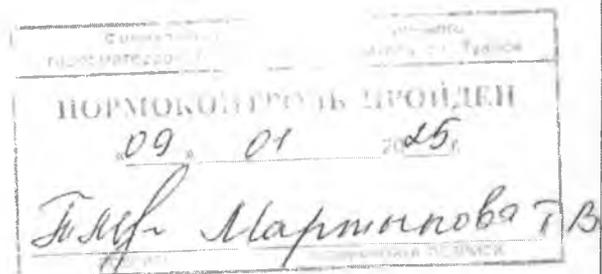
«К защите допускаю»

Руководитель кафедры \_\_\_\_\_

кандидат экономических наук

Майборода Евгений Викторович

«20» 01 2025 г.



Туапсе  
2025

## ОГЛАВЛЕНИЕ

Введение.....	3
1 Анализ предметной области .....	5
1.1 Обзор существующих программных решений .....	5
1.2 Моделирование процессов в нотации языка BPMN.....	13
2 Анализ объекта и предмета исследования.....	20
2.1 Описание бизнес-процессов объектов исследования.....	20
2.2 Описание предмета и обоснование выбора средств проектирования информационной системы .....	27
3 Реализация проекта.....	31
3.1 Технологическое проектирование.....	31
3.2 Организационное проектирование.....	38
3.3 Организационно-экономическое обоснование проекта.....	54
Заключение .....	58
Список литературы .....	61
Приложение .....	63

## Введение

Изучая настоящую действительность, можно отметить наметившуюся тенденцию того, что компании столкнулись с массой негативных моментов относительно вопроса перехода на полную автоматизацию деятельности, за счет которой появится потенциальная возможность улучшения качества мониторинга и минимизации временных параметров по выполнению производственных задач. По причине быстрого роста количества ресторанов заметно возрастает уровень предприятий, способных вступать в конкурентную борьбу за право ведущего ресторана того или иного города или района, что несомненно влечет за собой быстрого применения наиболее эффективной методики, которая даст в скором времени заметный результат.

В этих условиях для успешного ведения бизнеса необходимо инвестировать в средства и инструменты его поддержания и развития. Один из основных инструментов развития ресторанного бизнеса - это современная система автоматизации ресторанов. Современная система автоматизации ресторана - это профессиональная система управления рестораном, многофункциональная и легко модернизируемая[9, С.14].

На сегодняшний день для поддержания конкурентоспособности ресторанного бизнеса автоматизация деятельности заведения общественного питания стала необходимым условием. В результате автоматизации каждого процесса сокращается время на обслуживание клиентов, уменьшаются расходы, повышается точность расчетов. В тоже время поиск программных решений поставленной задачи сопряжен со сложностями, связанными с отсутствием на рынке специализированного бюджетного предложения. Поставщики программных продуктов специально усложняют заведомо программы, дабы решать возникающие проблемы текущего характера. Нельзя не исключать того, что вся эта ситуация заметно усложняет ведение бизнеса и вносит дополнительные трудности связанные с внедрением нового программного оборудования, что конечно же увеличивает стоимость покупки и как следствие

очерчивает степень важности вопроса, который ложится в основу актуальности темы выпускной квалификационной работы.

Объект исследования – процессы деятельности предприятия общественного питания африканского ресторана «Лимпопо» г.Москва.

Предмет исследований – моделирование и проектирование бизнес-процессов для автоматизированной информационной системы.

Целью работы является проектирование информационной системы для автоматизации деятельности ресторана «Лимпопо». Для достижения поставленной в выпускной квалификационной работе цели были определены следующие задачи:

- моделирование предметной области;
- анализ существующих информационных систем для автоматизации деятельности предприятий общественного питания;
- изучение основных бизнес-процессов, выполняемых в автоматизированной информационной системе;
- проектирование информационной системы;
- экономическое обоснование проекта.

Теоретико-методологической основой исследования послужили работы отечественных и зарубежных ученых в области информатики, учебные пособия, монографии, научные статьи, ресурсы сети Интернет.

Методы решения главной задачи работы – экспериментальный, методы системного анализа, моделирования, нормализации данных, теоретический.

Практическая значимость работы состоит в том, что она может быть использована при разработке информационной системы для автоматизации деятельности заведения сферы общественного питания.

## 1 Анализ предметной области

### 1.1 Обзор существующих программных решений

Учитывая заключения аналитиков в сфере ресторанного бизнеса, можно смело отметить динамику последних лет в области увеличения сложных вопросов ведения бизнеса. Ниоднократно отмечалось, что количество кафе и ресторанов резко растет, а вот сложности в данной отрасли остаются и умножаются. За последние несколько лет появившиеся информационные технологии последнего поколения показали определенную значимость во многих отраслях, данные технологические решения не обошли стороной и ресторанный бизнес, акцентируя свое внимание на всеобщее внедрение компьютерных технологий и автоматизацию бизнес-процессов [2,с.14].

Автоматизация - это применение в производстве технических средств, методов и систем управления, освобождающих человека от непосредственного участия в производственных процессах. Во всем мире оптимизация процесса продажи признана важным элементом любого процветающего бизнеса. Под системой автоматизации кафетерия предполагается программно-аппаратный комплекс управления процессами заведения. Данный инструмент предназначен для повышения эффективности управления кафетерием. В том числе для учета и контроля продаж, персонала, чтобы оптимально распределять ресурсы, минимизировать затраты, увеличить уровень сервиса, организовывать программы лояльности для посетителей и контролировать их эффективность. Именно возможности автоматизации оптимально сочетают скорость и качество.

Результат, которого ждут от автоматизации, всегда должен охватить повышение уровня производительности трудового процесса всех членов коллектива, не смотря на занимаемые должности [17,с.33].

Основными преимуществами автоматизации бизнес-процессов выделяют:

- оптимизация бизнес-процессов и снижение основных затрат;
- повышение эффективности использования программного обеспечения и оборудования;

— повышение качества производимого товара или услуги путем контроля производства;

— быстрая реакция на неполадки и сокращение времени простоев, ремонтных работ;

— оптимизация процесса взаимодействия с потребителями и поставщиками.

Именно с концептуальной модели использования информационных систем начинается эволюционное преобразование проектирования автоматизированного производства с применением компьютерных систем. Это прорыв на сегодняшний день поистине считается важнейшим шагом на пути к расширению всего информационного цикла. Заметим, что данную модель можно спроектировать за счет определенного состояния системы в определенный момент ее запуска и увидеть определенные моменты открытия, учитывая жизненные циклы проектного анализа[4,с.11].

Выделяют три класса методологии проектирования автоматизированной системы[18,с.92]:

— концептуальное моделирование предметной области;

— выявление требований и спецификация информационной системы через её макетирование;

— системная архитектура программных средств, поддерживаемая инструментальными средствами.

Чтобы детально описать объект автоматизации процесса, выделив при этом его функциональные и технические возможности, учесть полностью все нюансы технического задания проекта, изучают и на практике применяют основы методологии, что позволяет достигать гарантированно успехов в проектировании и в итоге разработать детальный план всех изменений, на которые нацелена оптимизация производственной основы.

Моделирование бизнес-процессов – начальная фаза реализации любой из информационных систем. Данный прием принято еще называть методикой повышения качества и эффективности организации, так как на первом этапе

жизненного цикла всегда определяется степень заинтересованности в получении изменений.

Сегодня множество инструментов, с помощью которых можно начинать моделирование с нуля. Не исключение из правил и нормативно-законодательная база, с помощью статей и документов которой решаются многие вопросы относительно автоматизации производства и которые полностью регламентируют те или иные изменения.

Системы обработки информации используют определенные алгоритмы для обработки информации, влияя на процесс принятия решений до тех пор, пока не будут получены результаты.

Обзор и детальное изучение всех программных систем, которыми заполнили рынок дает полную картину происходящего, а именно: на российском рынке активно многие применяют как отечественного производства разработки, так и зарубежного, хотя, в последнее время наложены жесткие санкции на многие корпоративные программы, но это не умаляет степени их востребованности и их число только растет.

Наиболее известны следующие системы: «R-Keeper», «Iiko», «Тракторь», «АСУУ:Рестораторь», «Галактика» и др. В основном большая часть информационных систем, присутствующая на российском рынке, ориентирована на предприятия малого и среднего бизнеса и рассчитана для торговых предприятий, на автоматизацию лишь некоторых функций предприятия [13,с.88].

Система представляет собой комплекс технических и программных мероприятий. Данное средство направлено в первую очередь на усиления эффектов сферы питания. На рынок был представлен ряд практических продуктов, предназначенных для улучшения ресторанного бизнеса.

«R-Keeper» - наиболее актуальная и самая, пожалуй, распространенная среди рестораторов программа, начавшая свою работу на рынке ресторанного бизнеса с 1992 года. Аппаратно-программный комплекс включает в себя огромное количество модульных элементов, нацеленных на работу сети

общепита и работающего в системе клиент-сервер (рисунок 1.1)

«R-Keeper» предоставляет доступ к инструментам управления заведением, складом и производством, а также инновационные технологические решения по организации эффективной работы персонала и управлению лояльностью гостей [31].



Рисунок 1.1 - «R-Keeper» - программно-аппаратный комплекс

Для автоматизации складского учета в системе «R-Кеерер» используется программа «StoreHouse», которая позволяет контролировать процесс управления. Программа «StoreHouse» имеет простой и удобный интерфейс, который позволяет быстро освоить основные функции программы (рисунок 1.2).

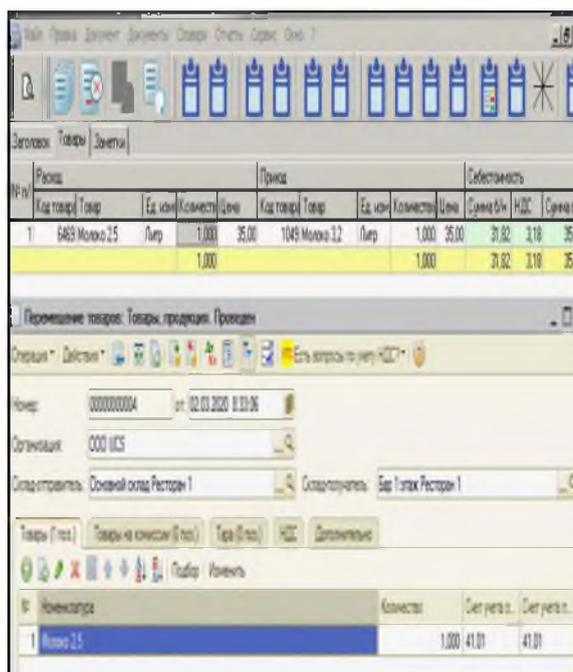


Рисунок 1.2 – Диалоговое окно системы «StoreHouse»

Изучая технические характеристики такой программы как iiko, можно отметить, что она обладает определенным набором качеств, делающие ее наиболее известной и востребованной в сфере питания. Программная платформа зарекомендовала себя известностью за счет ведения складского учета, определения многих параметров в области менеджмента и расчетов экономического характера, а так же тесного сотрудничества в сфере налоговых платежей, а это очень удобно в работе многих предприятий.

Одним из главных недостатков системы «R-keeper», помимо высокой цены, является то, что фронт-офис, выполняющий кассовые задачи имеет собственную базу, а «StoreHouse» имеет совершенно другую базу. Эти базы не имеют взаимосвязи в режиме реального времени друг с другом. Все данные необходимо перемещать из одной в другую специальным платным модулем.

В последние годы активно внедряется в сферу общественного питания программа «iiko». Программа включает в себя учет склада, финансов и управления, поддерживает работу с фискальными регистраторами разных моделей и производителей.

Регистрация заказов на закупку, отправка их поставщикам в электронном виде, отслеживание статусов их исполнения, а также получение в электронном виде приходных накладных (рисунок 1.3).

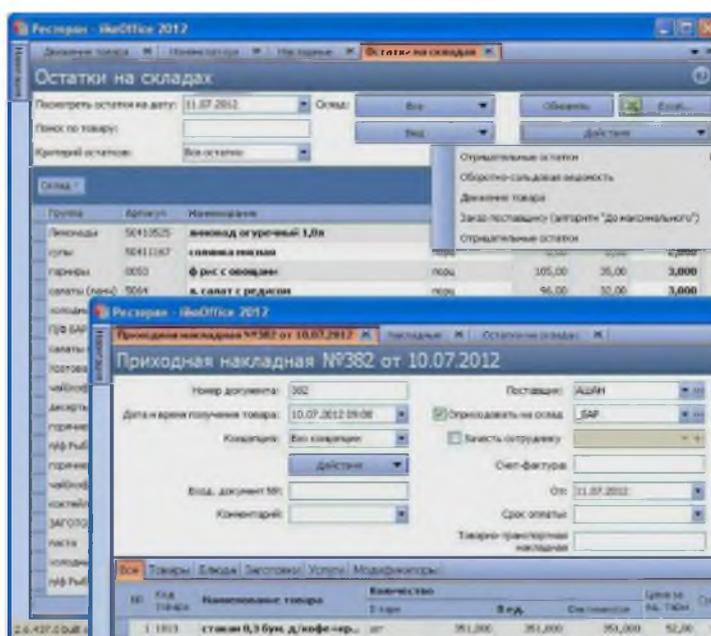


Рисунок 1.3 - Интегрированная система управления «iiko»

Выделим, что взаимоотношения со многими поставщиками непосредственно формируются за счет составления накладной документации и счетов-фактур, исключая, при этом рутинные потоки устаревшего ручного ввода информации и ведения процесса документирования.

Автоматизированно происходит принятие всех позиций на складской учет в оптимально короткие сроки с учетом полного оформления всей соответствующей документации по поступившему товару. Заметим, что в программу вложена функция полного перерасчета остатков товаров на момент экспертизы за счет учета их стоимостной ценности на конечную дату.

Определенное удобство создается программой по встроенному прайс-листу, что позволяет планировать и прогнозировать определенные сегменты товарной группы для определенных поставщиков.

Сама система имеет так же функционал расчета отклонения от нормативного значения цены конечной на товары и себестоимости, определения сегментов товарных групп, которые устарели и не пользуются спросом, а так же наиболее востребованные товарные позиции, что позволяет задавать стиль поведения с поставщиками и исключать конфликтных ситуаций без предварительных предупреждений.

Если продолжать применять платформу 1С для автоматизации предприятия, то наилучшим решением так же может стать активно используемая программа ТрактирЪ, которая входит в топ лучших за последние 5 лет существования на рынке ресторанного бизнеса.

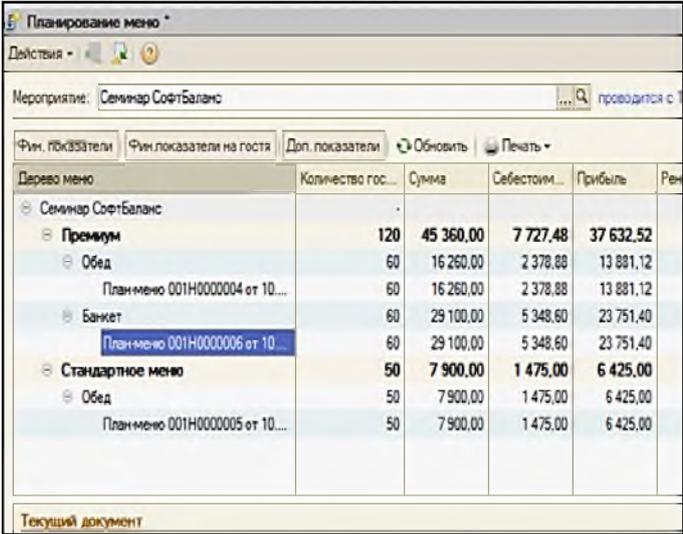
Модульная программа, способствует не только расширению спектра функциональных возможностей программного оборудования, но и выделяет новый инструментарий для осуществления полной автоматизации бизнес-процессов ресторана или кафе.

Работая с данной программой на платформе 1С: Предприятие 8.3 появляется возможность ведения бизнеса с любого аппаратного устройства, причем без привязки с географической точки зрения. Удобный интерфейс лаконично и гибко выстраивает возможность для создания пользователям

своего индивидуального рабочего пространства, что заметно повышает уровень разрешения экрана.

Программа «Трактирь» - продуктовая линейка для автоматизации заведений общественного питания на платформе «1С». В разработке каждой версии заложены современные подходы решения задач и технические средства для автоматизации бизнеса.

«Трактирь: Management» - бэк-офисная система, предназначенная для работы с поступлениями, реализацией, калькуляциями, выпусками продукции, планированием меню, контролем взаиморасчетов с поставщиками и другими операциями, связанными с ведением управленческого учета в общественном питании (рисунок 1.4).



The screenshot displays a web-based interface for menu planning. At the top, the title is 'Планирование меню'. Below it, the event name 'Семинар СофтБаланс' is entered. The interface includes several tabs: 'Фин. показатели', 'Фин. показатели на гостя', 'Доп. показатели', 'Обновить', and 'Печать'. The main area is a table with columns: 'Дерево меню', 'Количество гос...', 'Сумма', 'Себестоим...', 'Прибыль', and 'Рен'. The table data is as follows:

Дерево меню	Количество гос...	Сумма	Себестоим...	Прибыль	Рен
Семинар СофтБаланс	-	-	-	-	-
Премиум	120	45 360,00	7 727,48	37 632,52	
Обед	60	16 260,00	2 378,88	13 881,12	
План меню 001Н0000004 от 10...	60	16 260,00	2 378,88	13 881,12	
Банкет	60	29 100,00	5 348,60	23 751,40	
План меню 001Н0000006 от 10...	60	29 100,00	5 348,60	23 751,40	
Стандартное меню	50	7 900,00	1 475,00	6 425,00	
Обед	50	7 900,00	1 475,00	6 425,00	
План меню 001Н0000005 от 10...	50	7 900,00	1 475,00	6 425,00	

Рисунок 1.4 - Бэк-офисная система «Трактирь: Management»

«Трактирь: Management» построен на платформе «1С: Предприятие 8.3». Работа с решением ведется через браузер из любой точки мира, в том числе и с мобильных устройств. Всегда актуальная выгрузка данных в «1С: Бухгалтерия 8 ПРОФ» позволяет быстро подготовить регламентированную отчетность, в том числе и в режиме аутсорсинга бухгалтерских услуг, с выгрузкой только необходимых документов. Концепция нового пользовательского интерфейса позволяет пользователю самостоятельно конструировать рабочее пространство программы, что обеспечит комфортную работу на разных разрешениях экрана.

Если рассматривать далее программы, которые являются более экономичными и бюджетными, то стоит остановить свой выбор на системе «АСУУ:Рестораторь»(рисунки 1.5).



Рисунок 1.5 - Программное средство - «АСУУ:Рестораторь»

Многие видят в применении программ что-то сложное, затратное и громоздкое с точки зрения технических возможностей, но рассматривая применение «АСУ:Ресторатор» в деятельности своего ресторана его можно считать универсальным способом со всех сторон: прост в использовании, бюджетен в точки зрения приобретения и обслуживания и эргономичен в применении.

Нельзя считать полностью универсальным свойство изначально при проектировании степень доступности многих модулей программного продукта. Не исключением становятся корпоративные информационные системы последнего поколения, которые отличаются своей масштабностью и нацелены в первую очередь на полную автоматизацию сложных процессов, оптимизации бизнес-деятельности компаний и унификации корпоративного управления.

«QuickResto» - облачная система автоматизации ресторанного бизнеса, которая позволяет управлять меню, рассадкой гостей, принимать оплату и настраивать принтеры для печати чеков (рисунки 1.6).

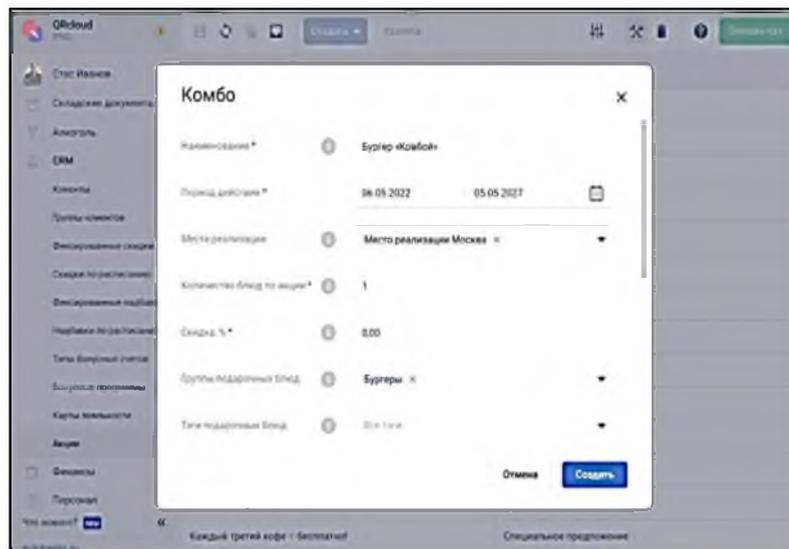


Рисунок 1.6 - «QuickResto» - облачная система

Имеющиеся программные средства носят универсальный характер. Корпоративная Информационная Система (КИС) - это масштабируемая система, предназначенная для комплексной автоматизации всех видов хозяйственной деятельности компаний, а также корпораций, требующих единого управления.

Современные ИС состоят из Front-офисных и Back-офисных подсистем. Front-офисные подсистемы предназначены для оптимизации обслуживания клиента, Back-офисные подсистемы автоматизируют менеджмент предприятия, считают расходы и доходы.

Автоматизация системы ресторанного бизнеса является фактором конкуренции на рынке. Качество — вот показатель ресторана, визуальный образ о цене и виде самого заведения, поэтому мы можем обеспечить обслуживание клиентов и подробную информацию о ресторане, меню, времени и т.д. непосредственно за счет информационной направленности[3,С.12].

## 1.2 Моделирование процессов в нотации языка BPMN

На протяжении длительного периода времени именно информационная система используется для автоматизации деятельности в сфере ресторанного

бизнеса[6,с.105].

За счет нее появляется возможность оптимизировать связность продуктов и услуг поставщиков и, в некоторых случаях получать доступ к информации, которая находится в распоряжении самого заведения. Начнем рассмотрение всей концепции автоматизации с понятий моделирования бизнес-процессов и структуры организации.

Моделирование бизнес-процессов - это отражение субъективного видения реально существующих в организации процессов при помощи графических, табличных, текстовых способов представления. Один из самых распространенных типов нотаций (синтаксиса графического моделирования) для создания исполняемых процессных моделей является BPMN (BusinessProcessModelandNotation)[20,с.38].

Моделирование бизнес-процессов в нотации BPMN используется для донесения широкого спектра информации до различных категорий пользователей. Конечный результат BPMN составляет диаграмма бизнес-процесса, отображающая поток работ, основанный на стандартах графической нотации. Для моделирования процессов, протекающих на каждом этапе, воспользуемся нотацией BPMN 2.0.

В данной выпускной квалификационной работе для моделирования процессов применены основные диаграммы: диаграмма оркестровки. Схемы взаимодействия (Collaboration) показывают обмен сообщениями между двумя и более участниками.

Диаграмма взаимодействия - используется для отображения состава и последовательности выполнения двух и более процессов в виде пулов с указанием взаимодействий между ними через потоки сообщений. Диаграмма оркестровки - это диаграмма, показывающая последовательность и логику выполнения заданий в рамках одного процесса[25,с.83].

Так же на диаграмме показываются взаимодействия между частным, рассматриваемым процессом и другими процессами или участниками, отображенными в виде свернутых пулов.

Для проектируемой системы процесс взаимодействия предприятия общественного питания и поставщика представлен на рисунке 1.7.

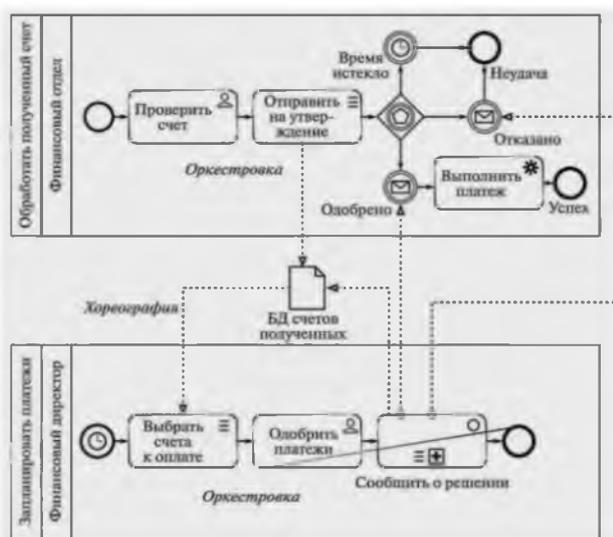


Рисунок 1.7 - Детализированная диаграмма межпроцессного взаимодействия «Процесс обработки заказа на поставку продукции»

Диаграмма отображает процесс обработки заказа на поставку продукции, изображенный в виде 2 пулов (зоны ответственности), и связывающие их потоки сообщений. В каждом пуле подробно и последовательно указаны действия каждого из участников. В верхнем пуле описаны действия поставщика. В пуле, описывающем действия самого ресторана, содержатся две дорожки, отображающие действия администратора и бухгалтерии. Чтение процесса всегда начинается со стартового события (зеленого кружка). После события - начала работы следует формирование заказа (операция, указанная в прямоугольнике)[21,с.97].

При создании позиции элементов меню идет обращение к хранилищу «БД Меню». Заказы на необходимую продукцию поступают через потоки сообщений к поставщику. Сообщение (Message) представляет собой содержимое диалога между двумя Участниками (Participants). Выделяют входящие и исходящие типы сообщений. Так если сообщение идет от инициатора, то на диаграмме это соответствующе отображается значком с закрашенным письмом.

Текущие заявки, которые поступают, видит любой провайдер в режиме настоящего временного пространства, что предусмотрено самой системой при проектировании. Когда поставщик получает запрос на товар, он начинает оформлять заявку на ее исполнение. Открыв систему предложений, составляется коммерческое предложение, за счет которого определяются будущие деловые отношения между организацией и коммерческим предприятием.

Следующий шаг предполагает рассмотрение и/или подтверждение /отклонение поступившего заказа. Когда ответ получен, задачей системного администратора становится полная проверка по всем параметрам поступившего ответа.

На схеме представлено взаимодействие одного шлюза – системного оператора и второго шлюза – немонопольного в процессе обмена данными и формирования потоков управления несколькими вариантами выполнения условий.

Получив ответ от поставщика, администратор в системе проверяет полученное предложение на соответствие отправленной заявке. На диаграмме это отображено в качестве логического оператора (шлюз). Не эксклюзивный шлюз (OR, «ИЛИ») используется для ветвления потока управления на несколько потоков, когда выполнение процесса зависит от выполнения условий. Шлюз изображается в виде ромба, внутри которого находится рисунок, отображающий тип шлюза. Так, для не эксклюзивного шлюза внутри ромба изображается (X)[24, с.55].

Таким образом, в случае если предложение соответствует заказу, то происходит подтверждение на доставку продукции, в противном случае наступает снова этап формирования списка элементов. Далее после получения накладной, которую формирует поставщик при подтверждении заказа администратором, бухгалтерия оплачивает продукцию. В системе происходит подтверждение оплаты полученной накладной и после этого обновление информации обо всех финансовых изменениях в хранилище «БД Финансы», в

которой отображается информация о денежных потоках. Перед отправлением продукции поставщик должен подтвердить получение платежа, что показано с помощью не эксклюзивного шлюза. В противном случае, снова отправляется накладная в бухгалтерию.

Последнее финальное событие в сфере работы с заявкой происходит в тот момент, когда получен ни отказ, а согласие на поставку, что определяет круговое движение на схеме, начерченный красной жирной линией[12,с.28].

Формируется хранилище базы данных в виде концептуальной структуры в номинации BPMN. Основу составляет именно процесс моделирования всех сведений в единую базу данных.

Хранилища представляют собой информационные модели, позволяющие упорядоченно хранить данные о группах объектов, обладающим одинаковым набором свойств. В «БД Меню» собрана информация по таким признакам как, наименование продукции, её количество, цена. Использование хранилищ помогает производить выборку с произвольным сочетанием признаков.

Направление стрелки, соединяющей базу данных с операцией, имеет определенное значение. От хранилища к операции - операция использует базу для получения данных. От операции к хранилищу - операция помещает данные в базу. Создавая заявку на заказ продукции, администратор, обращаясь к хранилищу «БД Меню», получает информацию о том, какую продукцию необходимо добавить в заявку.

Далее рассмотрим процесс обработки заказа клиента. Это центральный и наиболее важный процесс. Диаграмма оркестровки для данного процесса представлена схематично. Участниками этого процесса выступают официант и клиент, что показано на рисунке 1.8 при помощи двух пулов «Клиент» и «Ресторан».

Учитывая структуру рабочего процесса, выделим, что пользователь с правами «официант» или «клиент» получают возможность самостоятельного обращения к системе для последовательного формирования и в конечном этапе завершения заказа.

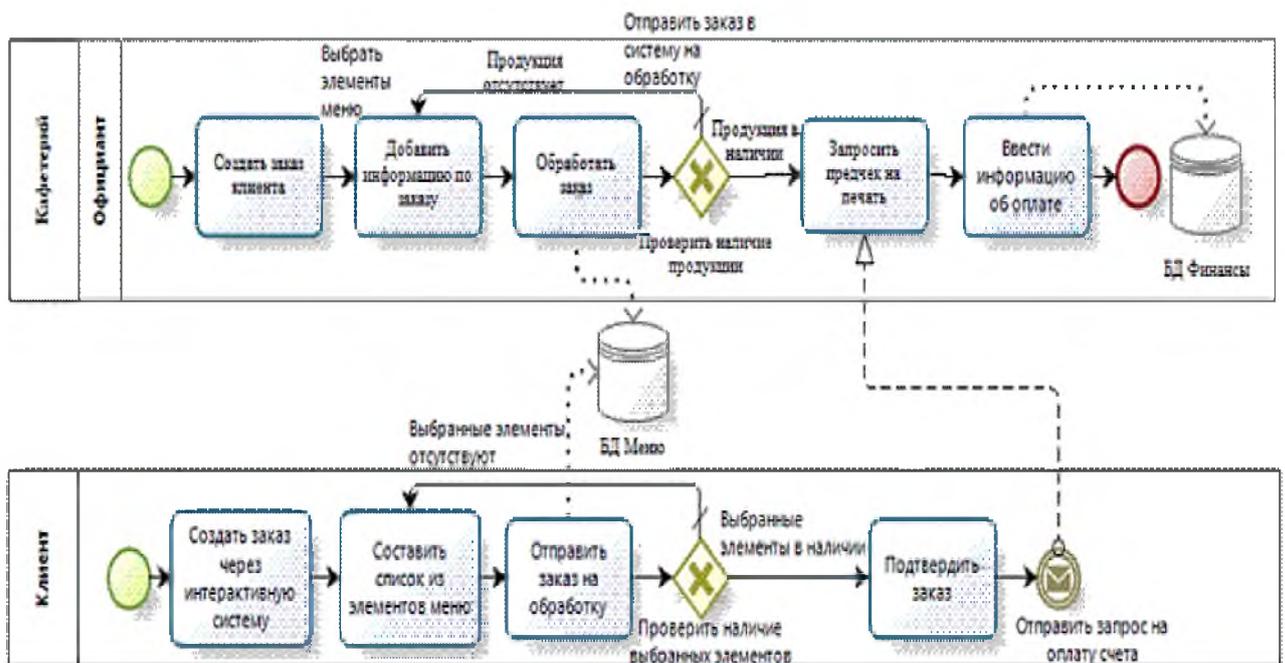


Рисунок 1.8 - Диаграмма оркестровки «Процесс обработки заказа клиента»

За счет возможности потоковой передачи сведений и базе товаров или услуг, что находится и формирует базу данных ресторана или кафе формируется заказ и передается ответственному за исполнение – будь это повар или бармен. Если по определенным причинам выбранной позиции нет в наличии, система предлагает вернуться назад и произвести выбор, исходя из предложенного списка самому заказчику. В случае конкретного состояния расчет стоимости технического обслуживания и оплата оказанной услуги производится обслуживающим лицом.

После запроса официантом предчека на печать, информация об оплате заказа клиента поступает в хранилище «БД Финансы». Производится закрытие заказа. На основании построенных двух моделей, которые описывают основные процессы в кафетерии, можно осуществить проектирование информационной системы для автоматизации деятельности заведения сферы общественного питания. Для этого необходимо определить главные роли, которые будут участвовать в процессе взаимодействия в автоматизированной системе. Для однозначности понимания, описать функциональный набор каждого из участников. Чтобы грамотно и детально осуществить изменения в структуре

информационного наполнения в первую очередь необходимо поменять действующие роли исполнителям, которые будут занимать важное место в разрабатываемой системе. Для более прозрачного видения распишем каждый элемент действий по функциональным признакам настройки с учетом каждого участника программного модуля.

Чаще всего для визуализации, подробного описания моделей используют графический язык UML. Этот язык обеспечивает поддержку всех этапов жизненного цикла информационной системы и содержит множество графических средств - диаграмм. Так на этапе создания логической модели, для определения функционально набора для каждого участника используют диаграммы прецедентов, диаграммы классов обычно используются на этапе создания физической модели проектирования [24,с.108].

## 2 Анализ объекта и предмета исследования

### 2.1 Описание бизнес-процессов объектов исследования

Ресторан «Лимпопо» был открыт в конце 2010 года в городе Москве. Ресторан расположен во встроенно-пристроенном помещении, пристройке к жилому дому. Помимо основного меню предлагается бизнес-ланч с 12.00 до 16.00 и постное меню во время Поста. Удачное месторасположение, качество предлагаемых блюд и приемлемые цены привлекают в ресторан гостей, количество которых, несмотря на существующую конкуренцию, с каждым днем растет, благодаря хорошему сервису (рисунок 2.1).



Рисунок 2.1 – Центральный вход в ресторан «Лимпопо»

Стиль ресторана «Лимпопо» соблюден в африканском направлении. Половина стен расписана морилками, другая половина картинами из жизни эфиопов, фигурки из черного дерева, стилизованные настенными светильниками, деревянные скамейки и диваны, а также столами на 4 человека. В меню экзотические блюда: мясо аллигатора, обжаренное в нежном белом блинчике, кенгуру с овощами и ликером, фаршированная акула (рисунок 2.2).

Любителям простой русской кухни предлагаются русские супы, щи, рыба домашнего посола и пельмени. Звучит в качестве фоновой – музыка народов африки. С целью максимизации картотеки постоянных клиентов руководство ресторана активно применяет систему дисконтирования с учетом скидок на 5%

и 10%, а так же VIP-карты для особых гостей ресторана.



Рисунок 2.2 – Интерьер ресторана «Лимпопо»

Источником стабильности и высокого уровня конкурентных позиций на рынке ресторанного бизнеса можно считать взаимодействие с надежными поставщиками сырья и полуфабрикатов, а также готовой продукции (сигары, сигареты, алкогольные напитки, вода и т.д).

Следует учесть, что все оборудование, которым оснащен ресторан «Лимпопо» представлено современными позициями, а приобретенные с его помощью технологии обеспечивают высокое качество продукции и возможность конкурировать с другими ресторанами района.

Согласно изученным нормативным документам, ресторан «Лимпопо» является обществом с ограниченной ответственностью, право лица, владеющего имуществом, которое учитывается в его нематериальном балансе, может действовать от своего имени и устанавливать имущественные и личные неимущественные права, а так же прслеживается определенная степень ответственности. Основным документом ресторана является Устав. Собственность - частная.

Организационная структура ресторана «Лимпопо» имеет линейно-функциональный вид, распространенный среди предприятий малого и среднего бизнеса. Организационная структура представлена на рисунке 2.3.

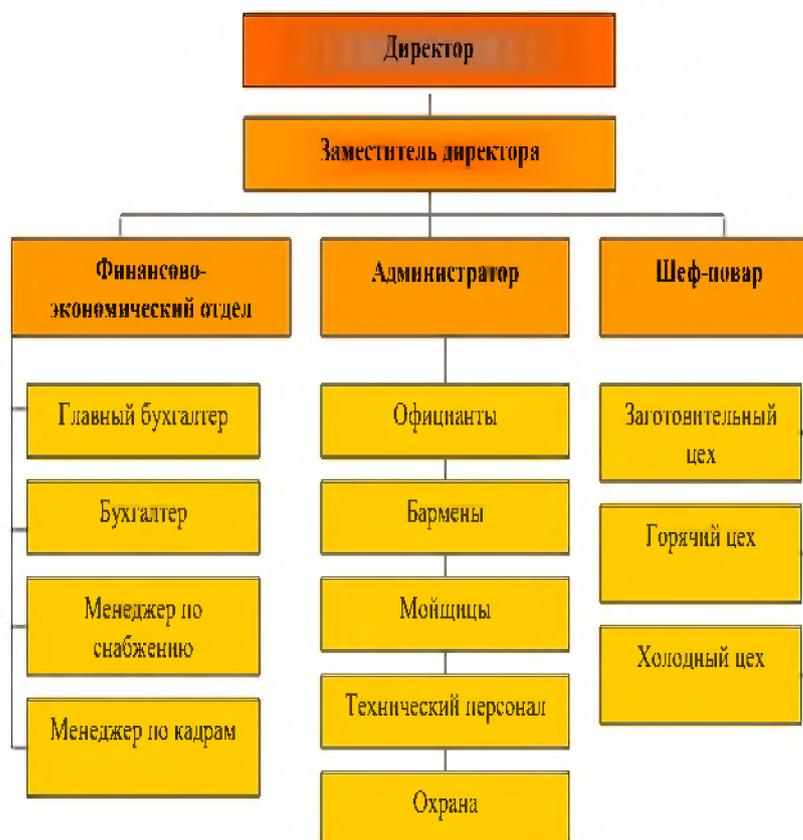


Рисунок 2.3 –Организационная структура ресторана «Лимпопо»

Линейная структура управления, являясь классической на протяжении многих десятков лет, упрощает функциональное подчинение всех сотрудников предприятия, а руководителю помогает выстраивать четкое подчинение всех участников сплоченной команды.

Анализ системы управления рестораном позволил изучить штатные единицы и выделил заместителя директора, на которого так же ложится определенная степень административного управления, помимо которой, он обязан контролировать деятельность снабжения и следить за качеством нитолько готовых блюд, но и обслуживания в целом.

Заместитель имеет те же права, что и директор, он отвечает за производство в отсутствие самого руководителя.

Шеф-повар является заведующим производством и осуществляет контроль за соблюдением рецептур блюд, технологии их приготовления, проверкой готовой продукции, разработкой рецептур новых блюд. Он

обеспечивает соблюдение правил санитарии и личной гигиены, охраны труда и техники безопасности на кухне.

Администратор контролирует работу официантов, барменов, уборщиков, обеспечивает соблюдение правил обслуживания ресторана, внутреннего распорядка, ношения форменной одежды и т.д.

Показатели финансовых результатов характеризуются снижением нераспределенной прибыли и потерей рентабельности, что всегда является свободой активов, а это основа финансовой стабильности.

Основным источником информации при анализе прибыли является Отчет о финансовых результатах (таблица 2.1). Как видно из анализа, что предприятие в течение трех лет заметно улучшило свое финансовое положение, так как чистая прибыль по итогам 2023 года заметно увеличилась. Самая высокая прибыль в балансе за 2022-2023 гг. за чет получения выручки от продаж. Увеличение прибыли связано с увеличением количества посетителей, что, несомненно, свидетельствует о стабильной репутации самого ресторана. Данный факт так же можно прокомментировать введением гибкой системы скидок и применением карт лояльности для постоянных клиентов.

Таблица 2.1 - Анализ финансовых результатов деятельности ресторана «Лимпопо», тыс.руб.

Показатели	2022 год	2023 год	Абсолютное отклонение		Темп роста, %	
			2023 к 2022 гг	2022 к 2021гг	2023 к 2022гг	2022 к 2021гг
Валовая прибыль	38351	69518	31167	6618	181,27	120,86
Прибыль от реализации	7848	24620	16772	2634	313,71	150,52
Прочие доходы и расходы	194	867	673	-2345	446,91	7,64
Прочие доходы	232	970	738	-2622	418,10	8,13
Прочие расходы	38	103	65	-277	271,05	12,06
Общая сумма прибыли до налогообложения	8042	25487	17445	289	316,92	103,73
Чистая прибыль	6112	19370	13258	220	316,92	103,73

Наибольшую долю в балансовой прибыли на протяжении 2022-2023 гг. имеет прибыль от реализации. На увеличение прибыли сказалось увеличение объема посетителей, что, несомненно, указывает на улучшение работы самого ресторана.

Метод DFD (данные FlowDiagramming) используется для записи рабочих процессов по восстановлению. Эта заметка применима так же для фиксации даты, что заметно облегчает отсрочку доступности работы, выполняемой в ходе процесса, и поток информации относительно периода работы сотрудников. Кроме того, DFD допускает движение документов и движение материалов. Диаграмма процесса в DFD определяет основные потоки данных (рисунок 2.4).

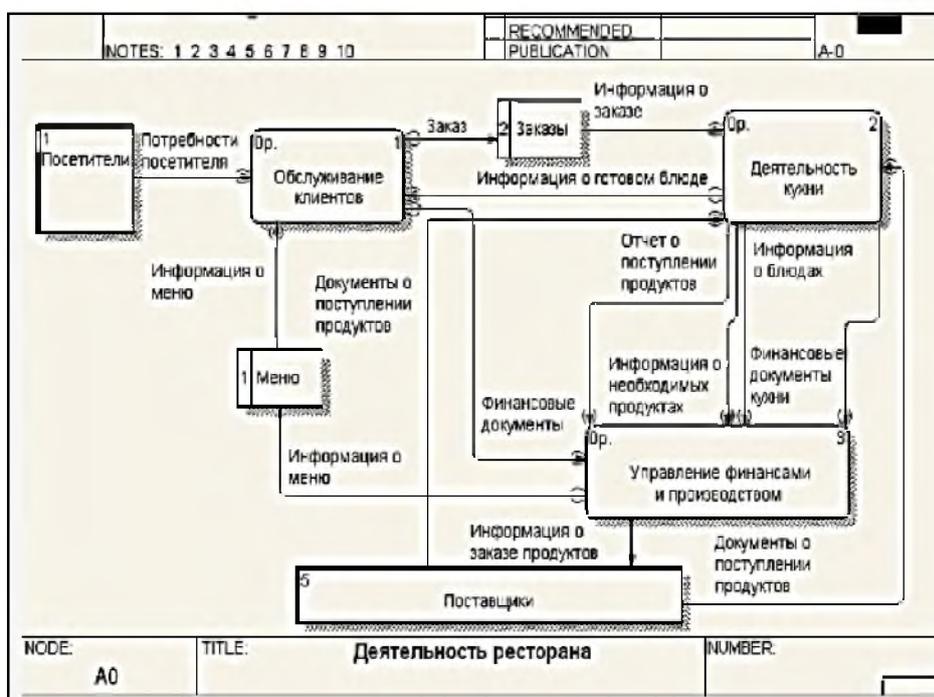


Рисунок 2.4 - Декомпозиция диаграммы IDEF0 по функциям

На рисунке 2.5 представлен каждый отдельно взятый элемент проектируемой системы, который представлен в виде декомпозиционной структуры. Диаграмма декомпозиции предназначена для детализации работы. В отличие от моделей, отображающих структуру организации, работа на диаграмме верхнего уровня в IDEF0 - это не элемент управления нижестоящими работами. Работы нижнего уровня - это то же самое, что работа верхнего уровня, но в более детальном изложении [20,с.33].

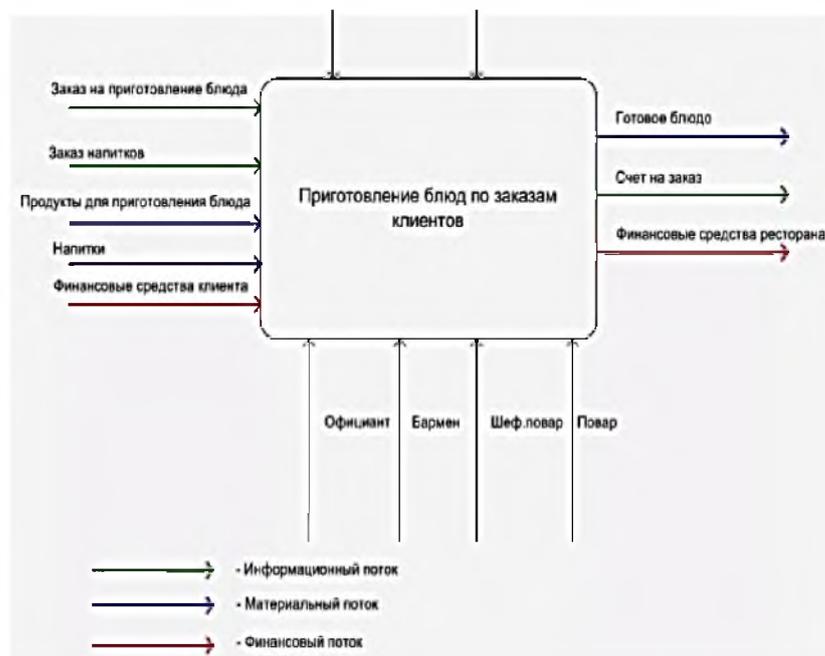


Рисунок 2.5 - Схема информационных потоков в ресторане

Весь процесс функционирования системы ресторана разбивается на следующие потоки:

Оформление заказов

Изготовление заказа

Предоставление готовых блюд

Система управления базой данных - это комплекс программных средств, который предназначен для создания структуры новой базы, редактирования содержимого и визуализации информации, т. е. отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройство вывода или передача по каналам связи.

Access - это реляционная система управления базами данных (СУБД), входящая в пакет MS Office. Все составляющие базы данных, такие как таблицы, отчеты, запросы, формы и объекты, в Access хранятся в едином дисковом файле, который имеет расширение. Mdb [16,с.8].

Создание схемы данных позволяет упростить конструирование таблиц, форм, запросов, отчетов, а также обеспечить поддержание целостности взаимосвязанных данных при корректировке таблиц. Схема данных наглядно

отображает таблицы и связи между ними, а также обеспечивает использование связей при обработке данных и целостность БД (рисунок 2.6).

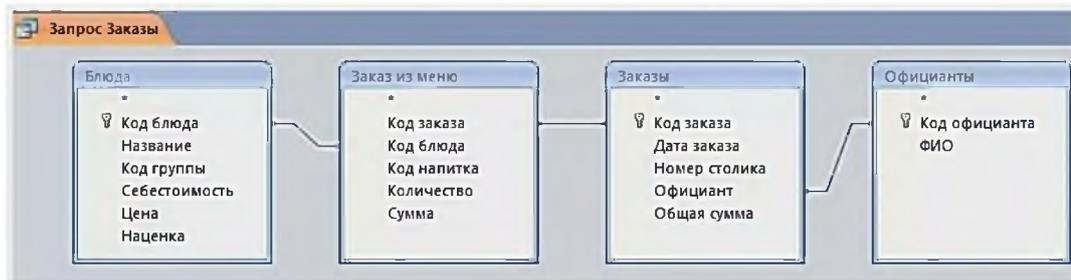


Рисунок 2.6 - Схема данных ресторана

Схема данных задает структуру БД, которая является графическим образом. Схема данных базы графически отображается в своем окне, таблицы представлены списками полей, а связи - линиями между полями разных таблиц. Схема данных ориентирована на работу с таблицами, отвечающими требованиям нормализации, между которыми установлены связи 1: М и 1: 1 с обеспечением целостности БД. Поэтому схема данных строится в соответствии с информационно-логической моделью.

База данных относится к структуре, которая представляет собой графическое описание всех имеющихся элементов. Схема базы данных отображается графически и прослеживается в окнах или таблицах, которые взаимосвязаны между собой логически.

При этом возникает вопрос о выборе подходящего программного обеспечения. Есть универсальные варианты, созданные на рынке программного обеспечения, а также варианты, которые будут запрограммированы для конкретных областей, где не будет избыточности функций, сложных интерфейсов и прочего. Ресторан «Лимпопо» имеет собственную внутреннюю систему, которая никак не соприкасается с уже работающим на полную мощность сайтом по доставке готовой еды и напрямую должен интегрировать с внутренней системой самого ресторана.

Данный сайт является адаптивным и просматриваемым на экране любого разрешения, т.е. он представлен на различных мобильных устройствах ПК,

планшетах, смартфонах. На последних этапах разработки сайта он интегрируется по протоколу SOAP с внутренней системой, но данная интеграция не выполняет следующие задачи:

- передача заказов с сайта в систему;
- передача контактной информации клиентов с сайта в систему;
- скачивание и обновление справочников с системы на сайт (города, улицы и пр.).

Все ранее изученные информационные версии не отвечают основным требованиям, предъявляемым в данной предметной области. Разработанные программные продукты имеют и другие недостатки, значительную избыточность функций, нет незагружаемого интерфейса, нет функции перемещения заказов на доставку.

В связи с отсутствием определенного функционала было принято решение спроектировать информационную систему, которая будет отвечать всем параметрам в сфере ресторанного бизнеса.

## 2.2 Описание предмета и обоснование выбора средств проектирования информационной системы

Разработка информационной модели в первую очередь начинается с рассмотрения бизнес-процессов и разработки их модели с помощью SADT методологии. SADT методология позволяет представить процессы системы средней сложности, протекающие в организации, в графическом виде. Она наглядно показывает блоки процессов, их взаимосвязи, входные и выходные данные[26,с.138].

Методология IDEF0 позволяет описывать широкий круг систем с учетом использования анализа системы, например, для проведения реинжиниринга. Она позволяет практически полностью устранять неоднозначности возникающие в процессе описания. Данная методология была выбрана из-за легкости восприятия построенных с помощью нее диаграмм.

Рассматриваемый процесс - это процесс доставки еды в нотации IDEF0 (рисунок 2.7).

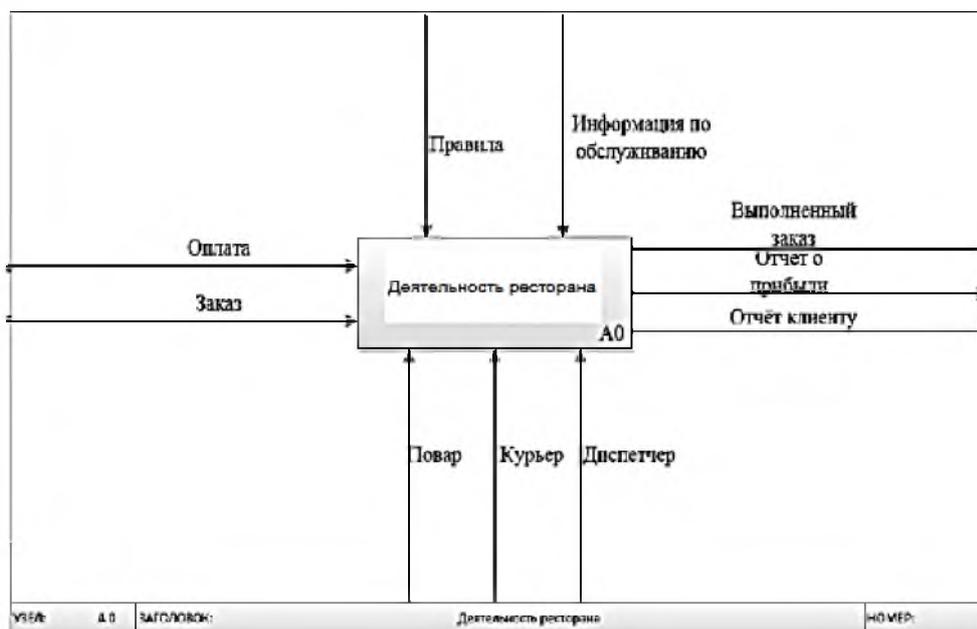


Рисунок 2.7 - Контекстная диаграмма «Деятельность ресторана»

В каждом бизнес-процессе есть входная и выходная информация.

Входная информация - это вся информация, необходимая для выполнения задачи. Она может быть представлена в различном виде: в памяти персонального компьютера, на первичных документах, машинных носителях.

Выходная информация и есть цель обработки планово-учетных данных, и представляет собой сведения, полученные в процессе решения задач, и используется работниками для планирования, учёта, контроля, анализа, принятия решений. Для данного бизнес-процесса входной информацией являются заказ клиента и оплата, то есть деньги клиента.

В итоге выполнения процесса мы получаем выходную информацию, а именно выполненный заказ, отчёт о полученной от этого прибыли и счет, который получает клиент. Для того чтобы выполнить этот процесс необходимы сотрудники ресторана, офисное оборудование и программное обеспечение.

По диаграмме учета заказов выясним, какую информацию о клиенте заносят в случае, если этот клиент заказывает в первый раз. Так класс «Официант» вызывает через интерфейс «Координатор клиента» функции

создания и просмотра класса «Заказ клиента», которые реализуются в «Личном кабинете клиента» (рисунок 2.8).

Один интерфейс может вызывать несколько классов. Так классы «Администратор», «Директор», «Официант» и «Клиент» обращаются к интерфейсу «Координатор «БД Меню»» для просмотра класса «БД Меню».

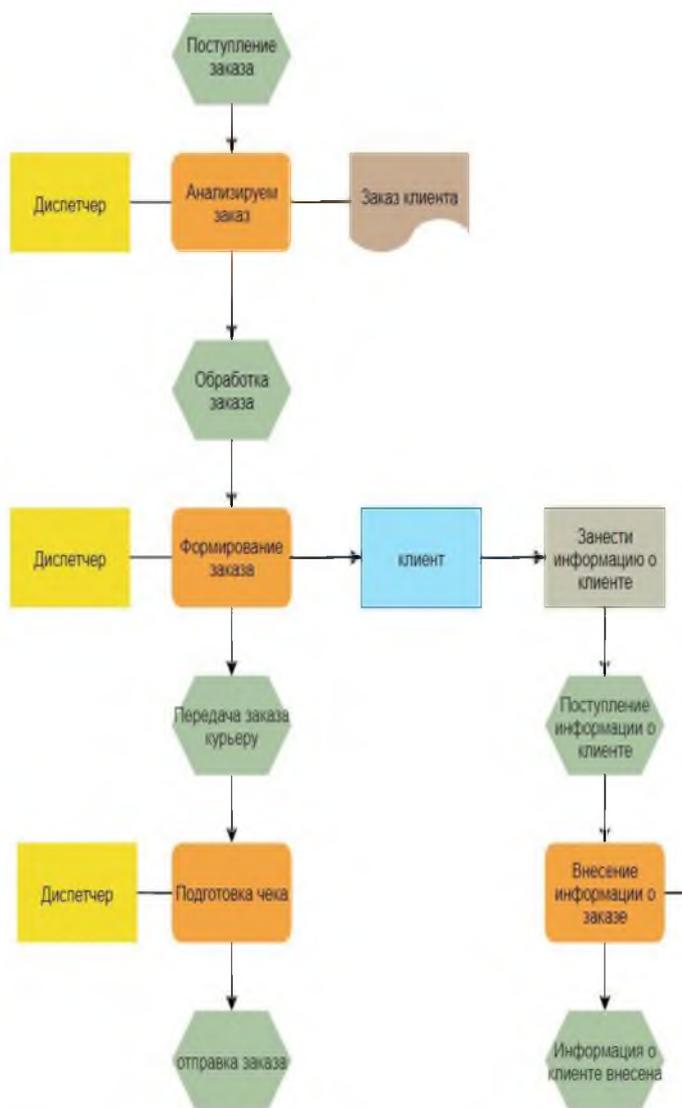


Рисунок 2.8 - Диаграмма учета заказов

При построении диаграммы классов для изображения интерфейсов используется специальный графический символ - прямоугольник класса с ключевым словом или стереотипом «interface». При этом секция атрибутов у прямоугольника отсутствует, а указывается только секция операций [28,с.64].

Между интерфейсами и классами применяются два отношения:

— класс использует интерфейс - это показывается с помощью зависимости со стереотипом «call»;

— класс реализует интерфейс - это показывается с помощью отношения реализации «realize».

Диаграммы последовательностей отражают поток событий, происходящих в рамках варианта использования. Преимущество диаграммы последовательностей заключается в простоте отслеживания сообщений, передаваемых последовательно. В диаграмме последовательностей основные сущности изображаются горизонтально вдоль верхней границы. Из пиктограммы каждого объекта выходит вертикальная пунктирная линия - «линия жизни».

Сообщения (которые часто означают события или инициирование операции) изображается в виде горизонтальных линий.

Порядок сообщений указывается в вертикальном направлении, что позволяет не использовать нумерацию.

Тип сообщения отображается с помощью разнообразных линий и стрелок. Синхронное сообщение, представляющее собой вызов операции, изображается в виде сплошной линии с закрытой стрелкой. Передача управления обратно участнику, который до этого инициировал сообщение, происходит с помощью ответа на сообщение, которое изображается штриховой линией с открытой стрелкой [23,с.81].

### 3 Реализация проекта

#### 3.1 Технологическое проектирование

Непосредственно в первую очередь необходимо остановиться на составе и структуре основных вспомогательных частей самой системы автоматического управления.

Система автоматического управления рестораном «Лимпопо» предусматривает реализацию следующих функций: формирование заказа от клиента, выбор блюд в основном меню, создание справочников, а в конце работа с отчетами. Все это принципиально точно показано на рисунке 3.1.



Рисунок 3.1 – Структура АСУ ресторана «Лимпопо»

Изучив детально структуру ресторана, выделим функционал, который необходимо оптимизировать с помощью разрабатываемой автоматизированной системы управления:

- регистрация посетителя с указанием первичной информации;
- создание заказа с непосредственным бронированием столика;
- заполнение справочников клиентов, блюд, столиков, организаций, персонала;
- построение отчетной документации

Реализация функции «Меню» предусматривает подключение и отключение, а так же выход из системы.

Реализация функции «Работа с заказами» предусматривает:

- регистрацию клиента;
- создание заказа / редактирование заказа;
- добавление позиций в заказ;
- просмотр списка заказов;
- печать чека / возможность предварительного просмотра.

Реализация функции «Отчеты» предусматривает построение аналитических отчетов по заданным критериям (дата, время, организация, блюда, персонал).

Реализация функции «Справочники» предусматривает добавление, редактирование, удаление первичной информации, необходимой для работы АСУ «Ресторан «Лимпопо»».

Реализация функции «Справка» предусматривает вывод списка изменений в текущей версии системы.

Учитывая те ниши бизнес-процесса, которые не войдут в автоматизацию, будем применять выбранное средство для построения диаграмм Microsoft Visio, что в итоге позволит нитолько упростить, но и полностью увязать в единое целое информацию. Выбор данного инструментария был не случайным. Только за счет наличия мощного интерфейса, которым обладает Microsoft Visio, и множеством интересных опций для настроек предпочтение было отдано именно данному векторному редактору[1,с.28].

Интерес видится в получении исходных данных, показательных результатов опций и определенных необходимых ресурсов, которые можно

получить, применяя именно графический набор редактора. За счет открытия многих его функциональных возможностей, есть надежда получить в итоге взаимоувязанные связи между отдельными элементами, делая правильные шаги в области моделирования. Основное назначение системы – своевременный учет выполнения заказов клиентов, для чего требуется хранить и обрабатывать информацию о неиспользуемой продукции.

Автоматизируя процесс, ресторан «Лимпопо» сможет удовлетворить вкусы потребителей и повысить популярность организации среди потребителей, что в конечном итоге повысит доходы. Пользовательский запрос по базе данных чеков в окне интерфейса будет сформирован и обработан только при заданных настройках парсера, обрабатывающего данные чеков (таблица 3.3).

Редактор вопросов позволяет выполнять следующие операции: создавать и вносить правки в список запросов. Операция выполняется путем импорта параметров из файла конфигурации текстового запроса (POS-файл запроса) и/или с использованием внутренней процедуры [11, с.63].

Таблица 3.1 – Справочник «Чек» в БД АСУ «Ресторан «Лимпопо»»

Название столбца	Тип	Описание	Ключ
Id/№ записи	INT(11)/ число	Обязательное, неизменяемое поле	Primary
id_zak/ № заказа	INT(11)/число	Обязательное, изменяемое поле	
id_bl/ № блюда	INT(11)/ число	Обязательное, изменяемое поле	
Kolvo/количество блюд	INT(11)/ число	Обязательное, изменяемое поле	
Summa/сумма	DOUBLE (14,2)/ Число с плавающей запятой	Обязательное, изменяемое поле	

Редактор запросов позволяет осуществлять следующие операции: Создание и редактирование списка запросов. Операция производится

импортированием параметров из текстового файла конфигурации запросов (POS query's file) и/или с использованием встроенных процедур[11,с.63].

Объект Пользователь базы данных (БД) — это физическое или юридическое лицо, которое будет иметь доступ и пользоваться услугами информационной системы для получения информации (таблица 3.2).

Таблица 3.2 – Справочник «Пользователи» в БД «АСУ «Ресторан «Лимпопо»»

Id/№ записи	INT(11)/ число	Обязательное, неизменяемое поле	Primary
Login/логин пользователя	VARCHAR (255)/ строка	Обязательное, изменяемое поле	
Pass/пароль	VARCHAR (255)/ строка	Обязательное, изменяемое поле	
Prav/набор прав	VARCHAR (255)/ строка	Обязательное, изменяемое поле	
Name/имя пользователя	VARCHAR (50)/ строка	Обязательное, изменяемое поле	
Priznak/Наличие (признак)	VARCHAR (255)/строка	Обязательное, изменяемое поле	

Объект конфигурации «Справочник меню» является прикладным объектом и предназначен для описания списков данных. Его характерной особенностью является то, что пользователь в процессе работы может самостоятельно добавлять новые элементы в справочник (таблица 3.3).

Таблица 3.3 – Справочник «Блюда» в БД «АСУ «Ресторан «Лимпопо»»

Id/№ записи	INT(11)/ число	Обязательное, неизменяемое поле	Primary
Name/наименование блюда	VARCHAR (255)/ строка	Обязательное, изменяемое поле	
Calories/калорийность блюда	INT(8)/ число	Обязательное, изменяемое поле	
Cena/Цена блюда	DECIMAL (10,0)/ Точное число с десятичной точкой	Обязательное, изменяемое поле	

Продолжение таблицы 3.3

Discount/Возможная скидка	DECIMAL (10,0)/ Точное число с десятичной точкой	Необязательное, изменяемое поле	
Vid/Группа блюд	VARCHAR (255)/ строка	Обязательное, изменяемое поле	
Massa/масса блюда	DECIMAL (10,0)/ Точное число с десятичной точкой	Обязательное, изменяемое поле	
Priznak/Наличие (признак)	VARCHAR (255)/ строка	Обязательное, изменяемое поле	

Пользовательский объект базы данных (БД) представлен физическим или юридическим лицом, которое будет иметь доступ и позволит использовать службы информационной системы для получения необходимой информации (таблица 3.4).

Таблица 3.4 – Справочник «Клиенты» в БД «АСУ «Ресторан «Лимпопо»»

id_kl_f/№ записи	INT(11)/ число	Обязательное, неизменяемое поле	Primary
Name/ инициалы клиента	VARCHAR (50)/ строка	Обязательное, изменяемое поле	
Birthdate/ дата рождения	DATE/дата	Обязательное, изменяемое поле	
Dopinf/ дополнительная информация	VARCHAR (255)/ строка	Необязательное, изменяемое поле	
Nphone/ номер телефона	VARCHAR (15)/ строка	Обязательное, изменяемое поле	
name_org/ наименование организации	VARCHAR (255)/ строка	Необязательное, изменяемое поле	
Priznak/признак присутствия	VARCHAR (255)/ строка	Необязательное, изменяемое поле	

Справочник «Столики», направлено на текущее состояние бронированных и свободных столов, а так же позволяет определиться с выбором стола, заказать блюда из меню ко времени прихода (таблица 3.5).

Таблица 3.5 – Справочник «Столики» в БД АСУ «Ресторан «Лимпопо»»

Id/№ записи	INT(11)/ число	Обязательное, неизменяемое поле	Primary
Name/наименование объекта	VARCHAR (255)/ строка	Обязательное, неизменяемое поле	
dop_inf/ дополнительная информация	VARCHAR (255)/ строка	Необязательное, изменяемое поле	
date_n_a/дата начала аренды	DATETIME/дата и время	Обязательное, изменяемое поле	
id_kl/номер клиента	INT(11)/ числовой	Обязательное, изменяемое поле	

В информационной системе будут храниться данные о всех заказах и чеках, кроме того в ней составляется меню, в котором указана информация о блюдах и устанавливается их стоимость. Помимо таких базовых возможностей, в системе будет реализована возможность хранения информации о клиенте, официанте и сведениях о забронированных столиках (таблица 3.6).

Таблица 3.6 – Справочник «Заказы» в БД АСУ «Ресторан «Лимпопо»»

Id/№ записи	INT(11)/ число	Обязательное, неизменяемое поле	Primary
id_kl/уникальный номер клиента	INT(11)/ число	Обязательное, неизменяемое поле	
date_n/дата начала заказа	DATETIME/дата и время	Обязательное, неизменяемое поле	
date_o/дата окончания заказа	DATETIME/дата и время	Обязательное, изменяемое поле	
dop_inf/ дополнительная информация	VARCHAR (255)/ строка	Необязательное, изменяемое поле	
name_obj/ наименование столика	VARCHAR (255)/ строка	Обязательное, неизменяемое поле	
Priznak/ признак выполнения заказа	VARCHAR (255)/ строка	Обязательное, изменяемое поле	
Persona/персонал	VARCHAR (255)/ строка	Обязательное, неизменяемое поле	

Далее спроектируем базу данных, для этого используем СУБД MySQL. Для визуального проектирования БД используем dbForge Studio for MySQL. Благодаря ему можно наглядно представить структуру будущих таблиц и связи между ними. Это довольно простое решение, но оно позволяет спроектировать таблицы, обозначить первичные и внешние ключи, проверить связи. Схема представлена на рисунке 3.2.

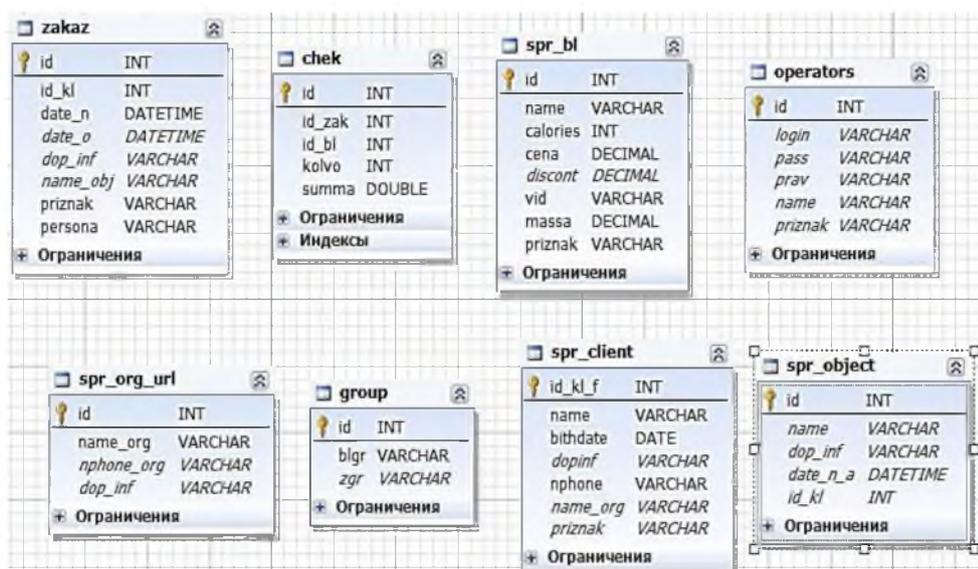


Рисунок 3.2 – Схема базы данных АСУ «Ресторан «Лимпопо»»

### 3.2 Организационное проектирование

Процесс организационного проектирования всегда предполагает то, что нужно изучить степень правильности выбранного программного решения, приемы и способы воплощения в сам проект определенных задумок. В нашем случае будем использовать язык Delphi и свободную реляционную базу данных MySQL.

#### Соединение с базой данных MySQL-Server

Для работы программного продукта с базой данных mysql был использован комплект компонентов доступа ZeosDBO[15,с.66].

Данный комплект не использует дополнительных посредников для передачи информации, т.е. работает напрямую. Для соединения с базой данных используется компонент ZConnection.

От использованной версии напрямую зависит название самого протокола.  
Пример настройки компонента ZConnection представлен на рисунке 3.3.

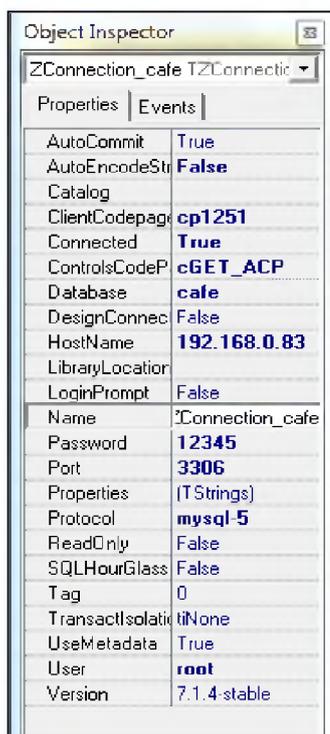


Рисунок 3.3 - Настройки компонента ZConnection

Рассматриваемый элемент управления представляет собой сетку, в которой строки соответствуют записям, а столбцы — полям в таблице базы данных (рисунок 3.4).

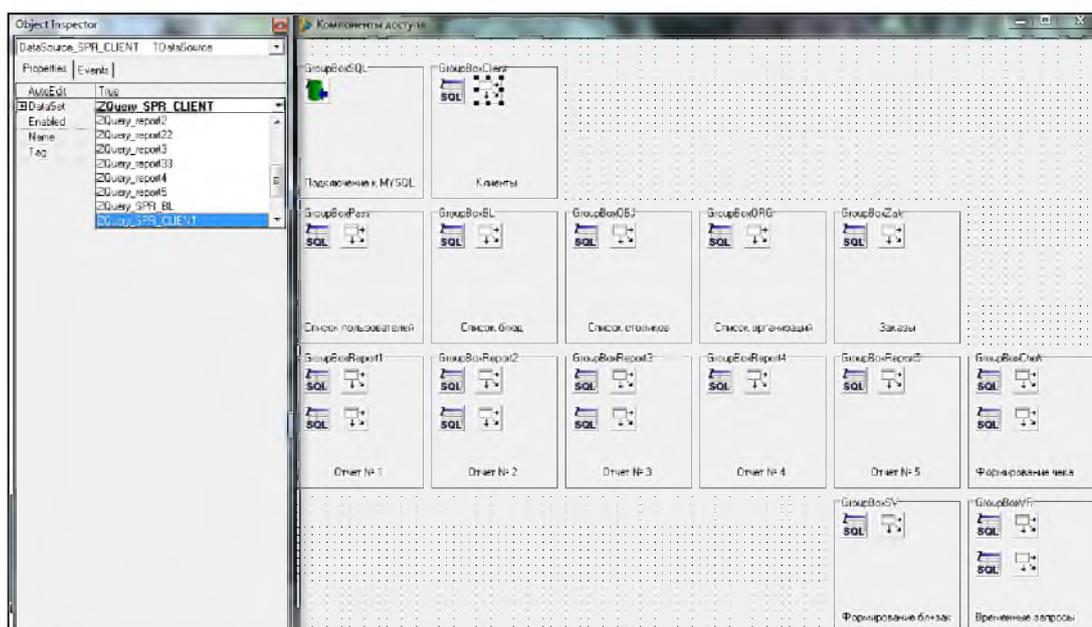


Рисунок 3.4 - Подключение компонента DataSource к ZQuery

В результате в ячейках сетки «Источники данных» отображаются конкретные значения полей для всех выбранных записей в базе данных. Соответственно в ячейках сетки DataSource отображаются конкретные значения полей для всех выбранных из БД записей. Также возможно отображение данных из базы данных в табличном виде с помощью других элементов управления[30,с.58].

Компонент выполнения запросов ZQuery соединяется с ZConnection. В свойстве «SQL» данного компонента прописывается запрос к базе данных MYSQL. Запросы к базе данных можно изменять в процессе работы программы (рисунок 3.5). Для удобства использования компоненты названы в соответствии с их функциями, например, для корректного подключения к таблице «Клиенты».



Рисунок 3.5 - Подключение компонента ZQuery к ZConnection

В качестве примера приведем алгоритм подключения таблицы «Клиенты», что позволит выделить наиболее существенные ниши и модульные позиции в самой работе.

ZQuery\_SPR\_CLIENT. В свойстве «sql» прописываем запрос «select \* from spr\_client». В свойстве «Connection» выбираем ZConnection\_cafe. В свойстве «Active» выбираем «true»(включаем выполнение запроса, если запрос правильной структуры «включение» выполнится).

DataSource\_SPR\_CLIENT подключается к ZQuery\_SPR\_CLIENT через свойство «dataset», в котором указывается компонент выполнения запроса ZQuery\_SPR\_CLIENT(рисунок 3.6).

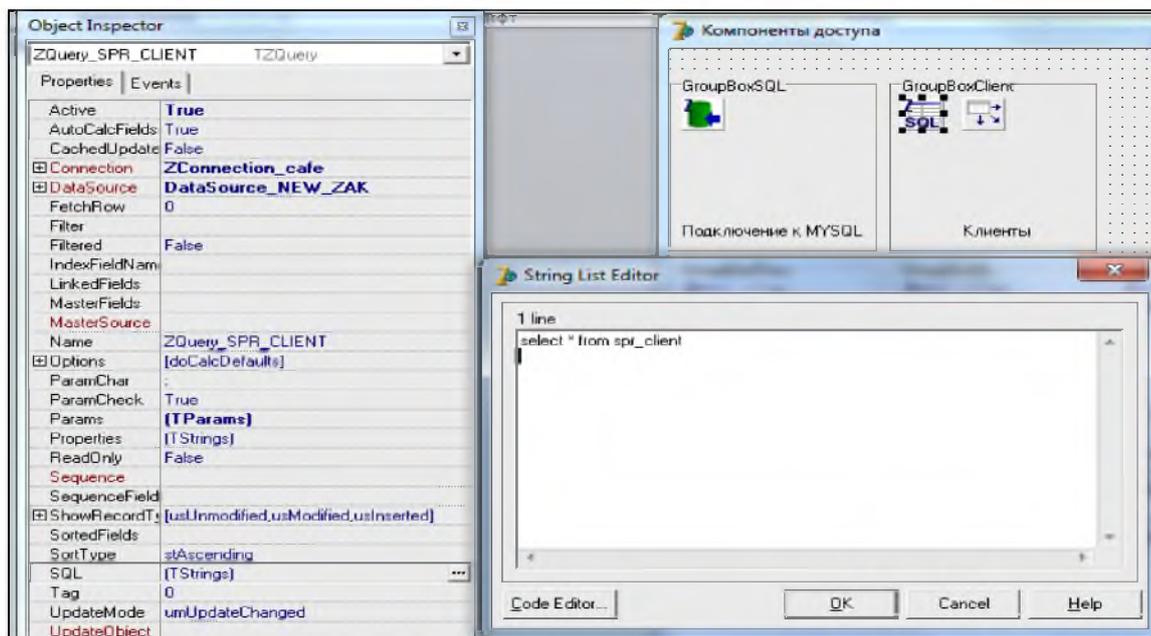


Рисунок 3.6 - SQL запрос в компоненте ZQuery

Вывод информации после выполнения действия или запроса в виде таблицы. Выводить информацию можно напрямую передавая ее свойству компонента. Для вывода информации в виде таблицы используется компонент DBGrid. Для соединения данного компонента используется свойство «DataSource», в котором выбирается источник данных DataSource. Результатом заполнения данного свойства станет вывод из базы данных информации указанной в запросе нашего ZQuery. Компонент GroupBox используется для визуальной группировки компонентов (рисунок 3.7).

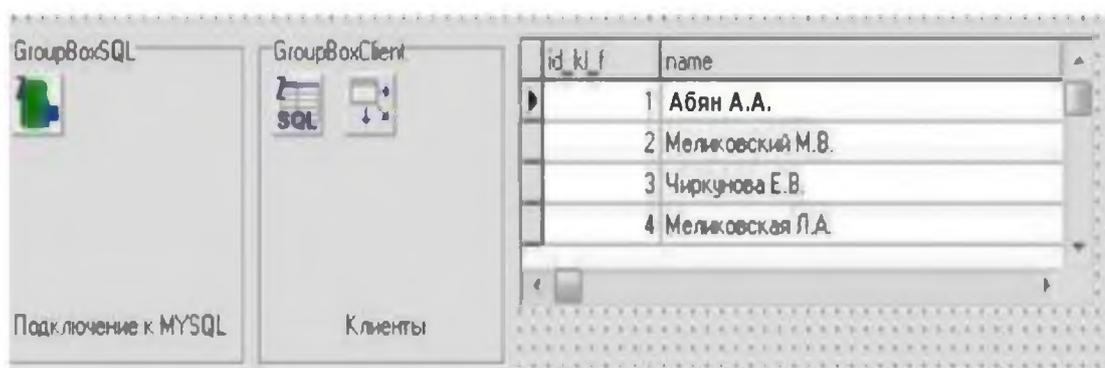


Рисунок 3.7 - Вывод информации из компонента

Пример 2. Вывод информации из таблицы «Пользователи»(SPR\_CLIENT) в текстовое поле напрямую из запроса представлен на рисунке 3.8.

В данном случае «инициатором» выполнения действия будет компонент BitBtn(кнопка).

В процедуре, которой написан следующий код:

```
procedure TForm_PASS.BitBtn1Click(Sender: TObject); begin
Form_DM.ZQuery_PASS.Close;
Form_DM.ZQuery_PASS.SQL.Text:='SELECT * FROM operators WHERE
login='1' ';
Form_DM.ZQuery_PASS.Open;
Form_KL_ZAK.Label11.Caption:=DBGrid1.DataSource.DataSet.fieldbyname
('prav').AsString;
Form_KL_ZAK.Label11.Caption:=DBGrid1.DataSource.DataSet.fieldbyname
('name').AsString;
End;
```

Пояснение: отключаем компонент выполнения запроса, присваиваем свойству «sql» новый запрос, выполняем запрос, передаем значение.

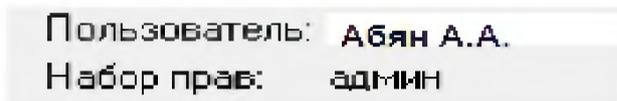


Рисунок 3.8 - Вывод информации из таблицы «Пользователи»

В качестве второго примера опишем алгоритм действий при выводе информации в текстовое поле из табличного значения «пользователи», соблюдая определенные параметры.

И тут необходимо помнить о том, что нивсегда компоненты архитектуры запроса можно получить, используя существующие параметры, которые применимы для вывода, редактирования или удаления информации по требованию.

В этом случае параметром запроса будет значение, введенное в текстовое поле:

```

Edit.text;
procedure TForm_PASS.BitBtn1Click(Sender: TObject); begin
Form_DM.ZQuery_PASS.Close;
Form_DM.ZQuery_PASS.SQL.Text:='SELECT * FROM operators WHERE
login=: p_login ';
Form_DM.ZQuery_PASS.Params.ParamByName('p_login').Value:= Edit1.text;
Form_DM.ZQuery_PASS.Open;
Form_KL_ZAK.Label11.Caption:=DBGrid1.DataSource.DataSet.fieldbyname
('prav').AsString;
Form_KL_ZAK.Label11.Caption:=DBGrid1.DataSource.DataSet.fieldbyname
('name').AsString;
End;

```

Структура базы данных, параметры компонентов Delphi.

Создание структуры базы данных подразумевает, что каждое поле в БД будет иметь свою структуру.

Например, число, дата, дата и время, текст. Для корректного выполнения запроса необходимо, чтоб параметр, участвующий в запросе был такого же типа, как и поле для которого устанавливается параметр.

Для того чтоб изменять тип полей в Delphi используются специальные функции [30,с.198].

Пример 3. Изменение типа данных для использования в запросах.

В данном случае, если есть желание выполнить запрос, который будет использовать параметр «дата и время» введенный в текстовое поле, необходимо выполнить преобразование.

Аналогично преобразуются и другие типы данных. Преобразование типов данных может происходить в любой момент выполнения действия. Число преобразований не ограничено.

StrToDateTime(MaskEdit\_DATE\_N.Text) - из «текста» в «дата и время»  
StrToInt(Edit1.Text)-из «текста» в «число».

Использование функции сравнения данных, выполнение по условию.

### Пример 3.1. Сравнение параметров с использованием переменных.

Для выполнения сравнения необходимо преобразовать данные и присвоить их переменным, которые будут их «хранить» для выполнения действия. Переменные тоже имеют свой тип, который зависит от того, что собираются в нем «хранить». В данном примере выполняется преобразование параметра «текст» в «дата и время», присваивается значение переменным и выполняется сравнение, в зависимости от выполнения сравнения выводится информационное сообщение. Для выполнения условий используется конструкция `if then else` (если...тогда...иначе), в которую вводятся условия [10,с.88]:

```
procedure TForm_report3_usl.BitBtn_predpClick(Sender: TObject); var
  a,b:TDateTime; begin
  a:=StrToDateTime(MaskEdit_DATE_N.Text);
b:=StrToDateTime(MaskEdit_DATE_O.Text); if (a>b) then begin
  ShowMessage('Дата не корректна'); end
else
  ShowMessage('Дата корректна');
```

Перечень используемых типов компонентов в структуре проекта:

— текстовые: Label (метка), LabelEdit (текст+метка), Edit (текст), MaskEdit (текст), Мемо(многострочный текст), DBGrid (таблица) (рисунок 3.9);

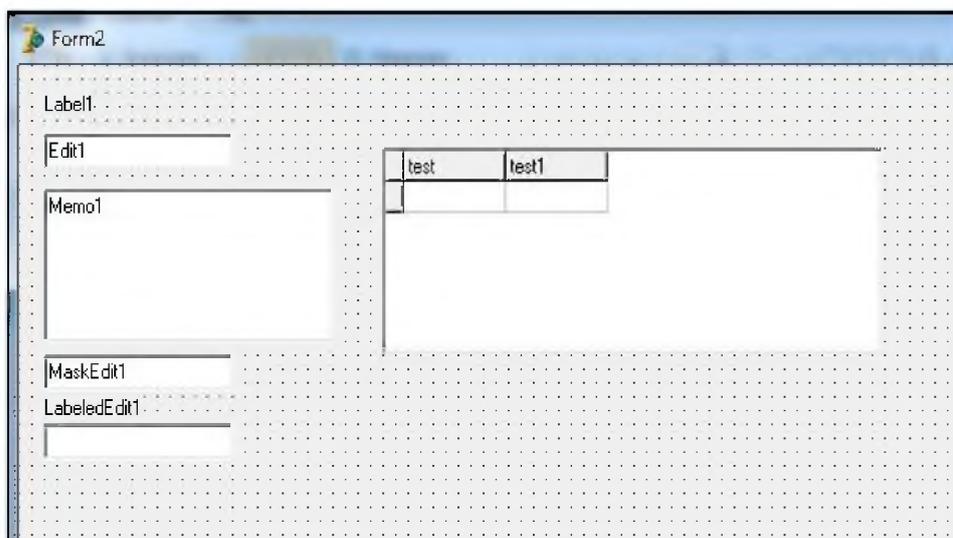


Рисунок 3.9 – Использование текстовых типов компонентов

— кнопки: BitBtn (графическая кнопка), Button (кнопка), RadioButton (переключатель), CheckBox(отметка)(рисунок 3.10);

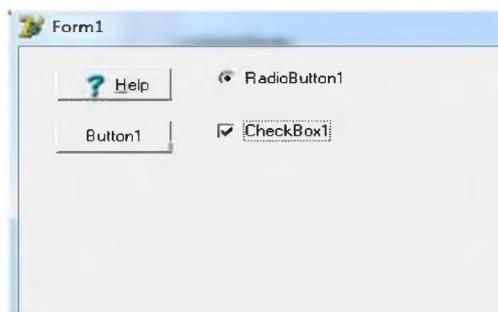


Рисунок 3.10 - Использование кнопок и переключателей

GroupBox (визуальные группировка), Panel (панель), MainMenu1 (меню), PageControl (вкладки) (рисунок 3.11).

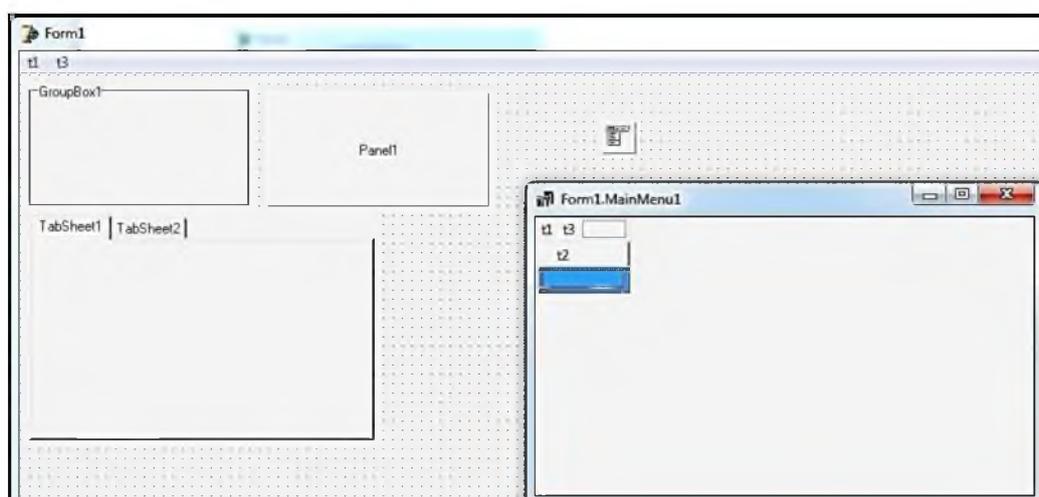


Рисунок 3.11 – Использование визуальных группировок

## 3.2 Организационное проектирование

Чтобы оценить степень качества проектного решения, продемонстрируем алгоритм работы программного продукта за счет индивидуальных вариантов.

Когда пользователь запускает программу для того, чтобы начать работу, он должен пройти авторизацию пользователя (рисунок 3.12).

Как будет выглядеть диалоговое окно самого администратора видно на следующем рисунке 3.13.

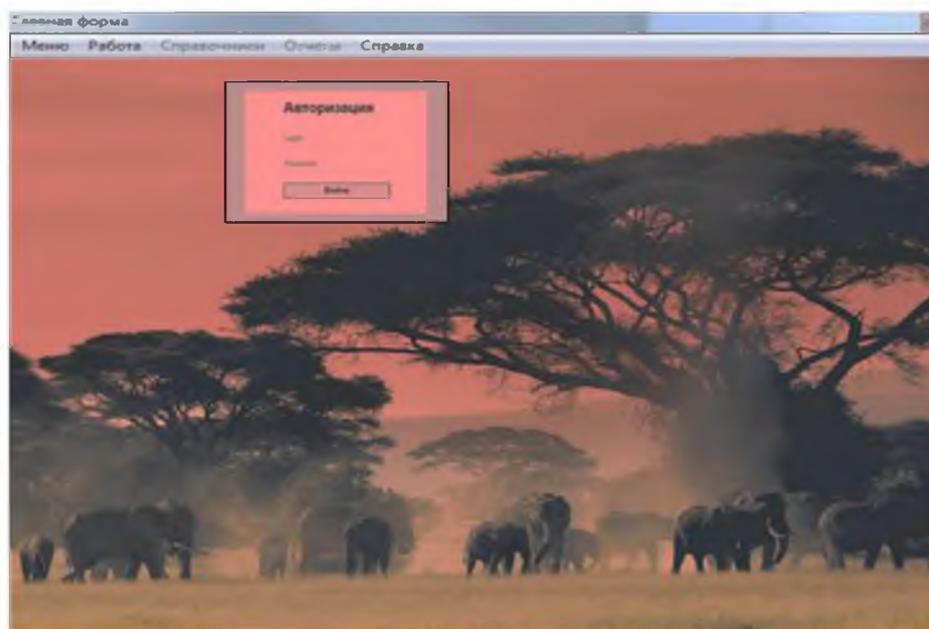


Рисунок 3.12 – Форма авторизации АСУ

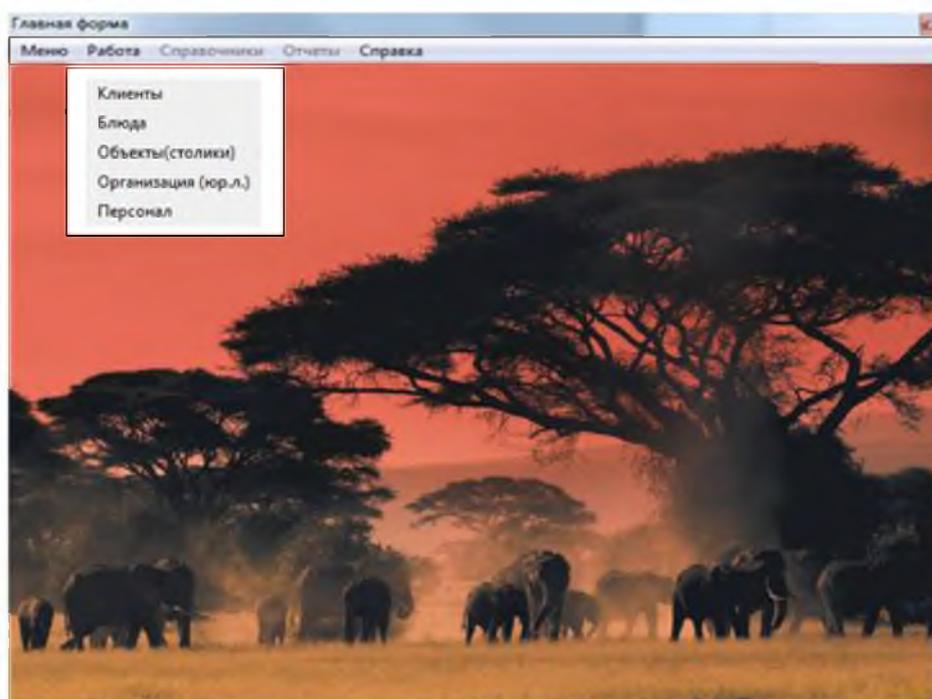


Рисунок 3.13 - Внешний вид формы для профиля прав «администратор»

Для работы с вкладкой заказы в меню программы выбираем:

Работа → Заказы.

В правом верхнем углу формы фиксируется какой пользователь вошел в программу и его профиль прав. В зависимости от прав пользователя изменяется

функционал. На верхней панели формы Клиенты и заказы располагаются кнопки:

Кнопка новый клиент. При нажатии на форме появляются поля для ввода информации о новом клиенте (рисунок 3.14).

Клиенты и заказы

Пользователь: Абян А.А.  
Набор прав: админ

Новый клиент      Редактировать      Удалить

Данные клиента

Ф.И.О.      Дата рождения      Номер телефона      Наименование организации

Яшин С.Ф.      15.08.1978      +7(316)201-5448      МУП "ЕИРЦ"

Дополнительная информация

Сохранить

Рисунок 3.14 – Добавление нового клиента

При нажатии кнопки «Редактировать» отображается форма изменений и корректировки. Открытое окно позволяет зайти в библиотеку и работать с источниками.

Пользователь с набором прав «официант» не наделен возможным функционалом к редактированию сведений и тем более для удаления информации из базы данных.

Возможность быстрого поиска с учетом динамического изменения параметров достигается за счет расположения табличных данных клиентов и заказов на средней панели. Обновление списка клиентов предполагает работу с базой данных, которая периодически пополняется или структурируется (рисунок 3.15).

Список клиентов

Поиск по фамилии: М      Обновить список

№ п.п.	Ф.И.О.	Дата рождения	Телефон	Дополнительная информация	Организация
2	Меликовский М.В.	15.08.1993	+7(908)692-1852		
4	Меликовская Л.А.	27.08.1971	+7(908)692-1852		ОАО "Сбербанк России"
6	Морозова Н.И.	15.09.1964	+7(903)465-1282		
20	Матвиенко Ю.В.	31.01.1978	+7(939)155-5535	Директор организации	ОАО "ТМТП"
32	Мелконян Р.А.	30.01.1971	+7(918)456-7512		ОАО "Еврохим"

Рисунок 3.15 – Технология поиска

Нижняя панель «Клиенты и заказы» связана со средней панелью выбранного клиента и состоит из двух вкладок.

В первую очередь появляется возможность выявления нового заказа для выбранного клиента с инструкциями сотрудников, работающих в ресторане и появляющегося из диалогового окна «новый заказ». Также возможно выполнение дополнительного информационного поля (рисунок 3.16).

Пользователи с набором прав «официант» не имеют возможности выбрать обслуживающий персонал из выпадающего списка официантов.

Новый заказ | Список Заказов

Дата и время начала: 28.08.2024 17:15

Выбор столика: столик №9

Дополнительная информация

Персонал: Киселева Л.В., Хохлова Т.И., Минаева Т.С., Кудряшева Н.С., Кулик И.В., Меликовский М.В., Печков С.В.

Сохранить

Рисунок 3.16 – Создание нового заказа

Создав и сохранив заказ, автоматически переходим на 2 вкладку – список заказов. Здесь представлен полный список заказов, так же существует возможность выбора только открытых заказов (рисунок 3.17).

Персоналу с набором прав «официант» доступны только персональные заказы.

Новый заказ | Список Заказов

персональный список | полный список |  только новые заказы

№ заказа	дата, время начала	дата, время окончания	столик	состояние	персонал
21	05.08.2024 18:24		столик №5	новый	Минаева Т.С.
22	05.08.2024 18:37		столик №10	новый	Хохлова Т.И.
23	05.08.2024 18:42		столик №1	новый	Киселева Л.В.
24	05.08.2024 18:53		столик №2	новый	Кудряшева Н.С.
25	05.08.2024 18:58		столик №9	новый	Кудряшева Н.С.

Предварительный просмотр чека

Печать чека

Закреть заказ

Список блюд | Редактировать состояние

Рисунок 3.17 – Список открытых заказов

После выбора заказа список блюд вносится в систему автоматического контроля путем выбора «меню», далее можно открыть информацию о текущем заказе и работать с ним, а именно вносить или удалять, добавлять или объединять сведения (рисунок 3.18).

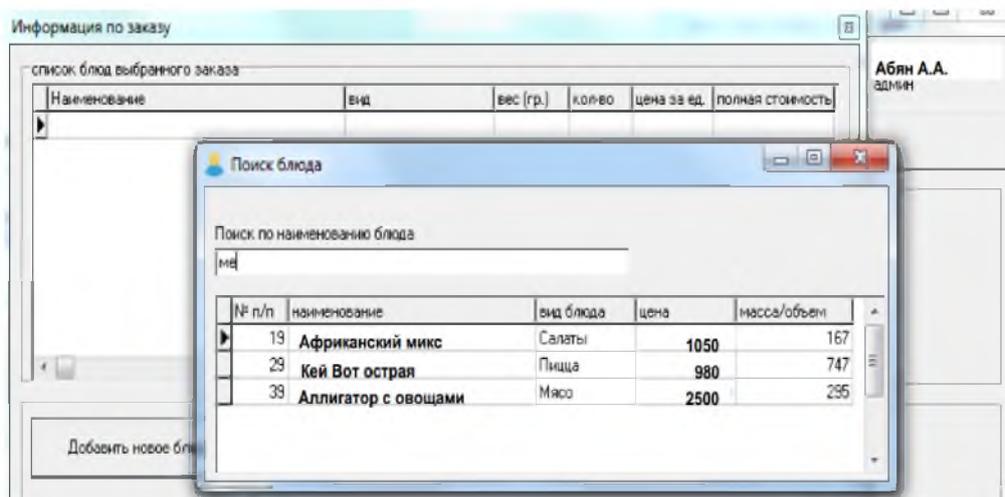


Рисунок 3.18 – Поиск блюда

После выбора блюда из справочника необходимо заполнить количество порций. Поле «сумма» считается автоматически, учитывая количество порций, возможную скидку, а также цену за единицу. Пример выбора блюда из справочника представлен на рисунке 3.19.

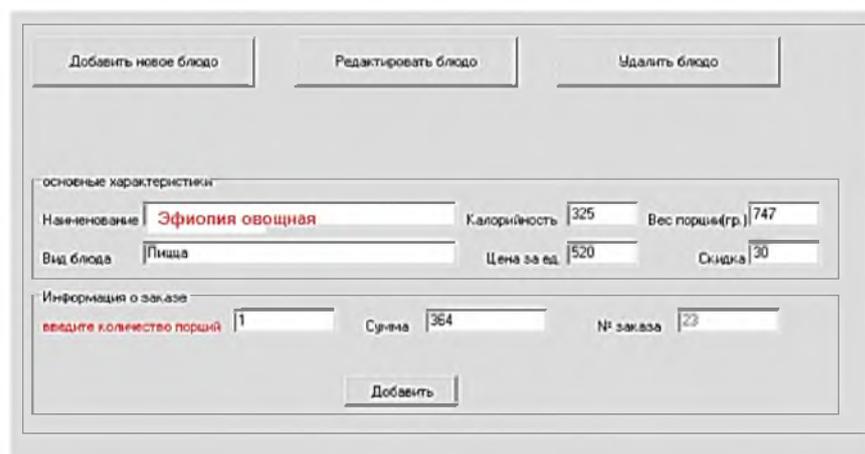


Рисунок 3.19 – Корректировка заказа

Чтобы получить и потом распечатать чек для отчетности, а многие клиенты запрашивают чековый квиток после закрытого заказа, нужно сделать

следующее: определить нужную дату, когда клиент сделал заявку на обед или ужин, потом закрыть обслуживание и это даст определенный сигнал программе сформировать полное закрытие по счету и выпуск финансового документа.

Если упустить что-то из последовательных действий, то программа воспринимает это как ошибку и закрывает программу автоматически.

В этом случае сам процесс работы начинают сначала.

Администратор не наделен полномочиями изменять что-то в рецептуре или списке уже сформированного меню блюд.

В данном случае программа ставит защиту в виде блокировки на компьютере доступа ко всем файлам. Но для продолжения работы блокировку можно снять путем нажатия на кнопку «редакция статуса заказ».

Данный функционал доступен только пользователю с набором прав «администратор». Когда заказ будет закрыт, можно переходить к печати чека при помощи соответствующей кнопки на вкладке заказы (рисунок 3.20).

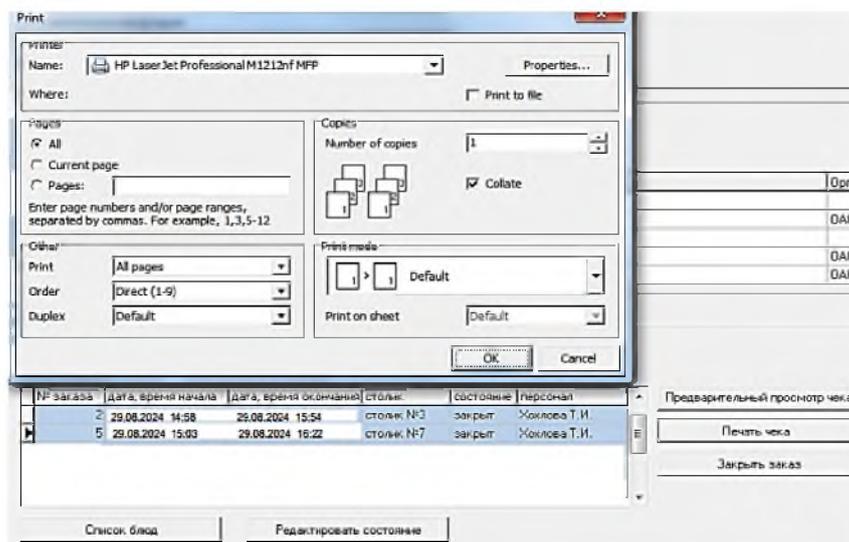


Рисунок 3.20 – Печать чека

Для построения аналитических отчетов используется вкладка отчеты.

Пользователю с набором прав «администратор» доступны следующие отчеты:

— Список блюд за определенный период формируется с учетом выбранного временного промежутка (рисунок 3.21).

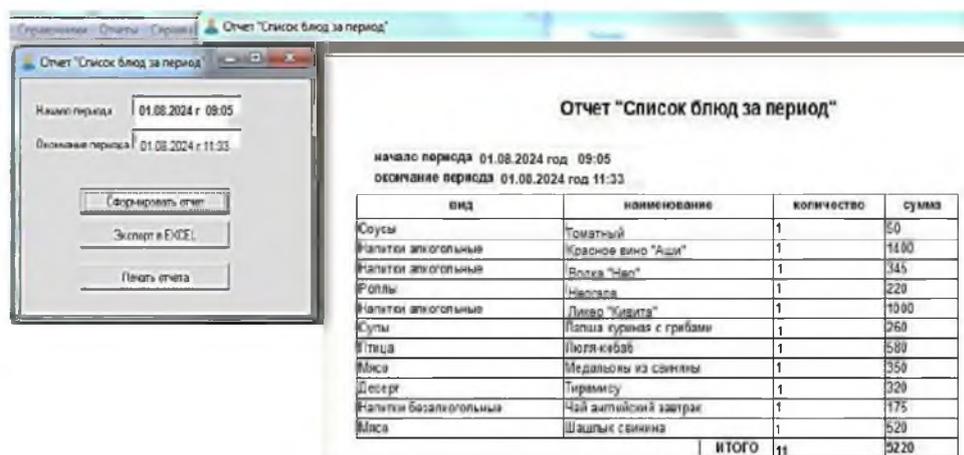


Рисунок 3.21 – Отчет «Список блюд за период»

— Работа официантов - формируется с учетом выбранного временного промежутка (рисунок 3.22);

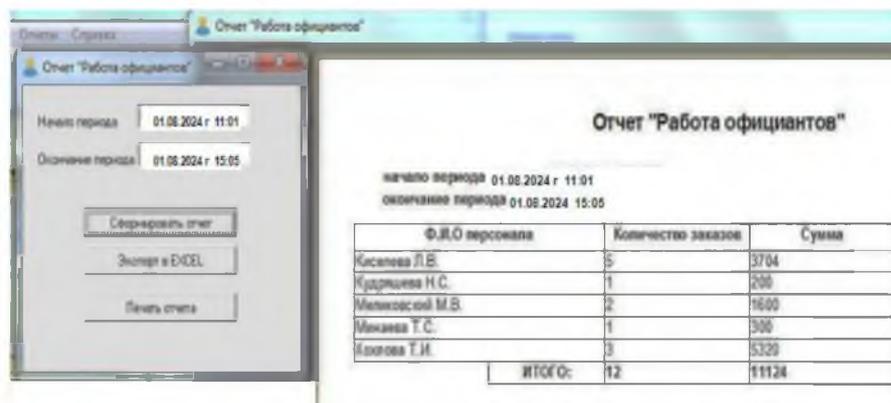


Рисунок 3.22 – Отчет «Работа официантов»

— Заказы и организации формируется с учетом выбранного временного промежутка, а также есть возможность указать конкретную организацию, по которой будет сформирован отчет (рисунок 3.23).

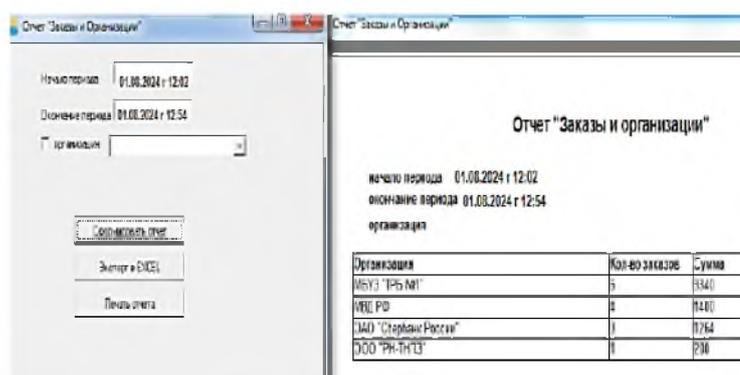


Рисунок 3.23 – Отчет «Заказы и организации»

Все отчеты имеют функцию основного отчета, печать и экспорт в Microsoft Excel. Microsoft Excel имеет административные ограничения при экспорте скрытых и отображаемых объектов отчета. Группы, строки и столбцы с редактируемыми элементами отчета преобразуются в структурные структуры Excel, которые позволяют разворачивать и сворачивать строки и столбцы во всей строке или столбце.

АСУ «Ресторан Лимпопо» возникла на основе 5 справочников:

Каталог «Клиент» содержит картографическую информацию о клиенте (рисунок 3.24). Для оформления заказа необходима регистрация клиента, которую можно сделать 2 способами:

Выбор нового лица – заполнение всех открывающихся полей для дальнейшей авторизации или верификации.

Физическое лицо вносится по определенному шаблону, а вот работа с юридическим лицом предполагает.

Применив справочник «Каталог клиентов» администратор может внести новое лицо без верификации.

Все действия по работе со сведениями в программе можно осуществлять только с правами «администратора», что не является одинаковым для других каталогов.

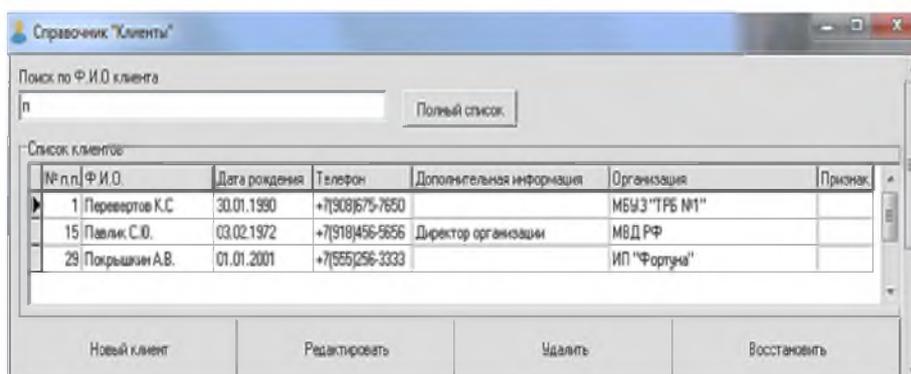


Рисунок 3.24 – Справочник «Клиенты»

Справочник «Организация» используется, если клиент является юридическим лицом и для регистрации используется название организации из реестра. Только пользователи с правами «администратора» имеют право

заполнять каталог организации (рисунок 3.25).

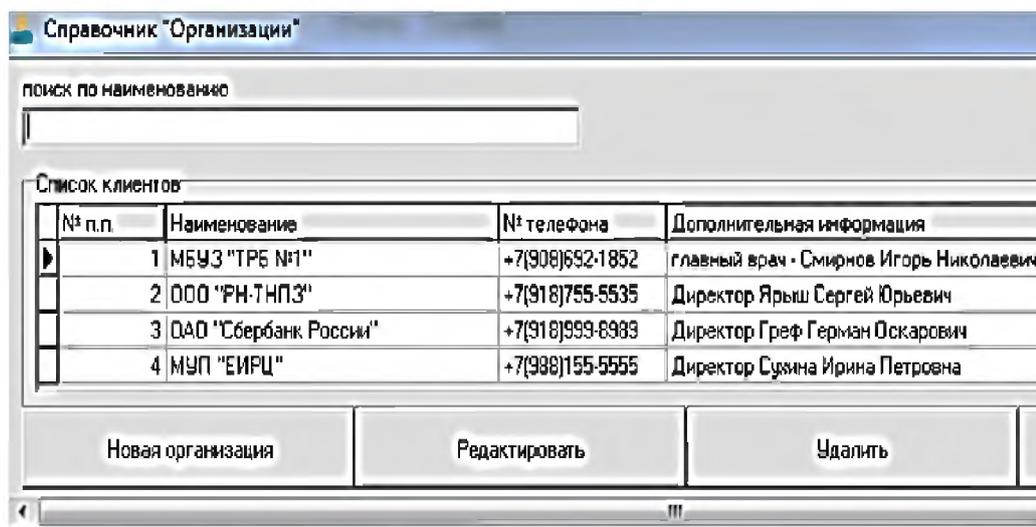


Рисунок 3.25 – Справочник «Организации»

Каталог объектов содержит много информации, которую вводит в справочное пособие администратор (рисунок 3.26).

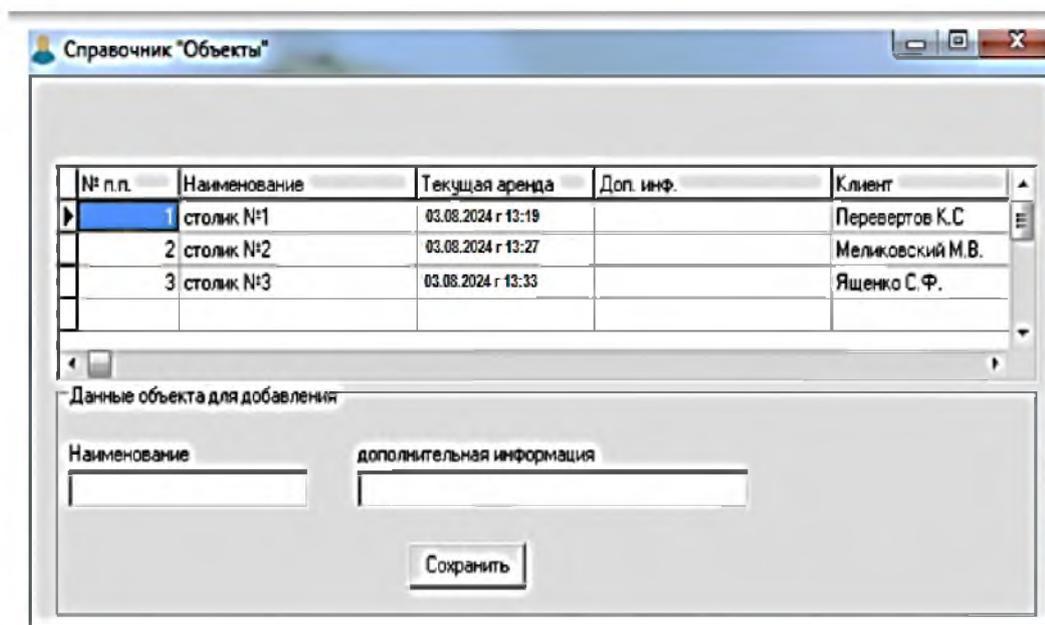


Рисунок 3.26 –Справочник «Объекты»

Справочник «Персонал» содержит в себе список лиц, имеющих право работать в программе.

Для персонала существует 2 набора прав: «официант» (ограниченные права) и «администратор» (полный набор прав) (рисунок 3.27).

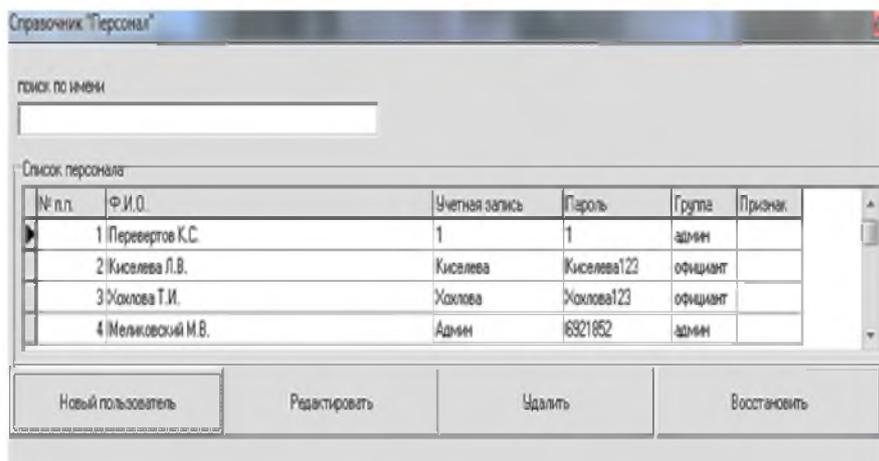


Рисунок 3.27 – Справочник «Персонал»

Справочник «Блюда» включает в себя список блюд, из которых формируется заказ.

Администратор имеет возможность «скрывать» блюда, входящие в «стоп - лист», т.е. официант при формировании заказа не сможет выбрать «скрытое» блюдо из справочника. Данный справочник имеет еще функционал ведения учета скидок для конкретного блюда, т.е. при формировании заказа программа будет учитывать скидку на конкретное блюдо (рисунок 3.28).

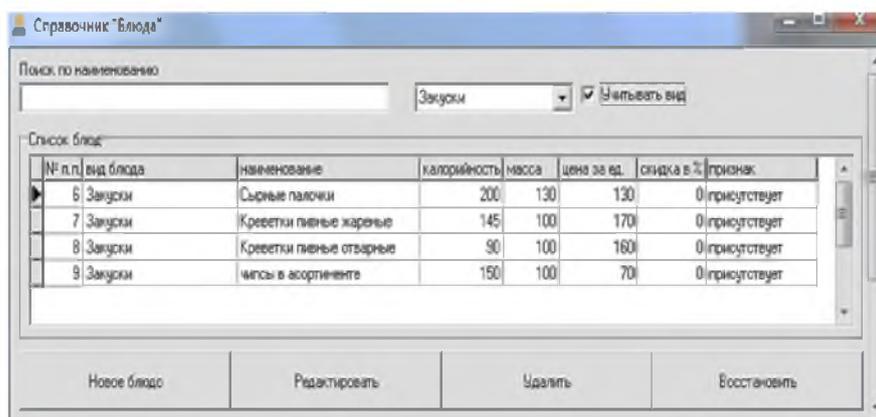


Рисунок 3.28 – Справочник «Блюда»

Для того чтобы что-то изменить или дополнить в справочные пособия программного продукта нужно воспользоваться кнопкой по редактированию сведения, а изменения необходимо внести, чтобы изменить форму и информацию, которая «сохраняется». Чтобы удалить запись из справочника, выбирают нужную строку в таблице и нажимают кнопку «Удалить».

Функция быстрого поиска также используется во всем, от формы до справочника. Чтобы выполнить быстрый поиск по полю поиска, вводят страницу персонажа, имя которого необходимо получить.

### 3.3 Организационно-экономическое обоснование проекта

Для того чтобы элементы цифровой инфраструктуры превратились в полноценную экосистему, их следует интегрировать. При этом все дополнительные типы, основанные на протоколе API, не учитываются. Их можно создать самостоятельно или обратиться за помощью к компаниям, предлагающим соответствующие или коммерческие услуги [8,с.19]. После завершения оптимизации бизнес-процессов ресторан получит множество преимуществ.

Польза от применения цифровизации, пожалуй, видна всем участникам технологического процесса, начиная от уборщицы, и заканчивая руководящим звеном ресторана. Качество обслуживания, например, напрямую зависит от технологического оснащения предприятия общепита. И это наглядно видно во многих аспектах, например, специалисты получают доступ к множеству уникальных инструментов для создания различных акций и поощрений для постоянных и новых клиентов.

Существуют различные варианты, такие как скидки в часы пик, резервирование мест и доставка по городу. Программы для автоматизации работы ресторана представляют собой сложные системы, требующие использования множества инструментов.

Платформы автоматизации предлагают множество преимуществ, включая оптимизацию работы сотрудников, увеличение доходов, создание прозрачной отчетности и упрощение взаимодействия с поставщиками. Цифровая инфраструктура интегрирует модули оплаты, маркетинга, управления запасами и меню, оказывая влияние на все аспекты функционирования заведения общественного питания.

Необходимо определить эффект автоматизации в африканском ресторане «Лимпопо» от цепочки организационного взаимодействия. В первую очередь в выигрышном положении оказывается зал. Сотрудники могут связаться с шеф-поваром напрямую, используя сенсорную панель или используя карманный компьютер или планшет, выданный компанией.

Программы для автоматизации учета в компаниях ресторанного бизнеса позволяют существенно упростить работу поваров. Через программное обеспечение реализован порядок подачи блюд посетителям. Посредством переданной информации шеф-повар точно знает, что и в какое время готовить.

Должностная задача бармена заключается в приготовлении напитков и обслуживании клиентов. В этом заведении процесс организован аналогично кухонной зоне. Все заказы фиксируются с помощью монитора или флажка. Данное программное обеспечение применяется среди прочего, для управления счетами. Здесь перечислены все товары, заказанные посетителями. Завершающим этапом является выдача фискальных квитанций, которые имеют статус документов, не имеющих значения до тех пор, пока они не действительны и не подчиняются всем правилам.

Оказывается, внедрение автоматизированной программы в деятельность ресторана «Лимпопо» существенно упрощает работу всех структурных подразделений и систематизирует все информационные потоки внутри компании.

Перейдем к расчету экономической эффективности проекта. В настоящее время многие уже понимают, что эффективность информационной системы определяется ее содержанием и качеством реализации: «правильностью» структур и процессов и т.д. [7, с.26].

В рамках комплексного проекта по внедрению клиентоориентированной стратегии и формированию системы продаж можно достичь значительно большего числа результатов, включая как эффекты снижения затрат, так и другие положительные результаты разных заказов. Для расчета экономического эффекта сравниваем существующую и автоматизированную тематическую

технологии по эксплуатационным затратам. Ожидаемый экономический эффект определяется по формуле [29,с.75]:

$$\Theta_0 = \Theta_r - E_n \times K_{\Pi} \quad (3.1)$$

где,  $\Theta_r$  - годовая экономия;

$K_{\Pi}$  - капитальные затраты на проектирование;

$E_n$  - нормативный коэффициент ( $E_n = 0,15$ ).

Годовая экономия  $\Theta_r$  складывается из экономии эксплуатационных расходов и экономии в связи с повышением производительности труда пользователя [29,с.107]:

$$\Theta_r = P_1 - P_2 \quad (3.2)$$

где,  $P_1$  и  $P_2$  - соответственно эксплуатационные расходы до и после внедрения с учетом коэффициента производительности труда.

Получим:

$$\Theta_r = 86007,5 - 8608,41 = 77399,09 \text{ руб.},$$

$$\Theta_0 = 77399,09 - 0,15 \times 57795,146 = 77399,09 - 8669,27 = 68729,82 \text{ руб.}$$

Рассчитаем фактический коэффициент экономической эффективности разработки по формуле [29,с.127]:

$$K_{\Theta\Phi} = \Theta_0 / K \quad (3.3)$$

$$K_{\Theta\Phi} = 68729,82 / 57795,146 = 1,18.$$

Так как  $K_{\Theta\Phi} > 0,2$ , проектирование и внедрение прикладной программы для ресторана «Лимпопо» эффективно.

Рассчитаем срок окупаемости разрабатываемого продукта:

$$T_{OK} = K / \Theta_0 \quad (3.4)$$

где,  $T_{ок}$  - время окупаемости программного продукта, в годах

Таким образом, срок окупаемости разрабатываемого проекта составляет:

$$T_{ок} = 57795,146/68729,82 = 0,84 \text{ (года).}$$

Проделанные расчеты показывают, что внедрение разработанной информационной системы имеет экономическую выгоду для предприятия сферы общественного питания (таблица 3.7). В ходе аналитической работы найдены все необходимые данные, доказывающие целесообразность и эффективность разработки данного программного обеспечения.

Таблица 3.7 - Сводная таблица экономического обоснования разработки и внедрения проекта для африканского ресторана «Лимпопо»

Показатель	Значение
Затраты на разработку проекта, руб.	57795,82
Общие эксплуатационные затраты, руб.	8608,41
Экономический эффект, руб.	68729,82
Коэффициент экономической эффективности	1,18
Срок окупаемости, лет	0,84

Затраты на разработку проекта составили 57795,82 руб., общие эксплуатационные затраты 8608,41, годовой экономический эффект от внедрения данной системы составит 77399,09 руб., ожидаемый экономический эффект составит 68729,82 руб., коэффициент экономической эффективности 1,18, срок окупаемости - 0,84 года.

Проделанные расчеты показывают, что внедрение разработанной информационной системы имеет экономическую выгоду для рассматриваемого ресторана.

## Заключение

К настоящему времени в России сформировалось большое количество ресторанных сетей, охватывающие как крупные города, так и регионы страны. Осознание прибыльности этого бизнеса и его бурное развитие способствует возникновению различных сетевых концепций.

Наряду с классическими сетями (фаст-фуд, кофейни, рестораны с традиционной формой обслуживания), отличающимися единой концептуальной направленностью появляются так называемые «сети по факту», холдинги, которые могут включать одновременно абсолютно не похожие по стилистике заведения, принадлежащие, однако, одному или нескольким владельцам. Подобное многообразие создаваемых на практике сетей диктует в свою очередь необходимость разработки различных схем управления.

Однако, эффективное управление сетевой ресторанной структурой невозможно без использования информационной (автоматизированной) системы управления.

Можно условно выделить два базовых подхода к управлению. Первый подход - централизованная схема управления. Для нее характерен следующий принцип работы: существует единый управляющий центр, в котором происходят формирование меню, назначение цен, закупочная деятельность, расчеты с поставщиками, а также отслеживание всех перемещений продукции с центрального распределяющего склада во все торговые точки сети. Здесь же ведется бухгалтерский учет, проводится анализ поступающих данных и формируется вся необходимая финансовая отчетность. Второй подход - децентрализованное автоматизированное управление. При такой схеме работы в офисе происходят только сбор и анализ данных о работе торговых точек и принятие глобальных решений о дальнейшем развитии предприятия, а вся текущая деятельность проводится в самих заведениях.

Однако, на практике схема управления строится преимущественно по смешанному сценарию. Хочется отметить, что автоматизированные системы

управления помогают бороться и с хищениями на предприятии питания, однако, панацеи от злоупотреблений в сфере общественного питания и услуг не существует, но наиболее эффективным способом является использование автоматизированной системы компьютерного учета. Внедряя подобную систему, владелец ресторана должен понимать, что система - это только инструмент, который может предоставить мощные рычаги контроля, не сравнимые по эффективности с ручным учетом. На этапе внедрения системы команда профессионалов объясняет, как пользоваться данными рычагами самой системы. Стоит отметить, что выбор модели управления предприятием питания зависит от многих факторов: квалификации персонала, концепции заведения, наличие склада-распределителя, уровня сервиса поставщиков и т.д. Задача же информационной системы - обеспечить модель управления, которую выбрало руководство предприятия, и в случае изменения подхода к управлению быть готовыми поддержать эти изменения.

Целью выпускной квалификационной работы ставилась работа по проектированию информационной системы для автоматизации деятельности африканского ресторана «Лимпопо».

Для достижения поставленной цели были решены следующие задачи:

- проведено исследование деятельности рассматриваемого предприятия, описаны его основные бизнес-процессы, обоснована необходимость разработки и внедрению информационной программы;
- дан краткий анализ уже существующих систем и выбрана стратегия автоматизации предприятия;
- спроектирована информационная база программного решения;
- реализовано программное решение и приведен контрольный пример;
- проведен расчет экономической эффективности от проектного решения.

Объектом исследования являлось предприятие сферы общественного питания - ресторан «Лимпопо». С помощью проектирования было предложено автоматизировать следующие функции:

- регистрация посетителя с указанием первичной информации;
- создание заказа с непосредственным бронированием столика;
- заполнение справочников клиентов, блюд, столиков, организаций, персонала;
- построение отчетной документации.

Для реализации программного решения было предложено использовать среду программирования Delphi и базу данных СУБД MySQL. Применение информационной программы позволит совершенствовать бизнес-процесс приема и обслуживания клиентов, что повысит конкурентоспособность и эффективность ресторана «Лимпопо».

## Список литературы

1. Автоматизированные системы управления предприятием: учеб. под ред. Прокопенко, А.Г. – М.: Кнорус, 2020. – 137 с.
2. Анализ рынка услуг ресторанов и кафе в России / под ред. Ромащенко В.В. – М.:Наука, 2023. -307 с.
3. Бакулев, А.Н. Новый стандарт автоматизации ресторанного бизнеса // IT News. - № 8 (11). - С. 12-17.
4. Батаев, А.В. Операционные системы и среды: учеб. - М.: Академия, 2023.- 272 с.
5. Библиотека стандартных подсистем: учеб. под ред. Мухович, И.Д. – М.: Стандарт, 2022. – 199 с.
6. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. - М.: ИНФРА, 2021. - 478 с.
7. Гагарина, Л.Г. Автоматизированные информационные системы: учеб. пособие. - М.: МИЭТ, 2022. - 144 с.
8. Гвоздева, Т.В. Проектирование информационных систем: технология автоматизированного проектирования. - СПб.: Лань, 2023. - 156 с.
9. Демьянова, О.В., Мур, Н.Н., Новикова, Е.Н. Информационные технологии // Проблемы современной экономики. – 2020. – № 1 (33). – С.14-17.
10. Диго, С.М. Создание баз данных в среде СУБД Access: учеб.пособие. - М.:МЭСИ, 2020. - 105 с.
11. Карпова, Т.С. Базы данных: модели, разработка, реализация: учеб. - СПб.:Питер, 2022. - 304 с.
12. Круглова, О.В., Пак, Е.Н. Информационные технологии в управлении учреждением сферы общественного питания:учеб.– М.:«Конкорд», 2021.–134 с.
13. Куперштейн, В.И. Современные информационные технологии в управлении ресторанным бизнесом. - СПб.:БХВ, 2020. - 205 с.
14. Логинов, В.Н. Информационные технологии: учеб.пособие. – М.: КНОРУС, 2021. -240 с.

15. Марков, А.С. База данных. Введение в теорию и методологию: учеб. - М.: Высшая школа, 2021. - 512 с.
16. Мартишин, С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench. - М.: Форум, 2023. - 62 с.
17. Мартишин, С.А. Автоматизация бизнес-систем.-М.:Форум,2023.- 94 с.
18. Методология проектирования автоматизированной системы: учеб. под ред. Житомирского, А.Г. – М.: Проспект, 2022. – 204 с.
19. Мясникова, Н.А. Алгоритмы и структуры данных: учеб.пособие.- Москва: КноРус, 2021. - 185 с.
20. Назаров, С.В. Программирование на MS Visual Basic: учеб.пособие. - М.: Финансы и статистика, 2022. - 320 с.
21. Назаров, С.В.Эффективность и оптимизация компьютерных систем.- Москва: Русайнс, 2023. - 219 с.
22. Нормативно - правовое регулирование деятельности сферы общественного питания: учеб.пособие под ред. Именко, О.Д. – М.:Высшая школа, 2022. -177 с.
23. Синаторов, С.В. Пакеты прикладных программ: учеб.пособие.- М.: КноРус, 2022. - 195 с.
24. Степаненко, В.В. Введение в UML. – М.:КноРус,2022. – 208 с.
25. Федоренко, Н.П. Справочник разработчика АСУ: учеб.пособие. - М.: Экономика, 2023. - 347 с.
26. Федосеева, Н.Н. Методология IDEF0: учеб.пособие. – М.:Высшая школа, 2021. – 309 с.
27. Хансен, Г. База данных. Разработка и управление: учеб. - М.: Беном, 2023. - 148 с.
28. Хомоненко, А.Д. Базы данных: учеб. - М.: Крона, 2021. - 589 с.
29. Черемных, С.В. Экономика предприятий и организацийб учеб. - М.:КФО,2021. - 532 с.
30. Шарина, А.А. Языки программирования: учеб.пособие. - СПб.:Питер Пресс, 2021. - 592 с.

Приложение  
Листинг программы

```
unit Menu;
interface uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus, StdCtrls, Buttons, jpeg, ExtCtrls;
type
TForm_MENU =class(TForm) MainMenu1: TMainMenu; N1: TMenuItem;
N2: TMenuItem;
N3: TMenuItem;
N4: TMenuItem;
N5: TMenuItem;
N6: TMenuItem;
N7: TMenuItem;
N8: TMenuItem;
N9: TMenuItem;
N11: TMenuItem;
N12: TMenuItem;
N14: TMenuItem;
Edit1: TEdit;
N10: TMenuItem;
Image1: TImage;
N13: TMenuItem;
N15: TMenuItem;
N16: TMenuItem;
N17: TMenuItem;
Edit2: TEdit;
N18: TMenuItem;
N19: TMenuItem;
```

```

N20: TMenuItem;
procedure N6Click(Sender: TObject); procedure N12Click(Sender: TObject);
procedure N14Click(Sender: TObject); procedure N9Click(Sender: TObject);
procedure N2Click(Sender: TObject); procedure N7Click(Sender: TObject);
procedure N11Click(Sender: TObject); procedure N8Click(Sender: TObject);
procedure N3Click(Sender: TObject); procedure N13Click(Sender: TObject);
procedure N15Click(Sender: TObject); procedure N16Click(Sender:
TObject);procedure N17Click(Sender: TObject); procedure N18Click(Sender:
TObject); procedure N20Click(Sender: TObject);
private
{ Private declarations } public
{ Public declarations } end;
var
Form_MENU: TForm_MENU; implementation
uses Glav, Data_module, Org_url, Klient_zakaz, PASS, SPR_CLIENT, SPR_BL,
SPR_PERS, SPR_OBJECT, INF_ZAK, REPORT1, REPORT1_usl, REPORT2,
REPORT2_usl, REPORT3_usl, REPORT4_usl, REPORT5_usl, SPRAVKA;
{$R *.dfm}
procedure TForm_MENU.N6Click(Sender: TObject); begin
Form_SPR_ORG_URL.Show; end;
procedure TForm_MENU.N12Click(Sender: TObject); begin
Form_KL_ZAK.Show;
if Form_KL_ZAK.Label11.Caption='админ' then begin
Form_KL_ZAK.DBLookupComboBox_PERSONA.Enabled:=true;
Form_KL_ZAK.BitBtn4.Enabled:=true;
end else
Form_KL_ZAK.DBLookupComboBox_PERSONA.KeyValue:=Form_KL_ZAK.Lab
el8.Caption; end;
procedure TForm_MENU.N14Click(Sender: TObject); begin
Form_MENU.close; end;

```

```

procedure TForm_MENU.N9Click(Sender: TObject); begin
Form_SPR_CLIENT.Show; end;

procedure TForm_MENU.N2Click(Sender: TObject); begin
Form_PASS.Show;
end;

procedure TForm_MENU.N7Click(Sender: TObject); begin
Form_SPR_BL.Show; end;

procedure TForm_MENU.N11Click(Sender: TObject); begin
Form_PERSONAL.Show; end;

procedure TForm_MENU.N8Click(Sender: TObject); begin
Form_OBJECT.Show; Form_DM.ZQuery_SPR_OBJECT.Close;
Form_DM.ZQuery_SPR_OBJECT.SQL.Clear;
Form_DM.ZQuery_SPR_OBJECT.SQL.Add('select * from SPR_OBJECT ,
spr_client where spr_client.id_kl_f=SPR_OBJECT.ID_KL');
Form_DM.ZQuery_SPR_OBJECT.Open; end;

procedure TForm_MENU.N3Click(Sender: TObject); begin
form_menu.N8.Enabled:=false; form_menu.N7.Enabled:=false;
form_menu.N6.Enabled:=false; form_menu.N9.Enabled:=false;
form_menu.N10.Enabled:=false; form_menu.N11.Enabled:=false;
form_menu.N5.Enabled:=false;
form_menu.N4.Enabled:=false; form_menu.N12.Enabled:=false; Form_PASS.Show;
form_pass.Edit2.Text:=""; Form_DM.ZQuery_NEW_ZAK.Close;
Form_DM.ZQuery_NEW_ZAK.SQL.Clear;
Form_DM.ZQuery_NEW_ZAK.SQL.Add('SELECT zakaz.id_kl, zakaz.id,
zakaz.date_n,zakaz.date_o, zakaz.name_obj,persona, zakaz.priznak from zakaz');
Form_DM.ZQuery_NEW_ZAK.Open;
Form_KL_ZAK.DBLookupComboBox_PERSONA.Enabled:=false;
Form_KL_ZAK.BitBtn4.Enabled:=false;
end;

procedure TForm_MENU.N13Click(Sender: TObject);

```

```

begin Form_report1usl.show; end;
procedure TForm_MENU.N15Click(Sender: TObject); begin
Form_report2usl.Show; end;
procedure TForm_MENU.N16Click(Sender: TObject); begin
Form_report3_usl.Show;
Form_report3_usl.DBLookupComboBox_PERSONA.Enabled:=false; end;
procedure TForm_MENU.N17Click(Sender: TObject); begin
Form_report4_usl.Show; end;
procedure TForm_MENU.N18Click(Sender: TObject); begin
Form_report5_usl.Show;
Form_report5_usl.DBLookupComboBox_ORG.KeyValue:="";
Form_report5_usl.CheckBox1.Checked:=false;
Form_report5_usl.DBLookupComboBox_ORG.Enabled:=false; end;
procedure TForm_MENU.N20Click(Sender: TObject); begin
Form_SPRAVKA.Show; end;
unit Data_module;
interface uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ZAbstractRODataset, ZAbstractDataset, ZDataset,
ZAbstractConnection, ZConnection, StdCtrls, Grids, DBGrids;
type
TForm_DM = class(TForm) ZConnection_cafe: TZConnection; ZQuery_ORG_URL:
TZQuery;
DataSource_ORG_URL: TDataSource; ZQuery_SPR_CLIENT: TZQuery;
DataSource_SPR_CLIENT: TDataSource; ZQuery_SPR_ORG_URL: TZQuery;
DataSource_SPR_ORG_URL: TDataSource; ZQuery_NEW_ZAK: TZQuery;
DataSource_NEW_ZAK: TDataSource; DataSource_SPR_OBJECT: TDataSource;
ZQuery_SPR_OBJECT: TZQuery; ZQuery_SPR_BL: TZQuery;
DataSource_SPR_BL: TDataSource; ZQuery_CHECK: TZQuery;
DataSource_CHECK: TDataSource; ZQuery_CHEK_BL: TZQuery;

```

```

DataSource_CHEK_BL: TDataSource; ZQuery_PASS: TZQuery;
DataSource_PASS: TDataSource; ZQuery_CHEK_P: TZQuery;
DataSource_CHEK_P: TDataSource; ZQuery_WRITE: TZQuery;
DataSource_WRITE: TDataSource; GroupBox1: TGroupBox; DataSource_report1:
TDataSource
ZQuery_report1: TZQuery; ZQuery_report11: TZQuery; DataSource_report11:
TDataSource; GroupBox2: TGroupBox; ZQuery_report2: TZQuery;
ZQuery_report22: TZQuery; DataSource_report2: TDataSource;
DataSource_report22: TDataSource; GroupBox3: TGroupBox; ZQuery_report3:
TZQuery; ZQuery_report33: TZQuery; DataSource_report3: TDataSource;
DataSource_report33: TDataSource; GroupBox4: TGroupBox; ZQuery_report4:
TZQuery; DataSource_report4: TDataSource; GroupBox5: TGroupBox;
DataSource_report5: TDataSource; ZQuery_report5: TZQuery;
GroupBox6: TGroupBox; GroupBox7: TGroupBox; Label1: TLabel;
Label2: TLabel; Label3: TLabel; Label4: TLabel; Label5: TLabel; Label6: TLabel;
Label7: TLabel;
GroupBox8: TGroupBox;
Label8: TLabel;
GroupBox9: TGroupBox;
Label9: TLabel; GroupBox10: TGroupBox;
Label10: TLabel; GroupBox11: TGroupBox;
Label11: TLabel; GroupBox12: TGroupBox;
Label12: TLabel; GroupBox13: TGroupBox;
Label13: TLabel; GroupBox14: TGroupBox;
Label14: TLabel; GroupBox15: TGroupBox;
Label15: TLabel;
private
{ Private declarations } public
{ Public declarations } end;
var

```

```

Form_DM: TForm_DM; implementation
{$R *.dfm} unit INF_ZAK; interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs,
Grids, DBGrids, StdCtrls, Buttons;
type
TForm_INF_ZAK = class(TForm) GroupBox1: TGroupBox; GroupBox2:
TGroupBox; DBGrid1: TDBGrid; GroupBox4: TGroupBox; Edit_Name_bl: TEdit;
Label2: TLabel;
Label3: TLabel;
Edit_calor_bl: TEdit;
Label4: TLabel;
Edit_massa: TEdit;
Edit_group_bl: TEdit;
Label5: TLabel;
Edit1: TEdit; Edit_N_zak: TEdit; Edit_id_bl: TEdit;
procedure Button_osm_checkClick(Sender: TObject); procedure
Button_new_chekClick(Sender: TObject); procedure
Button_redact_chekClick(Sender: TObject); procedure
BitBtn_add_chekClick(Sender: TObject); procedure BitBtn_svernClick(Sender:
TObject); procedure BitBtn_RedactClick(Sender: TObject); procedure
FormActivate(Sender: TObject);
procedure Edit_summa_chekClick(Sender: TObject); procedure
Button1Click(Sender: TObject); procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject); procedure Button4Click(Sender:
TObject); procedure DBGrid1CellClick(Column: TColumn);
private
{ Private declarations } public
{ Public

```

```

declarations } end;

var
Form_INF_ZAK: TForm_INF_ZAK; implementation
uses Data_module, Klient_zakaz, ZAKAZ_Poisk_BL, REPORT_CHEK;
{$R *.dfm}
procedure TForm_INF_ZAK.Button_osm_checkClick(Sender: TObject); begin
Form_INF_ZAK.Height:=622;
Edit_Name_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('name').AsString;
Edit_calor_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('calories').AsString;
Edit_massa.Text:=DBGrid1.DataSource.DataSet.fieldbyname('massa').AsString;
Edit_group_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('group').AsString;
Edit_discont_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('discont').AsString;
Edit_cena_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('cena').AsString;
Edit_N_zak.Text:=Form_KL_ZAK.DBGrid_ZAKAZ.DataSource.DataSet.fieldbyname('id').AsString;
Edit_id_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString; end;
procedure TForm_INF_ZAK.Button_new_chekClick(Sender: TObject); begin
Form_POISK_BL.Show; Form_DM.ZQuery_SPR_BL.SQL.Clear;
Form_DM.ZQuery_SPR_BL.SQL.Add('select * from spr_bl WHERE
priznak=:PRIZNAK');
Form_DM.ZQuery_SPR_BL.Params.ParamByName('PRIZNAK').Value:='присутств
yer'; Form_DM.ZQuery_spr_bl.Open;
end;
procedure TForm_INF_ZAK.Button_redact_chekClick(Sender: TObject); begin
Form_INF_ZAK.Height:=622;
Edit_Name_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('name').AsString;
Edit_calor_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('calories').AsString;
Edit_massa.Text:=DBGrid1.DataSource.DataSet.fieldbyname('massa').AsString;
Edit_group_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('vid').AsString;
Edit_discont_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('discont').AsString;

```

```

Edit_cena_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('cena').AsString;
Edit_N_zak.Text:=Form_KL_ZAK.DBGrid_ZAKAZ.DataSource.DataSet.fieldbyname('id').AsString;
Edit_kolvo_chek.Text:=DBGrid1.DataSource.DataSet.fieldbyname('kolvo').AsString;
Edit_summa_chek.Text:=DBGrid1.DataSource.DataSet.fieldbyname('summa').AsString;
Edit_N_zak.Text:=Form_KL_ZAK.DBGrid_ZAKAZ.DataSource.DataSet.fieldbyname('id').AsString;
Form_INF_ZAK.BitBtn_Redact.Visible:=true; Edit_discont_bl.Enabled:=false;
end;
procedure TForm_INF_ZAK.BitBtn_add_chekClick(Sender: TObject); begin
Form_DM.ZQuery_CHECK.Close; Form_DM.ZQuery_CHECK.SQL.Clear;
Form_DM.ZQuery_CHECK.SQL.Add('INSERT INTO chek (id_zak, id_bl,kolvo,summa)'); Form_DM.ZQuery_CHECK.SQL.Add('VALUES (:Param1, :Param2, :Param3, :Param4)');
Form_DM.ZQuery_CHECK.Params.ParamByName('Param1').Value:=Form_KL_ZAK.DBGrid_ZAKAZ.DataSource.DataSet.fieldbyname('id').AsString;
Form_DM.ZQuery_CHECK.Params.ParamByName('Param2').Value:=StrToInt(Edit_id_bl.Text);
Form_DM.ZQuery_CHECK.Params.ParamByName('Param3').Value:=StrToInt(Edit_kolvo_chek.Text);
Form_DM.ZQuery_CHECK.Params.ParamByName('Param4').Value:=StrToInt(Edit_summa_chek.Text);
Form_DM.ZQuery_CHECK.ExecSQL; Form_DM.ZQuery_CHEK_BL.Close;
Form_DM.ZQuery_CHEK_BL.SQL.Clear;
Form_DM.ZQuery_CHEK_BL.SQL.Add('SELECT spr_bl.name,spr_bl.calories, spr_bl.cena,spr_bl.massa,spr_bl.vid, spr_bl.discont, chek.kolvo, chek.summa, chek.id_bl FROM chek, spr_bl WHERE (chek.id_bl=spr_bl.id) and (chek.id_zak=:ID)'); Form_DM.ZQuery_CHEK_BL.Params.ParamByName('ID').Value:=StrToInt(Edit_N_zak.Text); Form_DM.ZQuery_CHEK_BL.Open;

```

```

Form_INF_ZAK.Height:=355; Form_INF_ZAK.BitBtn_add_chek.Visible:=false;
Edit_Name_bl.Text:="";
Edit_group_bl.Text:=""; Edit_calor_bl.Text:=""; Edit_cena_bl.Text:="";
Edit_massa.Text:=""; Edit_discont_bl.Text:=""; Edit_kolvo_chek.Text:="";
Edit_summa_chek.Text:="";
end;
procedure TForm_INF_ZAK.BitBtn_svernClick(Sender: TObject); begin
Form_INF_ZAK.Height:=355; end;
procedure TForm_INF_ZAK.BitBtn_RedactClick(Sender: TObject); begin
Form_INF_ZAK.Height:=355;
Form_DM.ZQuery_CHEK_BL.Close; Form_DM.ZQuery_CHEK_BL.SQL.Clear;
Form_DM.ZQuery_CHEK_BL.SQL.Add('UPDATE chek SET
chek.kolvo=:KOLVO, chek.summa=:SUMMA WHERE id_bl=:ID');
Form_DM.ZQuery_CHEK_BL.Params.ParamByName('ID').Value:=StrToInt(Edit_ID
_BL.Text);
Form_DM.ZQuery_CHEK_BL.Params.ParamByName('SUMMA').Value:=StrToInt(E
dit_summa_chek.Text);
Form_DM.ZQuery_CHEK_BL.Params.ParamByName('KOLVO').Value:=StrToInt(E
dit_kolvo_chek.Text);
Form_DM.ZQuery_CHEK_BL.ExecSQL; ShowMessage('редактирование
завершено'); Form_DM.ZQuery_CHEK_BL.Close;
Form_DM.ZQuery_CHEK_BL.SQL.Clear;
Form_DM.ZQuery_CHEK_BL.SQL.Add('SELECT spr_bl.name,spr_bl.calories,
spr_bl.cena,spr_bl.massa,spr_bl.vid, spr_bl.discont, chek.kolvo, chek.summa,
chek.id_bl FROM chek, spr_bl WHERE (chek.id_bl=spr_bl.id) and
(chek.id_zak=:ID)');
Form_DM.ZQuery_CHEK_BL.Params.ParamByName('ID').Value:=StrToInt(Edit_N
zak.Text); Form_DM.ZQuery_CHEK_BL.Open;
Form_INF_ZAK.Height:=355; Form_INF_ZAK.BitBtn_Redact.Visible:=false;
Edit_Name_bl.Text:=""; Edit_group_bl.Text:=""; Edit_calor_bl.Text:="";

```

```

Edit_cena_bl.Text:="";
Edit_massa.Text:=""; Edit_discont_bl.Text:=""; Edit_kolvo_chek.Text:="";
Edit_summa_chek.Text:=""; Edit_ID_BL.Text:="";
end;
procedure TForm_INF_ZAK.FormActivate(Sender: TObject); begin
Form_INF_ZAK.Height:=355; end;
procedure TForm_INF_ZAK.Edit_summa_chekClick(Sender: TObject); var
a,b,c,f:real; d:real; begin
if Edit_kolvo_chek.Text= " then begin
ShowMessage('Введине количество') end
else
    if Edit_discont_bl.Text = '0' then begin
        Edit_summa_chek.Text:=IntToStr(StrToInt(Edit_cena_bl.Text) *
StrToInt(Edit_kolvo_chek.Text))
end
else //Иначе begin
a:=StrToInt(Edit_cena_bl.Text); b:=strtoint(Edit_discont_bl.Text);
c:=StrToInt(Edit_kolvo_chek.Text); f:=(a-(a*b/100))*c;
Edit_summa_chek.Text:=FloatToStr(f); end;
end;
procedure TForm_INF_ZAK.Button1Click(Sender: TObject); begin
Form_DM.ZQuery_CHEK_BL.Close; Form_DM.ZQuery_CHEK_BL.SQL.Clear;
Form_DM.ZQuery_CHEK_BL.SQL.Add('Delete from chek WHERE (id_bl=:ID)
and (id_zak=:IDI)');
Form_DM.ZQuery_CHEK_BL.Params.ParamByName('ID').Value:=strtoint(Edit_ID_
BL.Text);
Form_DM.ZQuery_CHEK_BL.Params.ParamByName('IDI').Value:=strtoint(Edit_N_
zak.Text); Form_DM.ZQuery_CHEK_BL.ExecSQL;
ShowMessage('Удаление завершено'); Form_DM.ZQuery_CHEK_BL.Close;
Form_DM.ZQuery_CHEK_BL.SQL.Clear;

```

```

Form_DM.ZQuery_CHEK_BL.SQL.Add('SELECT chek.id,
spr_bl.name,spr_bl.calories, spr_bl.cena,spr_bl.massa,spr_bl.vid, spr_bl.discont,
chek.kolvo, chek.summa, chek.id_bl FROM chek, spr_bl WHERE
(chek.id_bl=spr_bl.id) and (chek.id_zak=:ID)');
Form_DM.ZQuery_CHEK_BL.Params.ParamByName('ID').Value:=StrToInt(Edit_N_
zak.text); Form_DM.ZQuery_CHEK_BL.Open;
end;
procedure TForm_INF_ZAK.Button2Click(Sender: TObject); begin
Form_report_check.Show; Form_report_check.frxReport1.ShowReport; end;
procedure TForm_INF_ZAK.Button3Click(Sender: TObject); begin
Form_report_check.frxReport1.ShowReport;
Form_report_check.frxReport1.Export(Form_report_check.frxPDFExport1); end;
procedure TForm_INF_ZAK.Button4Click(Sender: TObject); begin
Form_report_check.frxReport1.ShowReport; Form_report_check.frxReport1.Print;
end;
procedure TForm_INF_ZAK.DBGrid1CellClick(Column: TColumn); begin
Edit_id_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id_bl').AsString;
Edit_N_zak.Text:=Form_KL_ZAK.DBGrid_ZAKAZ.DataSource.DataSet.fieldbyname('id').AsString;
end;
unit Klient_zakaz;
interface uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, Mask, DBCtrls, ComCtrls, Buttons, ExtCtrls, Grids, DBGrids;
type
TForm_KL_ZAK = class(TForm) BitBtn_NEW_KL: TBitBtn;
BitBtn_REDACT_KL: TBitBtn; BitBtn_DEL_KL: TBitBtn; GroupBox3:
TGroupBox;
PageControl1: TPageControl; TabSheet1: TTabSheet; TabSheet2: TTabSheet;
DBLookupComboBox_SPR_OBJECT: TDBLookupComboBox;

```

MaskEdit\_DATE\_N: TMaskEdit;  
Edit\_zak\_dop\_inf: TEdit; BitBtn\_NEW\_ZAK: TBitBtn; Label1: TLabel;  
Label3: TLabel; Label4: TLabel; Label5: TLabel;  
GroupBox2: TGroupBox; Label\_bithdate: TLabel; Label\_nphone: TLabel;  
LabeledEdit\_Name: TLabeledEdit; LabeledEdit\_dopin: TLabeledEdit;  
MaskEdit\_bithdate: TMaskEdit; MaskEdit\_nphone: TMaskEdit;  
BitBtn\_save\_new\_fizl: TBitBtn; Label6: TLabel; DBGrid\_SPR\_CLIENT: TDBGrid;  
ComboBox\_S\_ZAK: TComboBox; DBGrid\_ZAKAZ: TDBGrid;  
BitBtn\_PERS\_ZAK: TBitBtn; BitBtn\_INF\_ZAK: TBitBtn; BitBtn\_REDACT\_ZAK:  
TBitBtn; Button2: TButton;  
Edit\_poisk: TEdit;  
DBLookupComboBox\_ORG: TDBLookupComboBox; Label7: TLabel;  
BitBtn1: TBitBtn; MaskEdit\_DATE\_O: TMaskEdit; BitBtn2: TBitBtn;  
BitBtn\_ZAKAZ\_END: TBitBtn; Edit\_N\_ZAK: TEdit;  
MaskEdit1: TMaskEdit;  
DBLookupComboBox\_PERSONA: TDBLookupComboBox; Label2: TLabel;  
BitBtn4: TBitBtn;  
Edit\_Sost: TEdit;  
MaskEdit\_d\_a\_v: TMaskEdit; BitBtn6: TBitBtn;  
Label8: TLabel;  
Label9: TLabel;  
Label10: TLabel;  
Label11: TLabel;  
Edit1: TEdit;  
CheckBox1: TCheckBox; CheckBox2: TCheckBox;  
procedure BitBtn\_NEW\_KLClick(Sender: TObject);  
procedure BitBtn\_save\_new\_fizlClick(Sender: TObject);  
procedure BitBtn\_NEW\_ZAKClick(Sender: TObject);  
procedure BitBtn\_PERS\_ZAKClick(Sender: TObject); procedure  
BitBtn\_INF\_ZAKClick(Sender: TObject); procedure

```

BitBtn_REDACT_ZAKClick(Sender: TObject); procedure Button2Click(Sender:
TObject);
procedure Button1Click(Sender: TObject);
procedure DBLookupComboBox_SPR_OBJECTEnter(Sender: TObject); procedure
Button3Click(Sender: TObject);
procedure Edit_poiskChange(Sender: TObject); procedure
BitBtn_REDACT_KLClick(Sender: TObject); procedure
BitBtn_DEL_KLClick(Sender: TObject); procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn_ZAKAZ_ENDClick(Sender: TObject); procedure
BitBtn3Click(Sender: TObject);
procedure DBLookupComboBox_PERSONAEnter(Sender: TObject); procedure
BitBtn4Click(Sender: TObject);
procedure DBGrid_ZAKAZCellClick(Column: TColumn); procedure
DBGrid_SPR_CLIENTCellClick(Column: TColumn); procedure
BitBtn5Click(Sender: TObject);
procedure DBGrid_ZAKAZDrawColumnCell(Sender: TObject; const Rect: TRect;
DataCol: Integer; Column: TColumn; State: TGridDrawState);
procedure BitBtn6Click(Sender: TObject);
procedure DBLookupComboBox_ORGEnter(Sender: TObject); procedure
CheckBox1Click(Sender: TObject);
    private
        { Private declarations } public
        { Public declarations } end;
var
    Form_KL_ZAK: TForm_KL_ZAK; implementation
uses Data_module, New_redact_fizl, INF_ZAK, REPORT_CHEK, SPR_CLIENT,
Menu;
{$R *.dfm}
procedure TForm_KL_ZAK.BitBtn_NEW_KLClick(Sender: TObject); begin

```

```

LabeledEdit_Name.Visible:=true; LabeledEdit_dopinf.Visible:=true;
MaskEdit_bithdate.Visible:=true; Label_bithdate.Visible:=true;
MaskEdit_nphone.Visible:=true; Label_nphone.Visible:=true;
BitBtn_save_new_fizl.Visible:=true; DBLookupComboBox_Org.Visible:=true;
Label6.Visible:=true;
CheckBox1.Visible:=true end;
procedure TForm_KL_ZAK.BitBtn_save_new_fizlClick(Sender: TObject); begin
if
(LabeledEdit_Name.Text='') or (MaskEdit_nphone.text='') or
(MaskEdit_bithdate.Text='') then
begin
ShowMessage('Заполните обязательные поля!'); exit;
end else
if DBLookupComboBox_ORG.enabled=true then begin
Form_DM.ZQuery_SPR_CLIENT.Close;
Form_DM.ZQuery_SPR_CLIENT.SQL.Clear;
Form_DM.ZQuery_SPR_CLIENT.SQL.Add('INSERT INTO spr_client (name,
dopinf,nphone,bithdate,name_org)');
Form_DM.ZQuery_SPR_CLIENT.SQL.Add('VALUES (:Param1, :Param2, :Param3,
:Param4,
:Param5)');
Form_DM.ZQuery_SPR_CLIENT.Params.ParamByName('Param1').Value:=LabeledE
dit_Name.T ext;
Form_DM.ZQuery_SPR_CLIENT.Params.ParamByName('Param2').Value:=LabeledE
dit_dopinf.T ext;
Form_DM.ZQuery_SPR_CLIENT.Params.ParamByName('Param3').Value:=MaskEdi
t_nphone.Te xt;
Form_DM.ZQuery_SPR_CLIENT.Params.ParamByName('Param4').Value:=StrToDat
e(MaskEdit_bithdate.Text);
Form_DM.ZQuery_SPR_CLIENT.Params.ParamByName('Param5').Value:=DBLook

```

```

upComboBox_ORG.KeyValue;
Form_DM.ZQuery_SPR_CLIENT.ExecSQL; ShowMessage('Добавление
завершено'); Form_DM.ZQuery_SPR_CLIENT.Close;
Form_DM.ZQuery_SPR_CLIENT.SQL.Clear;
Form_DM.ZQuery_SPR_CLIENT.SQL.Add('select * from spr_client');
Form_DM.ZQuery_SPR_CLIENT.Open; DBLookupComboBox_Org.Keyvalue:=";
LabeledEdit_Name.Text:=";
MaskEdit_bithdate.Text:="; MaskEdit_nphone.Text:="; LabeledEdit_dopinf.Text:=";
LabeledEdit_Name.Visible:=false; LabeledEdit_dopinf.Visible:=false;
MaskEdit_bithdate.Visible:=false; Label_bithdate.Visible:=false;
MaskEdit_nphone.Visible:=false; Label_nphone.Visible:=false;
BitBtn_save_new_fizl.Visible:=false; DBLookupComboBox_Org.Visible:=false;
Label6.Visible:=false; CheckBox1.Visible:=false;
end else
if DBLookupComboBox_ORG.enabled=false then begin
Form_DM.ZQuery_SPR_CLIENT.Close;
Form_DM.ZQuery_SPR_CLIENT.SQL.Clear;
Form_DM.ZQuery_SPR_CLIENT.SQL.Add('INSERT INTO spr_client (name,
dopinf,nphone,bithdate)');
Form_DM.ZQuery_SPR_CLIENT.SQL.Add('VALUES (:Param1, :Param2, :Param3,
:Param4)');
Form_DM.ZQuery_SPR_CLIENT.Params.ParamByName('Param1').Value:=LabeledE
dit_Name.T ext;
Form_DM.ZQuery_SPR_CLIENT.Params.ParamByName('Param2').Value:=LabeledE
dit_dopinf.T ext;
Form_DM.ZQuery_SPR_CLIENT.Params.ParamByName('Param3').Value:=MaskEdi
t_nphone.Te xt;
Form_DM.ZQuery_SPR_CLIENT.Params.ParamByName('Param4').Value:=StrToDat
e(MaskEdit_bithdate.Text);
//Form_DM.ZQuery_SPR_CLIENT.Params.ParamByName('Param5').Value:=DBLoo

```

```

kupComboBox_Org.KeyValue;
Form_DM.ZQuery_SPR_CLIENT.ExecSQL; ShowMessage('Добавление
завершено'); Form_DM.ZQuery_SPR_CLIENT.Close;
Form_DM.ZQuery_SPR_CLIENT.SQL.Clear;
Form_DM.ZQuery_SPR_CLIENT.SQL.Add('select * from spr_client');
Form_DM.ZQuery_SPR_CLIENT.Open;
//DBLookupComboBox_Org.Keyvalue:="; LabeledEdit_Name.Text=";
MaskEdit_bithdate.Text="; MaskEdit_nphone.Text="; LabeledEdit_dopinf.Text=";
LabeledEdit_Name.Visible:=false; LabeledEdit_dopinf.Visible:=false;
MaskEdit_bithdate.Visible:=false; Label_bithdate.Visible:=false;
MaskEdit_nphone.Visible:=false; Label_nphone.Visible:=false;
BitBtn_save_new_fizl.Visible:=false;
DBLookupComboBox_Org.Visible:=false; Label6.Visible:=false;
CheckBox1.Visible:=false;
end; end;

procedure TForm_KL_ZAK.BitBtn_NEW_ZAKClick(Sender: TObject); begin
Form_DM.ZQuery_NEW_ZAK.Close; Form_DM.ZQuery_NEW_ZAK.SQL.Clear;
Form_DM.ZQuery_NEW_ZAK.SQL.Add('INSERT INTO zakaz (id_kl,date_n,
dop_inf,name_obj,priznak,persona)');
Form_DM.ZQuery_NEW_ZAK.SQL.Add('VALUES (:Param1, :Param2, :Param3,
:Param4,
:Param5, :Param6)');
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param1').Value:=DBGrid_S
PR_CLIENT.DataSource.DataSet.fieldbyname('id_kl_f').AsString;
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param2').Value:=StrToDateT
ime(MaskEdit_DATE_N.Text);
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param3').Value:=Edit_zak_d
op_inf.Text;
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param4').Value:=DBLookup
ComboBox_SPR_OBJECT.KeyValue;

```

```

Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param5').Value:=ComboBox
_S_ZAK.Text;
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param6').Value:=DBLookup
ComboBox_PERSONA.KeyValue;
Form_DM.ZQuery_NEW_ZAK.ExecSQL; ShowMessage('Добавление завершено');
PageControl1.ActivePageIndex := 1;
Form_DM.ZQuery_SPR_OBJECT.Close;
Form_DM.ZQuery_SPR_OBJECT.SQL.Clear;
Form_DM.ZQuery_SPR_OBJECT.SQL.Add('UPDATE spr_object SET
date_na=:DATEN, id_kl=:IDK where NAME=:NAME');
Form_DM.ZQuery_SPR_OBJECT.Params.ParamByName('DATEN').Value:=StrToD
ateTime(MaskEdit_DATE_N.Text);
Form_DM.ZQuery_SPR_OBJECT.Params.ParamByName('IDK').Value:=DBGrid_SP
R_CLIENT.DataSource.DataSet.fieldbyname('id_kl_f').AsString;;
Form_DM.ZQuery_SPR_OBJECT.Params.ParamByName('NAME').Value:=DBLook
upComboBox_SPR_OBJECT.KeyValue;
Form_DM.ZQuery_SPR_OBJECT.ExecSQL;
Form_DM.ZQuery_SPR_OBJECT.Close;
Form_DM.ZQuery_SPR_OBJECT.SQL.Clear;
Form_DM.ZQuery_SPR_OBJECT.SQL.Add('SELECT * from spr_object');
Form_DM.ZQuery_SPR_OBJECT.Open;
Edit_zak_dop_inf.Text:="";
if label11.Caption='админ' then begin
DBLookupComboBox_PERSONA.Keyvalue:="";
Form_DM.ZQuery_NEW_ZAK.Close; Form_DM.ZQuery_NEW_ZAK.SQL.Clear;
Form_DM.ZQuery_NEW_ZAK.SQL.Add('SELECT * from zakaz');
Form_DM.ZQuery_NEW_ZAK.Open;
end else
if label11.Caption='официант' then begin
DBLookupComboBox_PERSONA.Keyvalue:=Label8.Caption;

```

```

Form_DM.ZQuery_NEW_ZAK.Close; Form_DM.ZQuery_NEW_ZAK.SQL.Clear;
Form_DM.ZQuery_NEW_ZAK.SQL.Add('SELECT * from zakaz where
persona=:PERS ');
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('PERS').Value:=Label8.Capti
on; Form_DM.ZQuery_NEW_ZAK.Open;
end;
PageControl1.ActivePageIndex := 1; end;
procedure TForm_KL_ZAK.BitBtn_PERS_ZAKClick(Sender: TObject); begin
if (Label11.Caption='админ') and (CheckBox2.Checked=false) then begin
Form_DM.ZQuery_NEW_ZAK.Close; Form_DM.ZQuery_NEW_ZAK.SQL.Clear;
Form_DM.ZQuery_NEW_ZAK.SQL.Add('SELECT zakaz.id_kl, zakaz.id,
zakaz.date_n,zakaz.date_o, zakaz.name_obj,persona, zakaz.priznak from zakaz
where zakaz.id_kl=:ID');
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('ID').Value:=DBGrid_SPR_C
LIENT.Data Source.DataSet.fieldbyname('id_kl_f').AsString;
Form_DM.ZQuery_NEW_ZAK.Open; end
else
if (Label11.Caption='официант') and (CheckBox2.Checked=false) then
begin Form_DM.ZQuery_NEW_ZAK.Close;
Form_DM.ZQuery_NEW_ZAK.SQL.Clear;
Form_DM.ZQuery_NEW_ZAK.SQL.Add('SELECT zakaz.id_kl, zakaz.id,
zakaz.date_n,zakaz.date_o, zakaz.name_obj,persona, zakaz.priznak from zakaz
where zakaz.id_kl=:ID and persona=:PERS');
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('ID').Value:=DBGrid_SPR_C
LIENT.Data Source.DataSet.fieldbyname('id_kl_f').AsString;
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('PERS').Value:=Label8.Capti
on; Form_DM.ZQuery_NEW_ZAK.Open;
end else
if (Label11.Caption='админ')and (CheckBox2.Checked=true) then begin
Form_DM.ZQuery_NEW_ZAK.Close;

```

```

Form_DM.ZQuery_NEW_ZAK.SQL.Clear;
Form_DM.ZQuery_NEW_ZAK.SQL.Add('SELECT zakaz.id_kl, zakaz.id,
zakaz.date_n,zakaz.date_o, zakaz.name_obj,persona, zakaz.priznak from zakaz
where zakaz.id_kl=:ID and priznak=:PRI');
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('ID').Value:=DBGrid_SPR_C
LIENT.Data Source.DataSet.fieldbyname('id_kl_f').AsString;
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('PRI').Value:='новый';
Form_DM.ZQuery_NEW_ZAK.Open;
end else
if (Label11.Caption='официант')and (CheckBox2.Checked=true) then
begin Form_DM.ZQuery_NEW_ZAK.Close;
Form_DM.ZQuery_NEW_ZAK.SQL.Clear;
Form_DM.ZQuery_NEW_ZAK.SQL.Add('SELECT zakaz.id_kl, zakaz.id,
zakaz.date_n,zakaz.date_o, zakaz.name_obj,persona, zakaz.priznak from zakaz
where zakaz.id_kl=:ID and persona=:PERS and priznak=:PRI');
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('ID').Value:=DBGrid_SPR_C
LIENT.Data Source.DataSet.fieldbyname('id_kl_f').AsString;
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('PERS').Value:=Label8.Capti
on; Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('PRI').Value:='новый';
Form_DM.ZQuery_NEW_ZAK.Open;
end; end;
procedure TForm_KL_ZAK.BitBtn_INF_ZAKClick(Sender: TObject); begin
if
Edit_Sost.Text='новый' then begin Form_INF_ZAK.Show;
Form_DM.ZQuery_CHEK_BL.Close; Form_DM.ZQuery_CHEK_BL.SQL.Clear;
Form_DM.ZQuery_CHEK_BL.SQL.Add('SELECT chek.id, chek.id_bl, chek.id_zak,
spr_bl.name,spr_bl.calories, spr_bl.cena,spr_bl.massa,spr_bl.vid, spr_bl.discont,
chek.kolvo, chek.summa FROM chek, spr_bl WHERE (chek.id_bl=spr_bl.id) and
(chek.id_zak=:ID)');
Form_DM.ZQuery_CHEK_BL.Params.ParamByName('ID').Value:=Form_KL_ZAK.

```

```

DBGrid_ZAKAZ.DataSource.DataSet.fieldbyname('id').AsString;;
Form_DM.ZQuery_CHEK_BL.Open; end
else
ShowMessage('заказ закрыт'); end;
procedure TForm_KL_ZAK.BitBtn_REDACT_ZAKClick(Sender: TObject); begin
if
Edit_Sost.Text='закрыт' then begin
Form_DM.ZQuery_write.Close; Form_DM.ZQuery_write.SQL.Clear;
Form_DM.ZQuery_write.SQL.Add('select chek.id_zak, spr_bl.vid, spr_bl.name,
SUM(chek.kolvo), spr_bl.cena, spr_bl.discont, SUM(chek.summa) from chek, spr_bl
WHERE (chek.id_bl=spr_bl.id) and (chek.id_zak=:ID) group by spr_bl.name');
Form_DM.ZQuery_write.Params.ParamByName('ID').Value:=Form_KL_ZAK.DBGrid_ZAKAZ.
DataSource.DataSet.fieldbyname('id').AsString;;
Form_DM.ZQuery_write.Open; Form_DM.ZQuery_CHEK_P.Close;
Form_DM.ZQuery_CHEK_P.SQL.Clear;
Form_DM.ZQuery_CHEK_P.SQL.Add('SELECT sum(summa) FROM chek
WHERE (chek.id_zak=:ID)');
Form_DM.ZQuery_CHEK_P.Params.ParamByName('ID').Value:=Form_KL_ZAK.D
BGrid_ZAKAZ.DataSource.DataSet.fieldbyname('id').AsString;;
Form_DM.ZQuery_CHEK_P.Open; Form_report_check.frxReport1.ShowReport;
Form_report_check.frxReport1.Print;
end else
ShowMessage('закройте заказ'); end;
procedure TForm_KL_ZAK.Button2Click(Sender: TObject); begin
Form_DM.ZQuery_write.Close; Form_DM.ZQuery_write.SQL.Clear;
Form_DM.ZQuery_write.SQL.Add('select chek.id_zak, spr_bl.vid, spr_bl.name,
SUM(chek.kolvo), spr_bl.cena, spr_bl.discont, SUM(chek.summa) from chek, spr_bl
WHERE (chek.id_bl=spr_bl.id) and (chek.id_zak=:ID) group by spr_bl.name');
Form_DM.ZQuery_write.Params.ParamByName('ID').Value:=Form_KL_ZAK.DBGrid
_ZAKAZ. DataSource.DataSet.fieldbyname('id').AsString;;

```

```

Form_DM.ZQuery_write.Open;
Form_DM.ZQuery_CHEK_P.Close; Form_DM.ZQuery_CHEK_P.SQL.Clear;
Form_DM.ZQuery_CHEK_P.SQL.Add('SELECT sum(summa) FROM chek
WHERE (chek.id_zak=:ID)');
Form_DM.ZQuery_CHEK_P.Params.ParamByName('ID').Value:=Form_KL_ZAK.D
BGrid_ZAK AZ.DataSource.DataSet.fieldbyname('id').AsString;;
Form_DM.ZQuery_CHEK_P.Open; Form_report_check.Show;
Form_report_check.frxReport1.ShowReport;
end;

procedure TForm_KL_ZAK.Button1Click(Sender: TObject);
begin Form_DM.ZQuery_SPR_OBJECT.Close;
Form_DM.ZQuery_SPR_OBJECT.SQL.Clear;
Form_DM.ZQuery_SPR_OBJECT.SQL.Add('SELECT * from spr_object where
(date_o_a is NULL) or (date_o_a<:DATEO)');
Form_DM.ZQuery_SPR_OBJECT.Params.ParamByName('DATEO').Value:=StrToD
ateTime(Mas kEdit_DATE_N.Text);
Form_DM.ZQuery_SPR_OBJECT.Open; end;

procedure TForm_KL_ZAK.DBLookupComboBox_SPR_OBJECTEnter(Sender:
TObject); begin
Form_DM.ZQuery_SPR_OBJECT.Close;
Form_DM.ZQuery_SPR_OBJECT.SQL.Clear;
Form_DM.ZQuery_SPR_OBJECT.SQL.Add('SELECT spr_object.name from
spr_object where date_n_a is null');
Form_DM.ZQuery_SPR_OBJECT.Open; end;

procedure TForm_KL_ZAK.Button3Click(Sender: TObject); begin
Form_DM.ZQuery_NEW_ZAK.Close; Form_DM.ZQuery_NEW_ZAK.SQL.Clear;
Form_DM.ZQuery_NEW_ZAK.SQL.Add('INSERT INTO zakaz (id_kl,date_n,
date_o, dop_inf,name_obj,priznak)');
Form_DM.ZQuery_NEW_ZAK.SQL.Add('VALUES (:Param1, :Param2, :Param3,
:Param4,

```

```

:Param5, :Param6)');
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param1').Value:=DBGrid_S
PR_CLIENT.DataSource.DataSet.fieldbyname('id_kl_f').AsString;
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param2').Value:=StrToDateT
ime(MaskEdit_DATE_N.Text);
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param3').Value:=StrToDateT
ime(MaskEdit_DATE_o.Text);
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param4').Value:=Edit_zak_d
op_inf.Text;
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param5').Value:=DBLookup
ComboBox_SPR_OBJECT.KeyValue;
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('Param6').Value:=ComboBox
_S_ZAK.Text;
Form_DM.ZQuery_NEW_ZAK.ExecSQL; Form_DM.ZQuery_NEW_ZAK.Close;
Form_DM.ZQuery_NEW_ZAK.SQL.Clear;
Form_DM.ZQuery_NEW_ZAK.SQL.Add('SELECT zakaz.id_kl, zakaz.id,
zakaz.date_n, zakaz.date_o, spr_object.name, zakaz.priznak from zakaz, spr_object
where zakaz.name_obj=spr_object.name and (zakaz.id_kl=:ID)');
Form_DM.ZQuery_NEW_ZAK.Params.ParamByName('ID').Value:=DBGrid_SPR_C
LIENT.DataSource.DataSet.fieldbyname('id_kl_f').AsString;
Form_DM.ZQuery_NEW_ZAK.Open; ShowMessage('Добавление завершено'); end;
procedure TForm_KL_ZAK.Edit_poiskChange(Sender: TObject)
Form_INF_ZAK.Edit_Name_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('na
me').AsString
;
Form_INF_ZAK.Edit_calor_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('calo
ries').AsString;
Form_INF_ZAK.Edit_massa.Text:=DBGrid1.DataSource.DataSet.fieldbyname('massa
').AsString;
Form_INF_ZAK.Edit_group_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('vid

```

```
').AsString;  
Form_INF_ZAK.Edit_discont_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('discont').AsString;  
Form_INF_ZAK.Edit_cena_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('cena').AsString;  
Form_INF_ZAK.Edit_N_zak.Text:=Form_KL_ZAK.DBGrid_ZAKAZ.DataSource.DataSet.fieldbyname('id').AsString;  
Form_INF_ZAK.Edit_id_bl.Text:=DBGrid1.DataSource.DataSet.fieldbyname('id').AsString;  
Edit_poisk.Text:=""; Form_POISK_BL.Close; Form_INF_ZAK.Height:=622;  
Form_INF_ZAK.Show;  
Form_INF_ZAK.BitBtn_add_chek.Visible:=true;
```

