

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

Институт **математики, физики и информационных технологий**
Кафедра «**Прикладная математика и информатика**»

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В СОЦИАЛЬНОЙ СФЕРЕ

БАКАЛАВРСКАЯ РАБОТА

на тему **Автоматизация деятельности отдела информационных технологий ООО "Золотой берег"**

Студент	_____	М.А. Фрейдлин	_____
Руководитель	_____	Т.Г. Султанов	_____
Консультант по аннотации	_____	Н.В.Ященко	_____

Допустить к защите

Заведующий кафедрой к.тех.н., доцент А.В. Очеповский _____

« _____ » _____ 20__ г.

Тольятти 2017

АННОТАЦИЯ

к выпускной квалификационной работе

Тема: «Автоматизация деятельности отдела информационных технологии ООО "Золотой берег"»

Цель выпускной квалификационной работы – разработка автоматизированной информационной системы отдела информационных технологий ООО «Золотой берег».

Объектом исследования являются бизнес-процессы инвентаризации и сопровождения аппаратного и программного обеспечения в отделе информационных технологий ООО «Золотой берег».

Предметом исследования является информационная система (ИС) отдела информационных технологий ООО «Золотой берег»..

Методы решения задачи – структурное и объектно-ориентированное моделирование информационных систем.

В первой главе был произведен анализ деятельности компании, выделены основные процессы, которые требовали автоматизации. Сформированы требования к сайту компании. Были созданы схемы бизнес-процессов .

На стадии проектирования были построены диаграммы на основе языка UML, демонстрирующие роли пользователей внутри системы, а так же их связи. Была выбрана трехзвенная архитектура информационной системы с Web-сервером Apache, СУБД MySQL .Так же, была построена логическая модель базы данных.

В третьей главе описана итоговая информационная система с указанием основных функций, информация о тестировании системы перед её вводом в эксплуатацию, а так же примеры создания основных компонентов при помощи FoxPro.

В работе использовалось 3 таблицы, 25 рисунков, список использованной литературы содержит 15 источников, из них 5 – иностранная литература. Общий объем работы составляет 65 страниц.

ABSTRACT

The theme of the graduation project is «Automation of the OOO "Zolotoy bereg" IT Department activities »

The objective of the graduation project is to develop the informational system (IT system) for an Information Technology Department (IT Department) of OOO "Zolotoy bereg".

The object of the research are the business processes of inventorying and maintaining the hardware and software in the IT Department of OOO «Zolotoy bereg».

The subject of the study is the IT system of the OOO "Zolotoy bereg" IT Department.

The problem-solving technique is the structural and object-oriented information systems modeling.

The first chapter analyses of the company's activities and highlights the main processes that required automation. Information system requirements have been established. Business process diagrams have been made.

In the design phase, UML-based diagrams were made to demonstrate the roles of users within the system and their connections. The three-tiered architecture information system with the Apache Web server, DBMS MySQL has been selected. Also, a logical database model was made.

The third chapter describes the final information system, including the main functions, and information about testing the system before its putting into operation, and examples of creating basic components with Microsoft Visual FoxPro.

The graduation project consists of introduction, includes 25 figures, 3 tables, the list of 15 references including 5 foreign sources. The total amount of the work is 65 pages.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ АВТОМАТИЗАЦИИ И РАЗРАБОТКА ТРЕБОВАНИЙ К АИС.....	7
1.1 Характеристика предприятия ООО «Золотой берег»	7
1.2 Анализ бизнес-процессов ИТ-отдела.....	9
1.3 Выбор технологии концептуального моделирования предметной области	10
1.4 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»	11
1.5 Анализ существующих программных разработок для автоматизации ИТ-отдела.....	16
1.6 Постановка задачи на разработку автоматизированной информационной системы	20
1.7 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»	21
Выводы к 1-й главе	24
ГЛАВА 2 ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	25
2.1 Обоснование модели данных.....	25
2.3 Разработка логической модели АИС	26
2.3.1 Диаграмма вариантов использования АИС	27
2.3.2 Диаграмма классов АИС	30
2.3.3 Требования к аппаратно-программному обеспечению автоматизированной информационной системы.....	33
2.4 Разработка логической модели данных АИС	34
Выводы ко 2-й главе	38
ГЛАВА 3 ФИЗИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОЕКТНЫХ РЕШЕНИЙ ПО АВТОМАТИЗАЦИИ.....	39
3.1 Обоснование выбора средств для реализации АИС.....	39
3.1.1 Разработка типа архитектуры АИС	39
3.1.2 Выбор системы управления базами данных	42
3.1.3 Выбор среды разработки программного обеспечения.....	44

3.2. Разработка физической модели данных автоматизированной информационной системы	46
3.3. Структурная схема программного обеспечения АИС	48
Выводы к 3-й главе	59
ЗАКЛЮЧЕНИЕ	60
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	62
ПРИЛОЖЕНИЕ А	64

ВВЕДЕНИЕ

В условиях финансовых трудностей и высокой конкуренции между предприятиями сложно добиться успеха без развитой и гибкой, а также масштабируемой инфраструктуры информационных технологий (ИТ), которая представляет из себя совокупность аппаратного и программного обеспечения, способную обеспечивать и предоставлять информационную поддержку основных бизнес-процессов предприятия.

В связи с этим возрастает потребность каждого предприятия в создании собственной ИТ - службы, которая помимо задач по внедрению средств вычислительной техники (ВТ) и программного обеспечения, выполняет обязанности по сопровождению ИТ- инфраструктуры и оказанию определенного спектра услуг подразделениям предприятия, которые оформляются в виде заявок.

Как показывает практика, решение этих задач возможно только при организации в ИТ - отделе предприятия собственной автоматизированной системы, ориентированной на специфику деятельности данного отдела.

Все вышесказанное обуславливает **актуальность** темы дипломной работы.

Объект исследования являются бизнес-процессы инвентаризации и сопровождения аппаратного и программного обеспечения в отделе информационных технологий ООО «Золотой берег».

Предметом исследования является автоматизированная информационная система (АИС) отдела информационных технологий ООО «Золотой берег».

Область применения - информационно-технические подразделения ООО «Золотой берег» которое выполняет функцию учета оборудования и внедрением современных технологий на предприятии.

Цель работы - разработка АИС отдела информационных технологий ООО «Золотой берег».

Для достижения поставленной цели в работе решаются следующие **задачи**:

- изучение предметной области и разработка ее концептуальной модели;
- анализ существующих решений на рынке;
- обоснование выбора технологии разработки;
- обоснование выбора СУБД;
- проектирование АИС ИТ - отдела;
- реализация АИС ИТ - отдела.

Работа состоит из введения, трех глав, заключения, списка использованных источников и литературы и приложений.

Во введении данной работы обосновывается значимость и актуальность выбранной темы.

В первой главе приводится анализ предметной области с указанием актуальности разработки АИС. Была выбрана технология для концептуального моделирования, проделана работа по созданию и анализу модели бизнес процесса «КАК ЕСТЬ», после чего следовательно сделан вывод что некоторая часть процессов нуждается в автоматизации. Кроме этого, был проведен анализ существующих информационных систем для инвентаризации и учета оборудования, и программного продукта, на основе всех данных и сравнительного анализа существующих автоматических информационных систем, после чего был сделан вывод что необходима разработка новой информационной системы. На основе этих данных была поставлена задача по разработки модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ».

Во второй главе описана логическое проектирование, разработка и описание логической моделей объекта. Была выбрана технология

логического моделирования, в дальнейшем построена логическая модель автоматизированной информационной системы.

Третья глава, представляет из себя описание физического проектирования автоматизированной информационной системы. Была выбрана архитектура информационной системы, и технология разработки программного приложения, после чего была выбрана СУБД и выполнена разработка физической модели данных под выбранную СУБД, рассмотрена часть функционала разработанной информационной системы. В заключении было проведено тестирование разработанной информационной системы.

В приложении представлены иллюстрации отдельных положений исследуемой темы и результаты проектирования.

В заключении сформулированы основные выводы, которые были сделаны в процессе проведения дипломного исследования и описаны результаты практической реализации дипломной работы.

ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ АВТОМАТИЗАЦИИ И РАЗРАБОТКА ТРЕБОВАНИЙ К АИС

1.1 Характеристика предприятия ООО «Золотой берег»

ООО «Золотой берег» представляет из себя коммерческую организацию, которая реализует кондитерские изделия, орехи, сухофрукты с 1996 года.

Деятельностью ООО «Золотой берег» является оптовая, розничная и региональные продажи кондитерских изделий, фасовка для дальнейшей продажи.

На рис. 1.1 предоставлена организационная структура ООО «Золотой берег»:

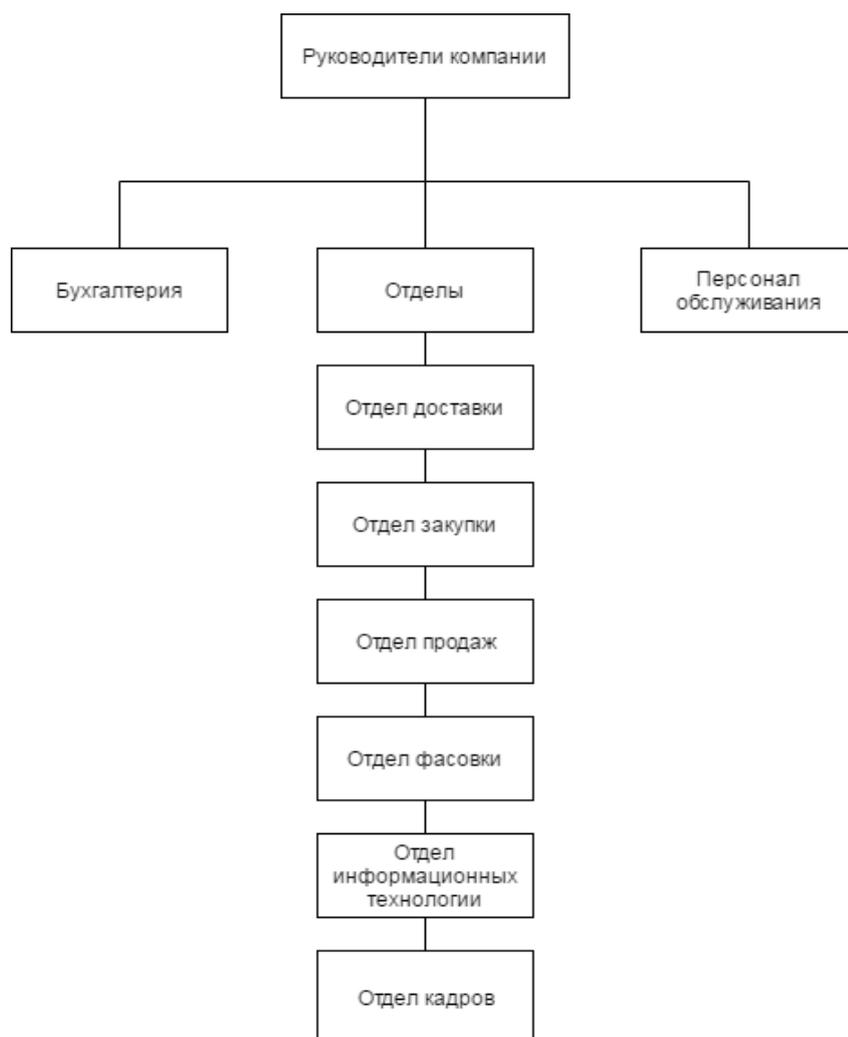


Рис. 1.1 Структура организации ООО «Золотой берег»

Основные структурные подразделения:

- отдел продаж: поиск и привлечение новых клиентуры, своевременное принятие заявок от клиента;
- отдел закупок: поиск наиболее выгодных поставщиков и товаров, а так же своевременный заказ товара;
- отдел доставки: своевременная отгрузка товара клиенту, контроль сроков и качества доставки;
- отдел фасовки: фасовка товара;
- бухгалтерия: ведение и сдача бухгалтерской отчетности, оплата счетов,;
- отдел кадров: управление кадрами (оформление сотрудников);
- отдел ИТ: обеспечение информационной безопасности, целостности данных и обеспечение бесперебойной работы оборудования;
- управление организацией и бизнес процессами;

Главным органом управления организации является генеральный директор.

В организации автоматизированы следующие процессы:

- управление производством;
- бухучет;
- управление планированием;
- управление финансами;
- управление продажами;
- управление закупками;
- логистика.

Аппаратная часть ИТ - инфраструктуры предприятия включает в себя следующее оборудование: локальные сети, состоящие из компьютеров-клиентов и серверов, лазерные принтеры, сканеры, ксероксы, цифровая мини-АТС.

1.2 Анализ бизнес-процессов ИТ-отдела

Отдел ИТ является структурным подразделением, в своей деятельности подчиняется Руководителю отдела ИТ .

Основными задачами отдела ИТ являются:

- реализация концепции развития ИТ- инфраструктуры;
- обеспечение бесперебойного функционирования аппаратно-программных комплексов;
- обеспечение информационной и технической поддержки средств вычислительной техники и программного обеспечения и средств связи;
- проведение работ по оптимизации информационно-технических ресурсов с целью максимальной их эффективности в эксплуатации;
- контроль за лицензированием программного обеспечения;
- выполнять надзор за исполнением нормативных документов по правилам работы с вычислительной техникой и офисным оборудованием и др.

Задания на сопровождение ИТ-инфраструктуры в ИТ-отдел поступают в форме заявки, которая представляет из себя бланк, который содержащие следующие данные:

- дата;
- заявитель (отдел, ФИО);
- описание работ (закупка либо ремонт оборудования);
- исполнитель;
- конечный результат;
- дата выполнения.

При выполнении заявки сотрудник ИТ-отдела опираются на нормативные документы, которые касаются технической эксплуатации и ремонта вычислительной и оргтехники, а также эксплуатации и сопровождения ПО в организации.

Бизнес-процесс выполнения заявок на обслуживание ИТ - инфраструктуры не автоматизирован.

1.3 Выбор технологии концептуального моделирования предметной области

Моделирование предметной области является одним из важных этапов на этапе когда проектируется информационная система. Для этого нужно провести сравнительный анализ известных нотации концептуального моделирования информационных систем:UML,IDEF0.

UML(United Modeling Language)-представляет из себя язык графического описания предназначенный для объектно-ориентированного моделирования бизнес-процессов для программных систем. Используется для создания абстрактной модели системы, которая носит название UMLмодель.

IDEF0(Integration Definition for Function Modeling)-методология функционального моделирования бизнес-процессов.IDEF0 показывает структуру и функции системы, а также информационные потоки которые связаны с этими функциями.

Результат анализа методологии представлен в таблице 1.1.

Таблица 1.1-Сравнительный анализ методологий

Критерий сравнения	Нотация IDEF0	Нотация UML
Легкость изучения	Легко	Средне
Удобство по созданию моделей	Легко	Сложно
Подход к проектированию	Функциональный	Объектно-ориентированный
Система хранения данных модели	Модели хранятся в файлах	Модели хранятся в файлах
Возможность декомпозиции	Неограниченная	Неограниченная

Исходя из результатов таблицы 1.1, можно сделать вывод, что IDEF0 является наиболее удобной и подходящей нотации, по причине ее легкого изучения и удобстве создания моделей, в сравнении с UML ,которая в свою

очередь более сложна в изучении, имеет некоторую специфику при создании модели, также модели UML более трудные для изучения и восприятия.

После выбора технологии концептуального моделирования, перейдем к разработке модели бизнес-процесса «КАК ЕСТЬ»

1.4 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»

Бизнес-процесс-это совокупность всех действия, направленных на создание определенного программного продукта либо информационной системы. Для наглядности бизнес-процессы часто показываются в виде блок-схем бизнес-процессов.

Модель бизнес-процесса «КАК ЕСТЬ» является существующей моделью состояния организации. Данная модель отображает процессы происходящие в ней прямо сейчас, а также используемые информационные объекты. Такую модель еще называют функциональной и делают при помощи различных CASE-средств с использованием различных графических нотаций. На этапе создания модели бизнес-процесса «КАК ЕСТЬ» важно создать максимально похожую модель, которая основана на реальных потоках процессов.

По результатам проведённого анализа бизнес-процесса оформления заявки на услугу построены диаграммы, которые предоставляют собой модель соответствующего бизнес-процесса «КАК ЕСТЬ» рис (1.2).

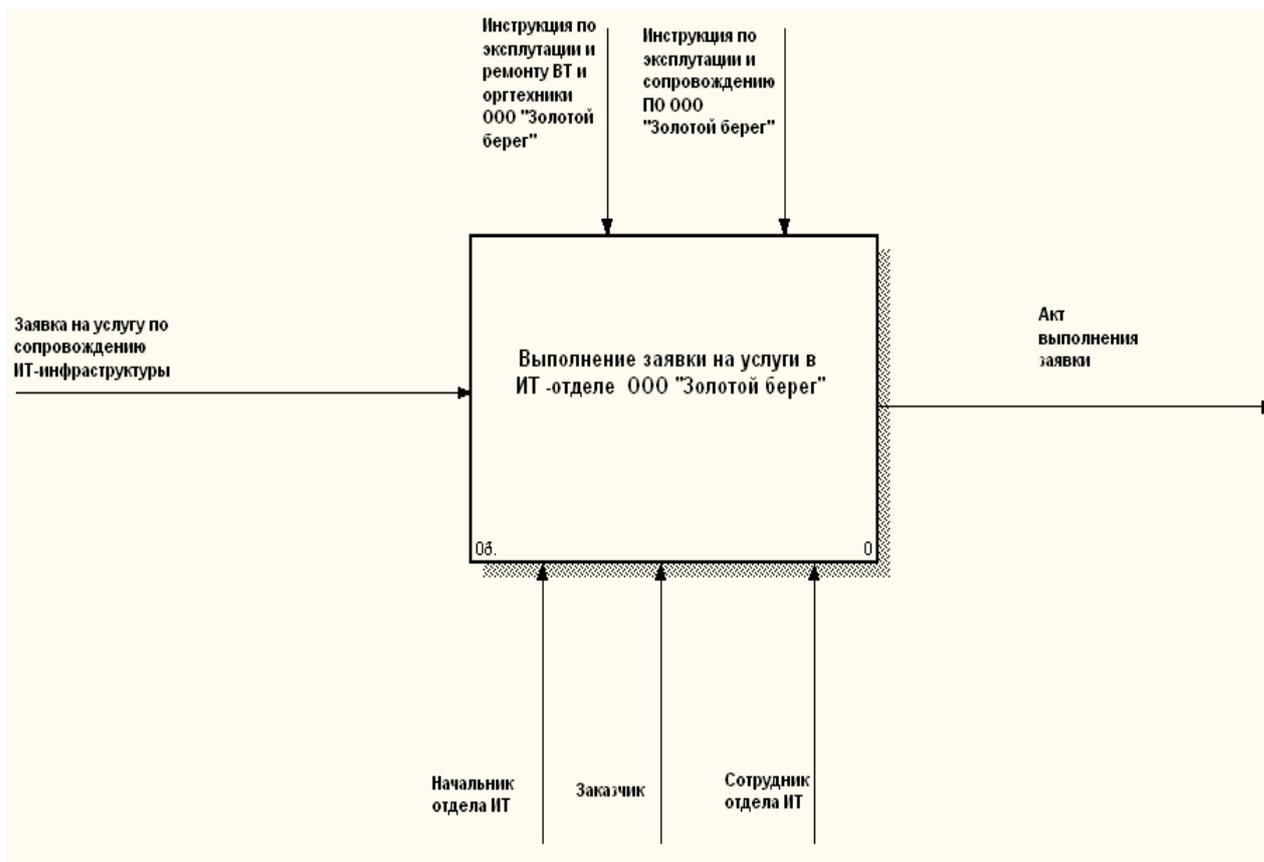


Рис. 1.2 Контекстная диаграмма (IDEF0) бизнес-процесса выполнения заявок на услуги по сопровождению ИТ-инфраструктуры ООО «Золотой берег» «КАК ЕСТЬ».

Для лучше понимания как происходят процессы рассмотрим рисунок 1.3, где происходит декомпозиция бизнес-процесса выполнения заявок на услуги по сопровождению ИТ-инфраструктуры ООО «Золотой берег» «КАК ЕСТЬ».

Бизнес-процесс оформления заявки на услугу по сопровождению ИТ инфраструктуры состоит из следующих операций:

- заказчик заполняет бланк заявки с указанием необходимых атрибутов;
- начальник ИТ - отдела рассматривает и визирует заявку, а затем передает ее ответственному лицу;

- ответственный за выполнение заявки сотрудник ИТ - отдела регистрирует заявку в журнале заявок и приступает к ее выполнению;
- результат работы акцептируется заказчиком
- сотрудник ИТ-отдела и заказчик формируют акт выполнения заявки.

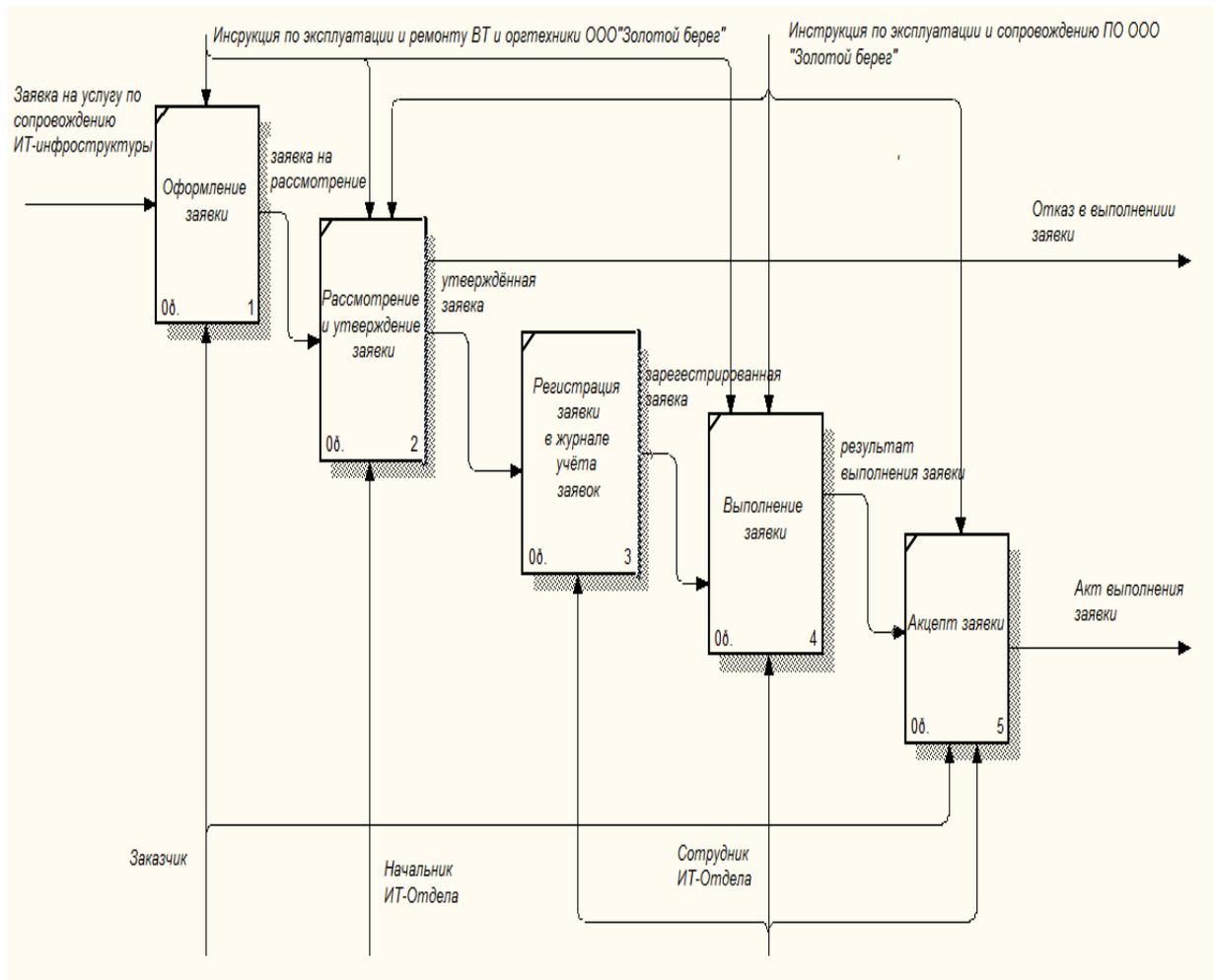


Рис. 1.3 Декомпозиция бизнес-процесса выполнения заявок на услуги по сопровождению ИТ - инфраструктуры ООО «Золотой берег» «КАК ЕСТЬ».

На рисунке 1.4 показана декомпозиция бизнес-процесса регистрации заявки в журнале учёта заявок «КАК ЕСТЬ», где:

- Сотрудник либо же начальник ИТ-отдела получает утверждённую заявку;

- Ответственный за выполнение заявки сотрудник либо начальник ИТ - отдела заносит данные заявки в журнал, потом назначается исполнителя в лице сотрудника ИТ-отдела, назначает сроки
- Назначается исполнитель в лице сотрудника ИТ-отдела.
- Происходит назначение сроков на выполнения поставленных задач или заказов.
- В конце происходит регистрация заполненной заявки с исходными данными и лицом, которой выполняет эту работу.

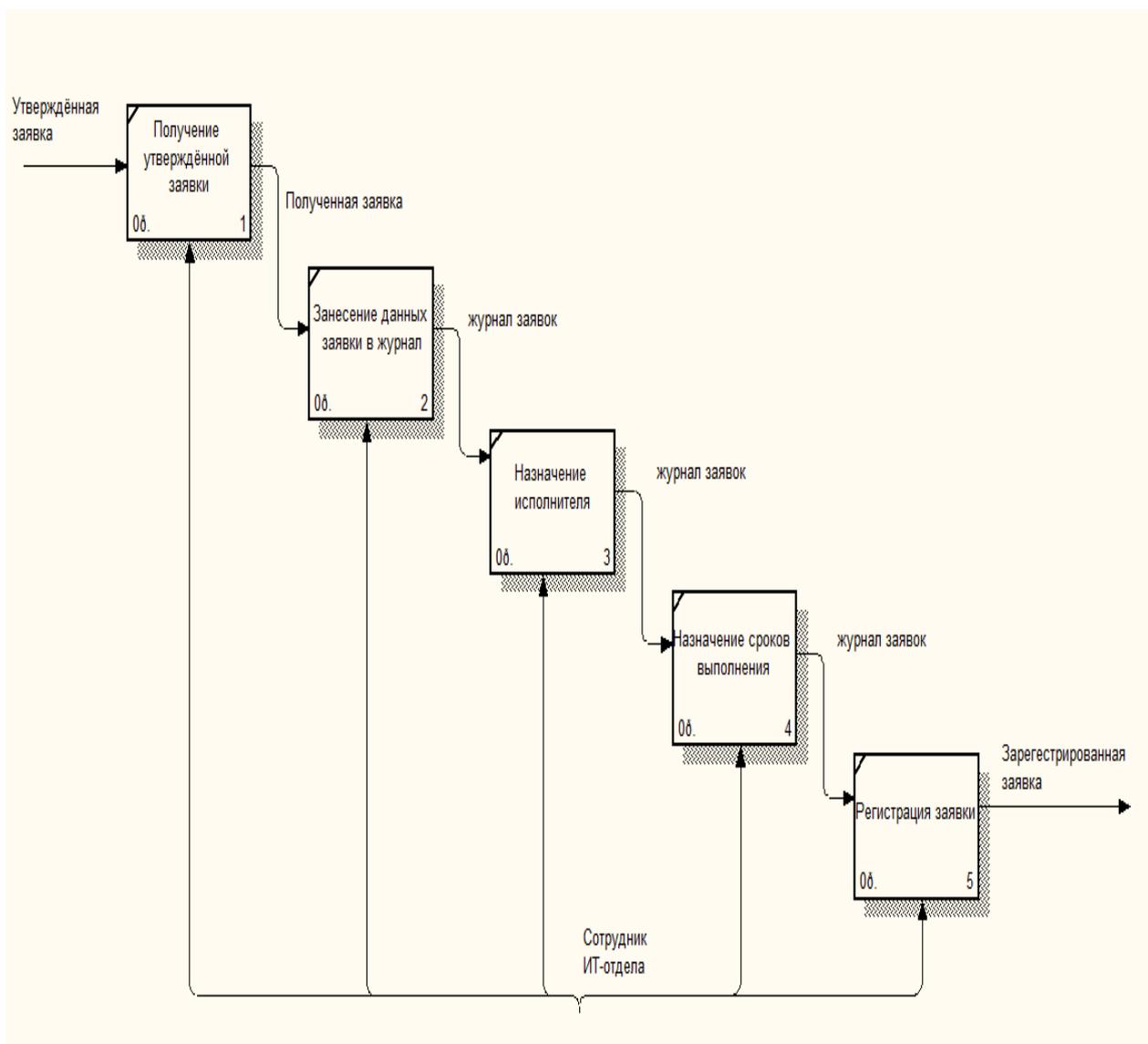


Рис 1.4 Декомпозиция бизнес-процесса регистрации заявки в журнале учёта заявок «КАК ЕСТЬ».

На рисунке 1.5 показана декомпозиция процесса утверждения заявки «КАК ЕСТЬ». Начальник ИТ - отдела получает заявку, далее проводится анализ заявки, после проведения анализа заявка утверждена либо отказана, если же заявка утверждена, она передаётся на регистрацию.

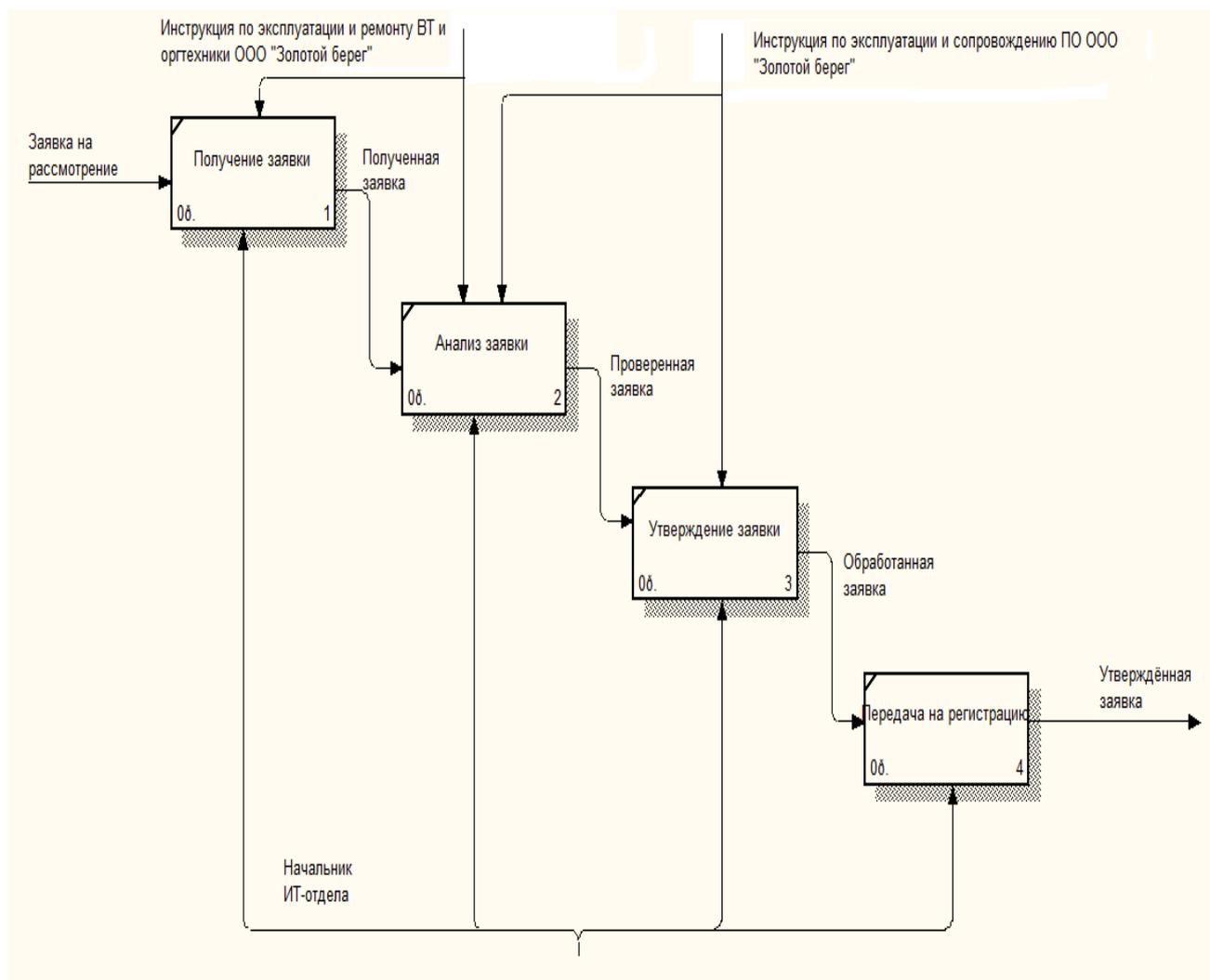


Рис 1.5 Декомпозиция процесса утверждения заявки «КАК ЕСТЬ»

Таким образом, главным недостатком данной модели «КАК ЕСТЬ» является ручная регистрация и сопровождение заявки на услуги.

По результатам анализа бизнес-модели «КАК ЕСТЬ» выполнения заявки на услуги по сопровождению ИТ-инфраструктуры было принято решение об автоматизации этого бизнес-процесса.

1.5 Анализ существующих программных разработок для автоматизации ИТ-отдела

В настоящее время на рынке программного обеспечения представляет собой огромный ряд программных продуктов, которые обеспечивают комплексную автоматизацию деятельности ИТ - служб предприятий.

Рассмотрим следующие аналоги программных продуктов АИС отдела информационных технологий.

Программа Hardware Inspector размещенная по адресу (<http://www.hwinspector.com/>)(рис. 1.8) является программным обеспечением для инвентарного учета компьютеров и оргтехники в организациях, и предназначена для обслуживающего персонала, руководителей отделов компьютеризации, администраторов сетей и др. ИТ - персонала.



Рис. 1.8 Заставка программы Hardware Inspector

Ключевые возможности:

- учет рабочих мест с детализацией до отдельных устройств;
- древовидная организация подразделений и рабочих мест;
- учет лицензий программного обеспечения;

- мониторинг инсталляции пользователями нелицензионного ПО;
- мониторинг изменений, которые произошли в конфигурации компьютера ;
- учет заявок от пользователей либо отделов;
- разграничение прав и доступа к данным;
- готовый набор гибко настраиваемых отчетов, а также экспорт в другие программы и форматы;
- поддержка многопользовательской работы с базой данных в сети и др.

Программа IT Invent размещенная по адресу (<http://it-invent.ru/>) создана для инвентаризационного учета компьютеров, IT оборудования, программного обеспечения, комплектующих и расходных материалов (рис. 1.9).

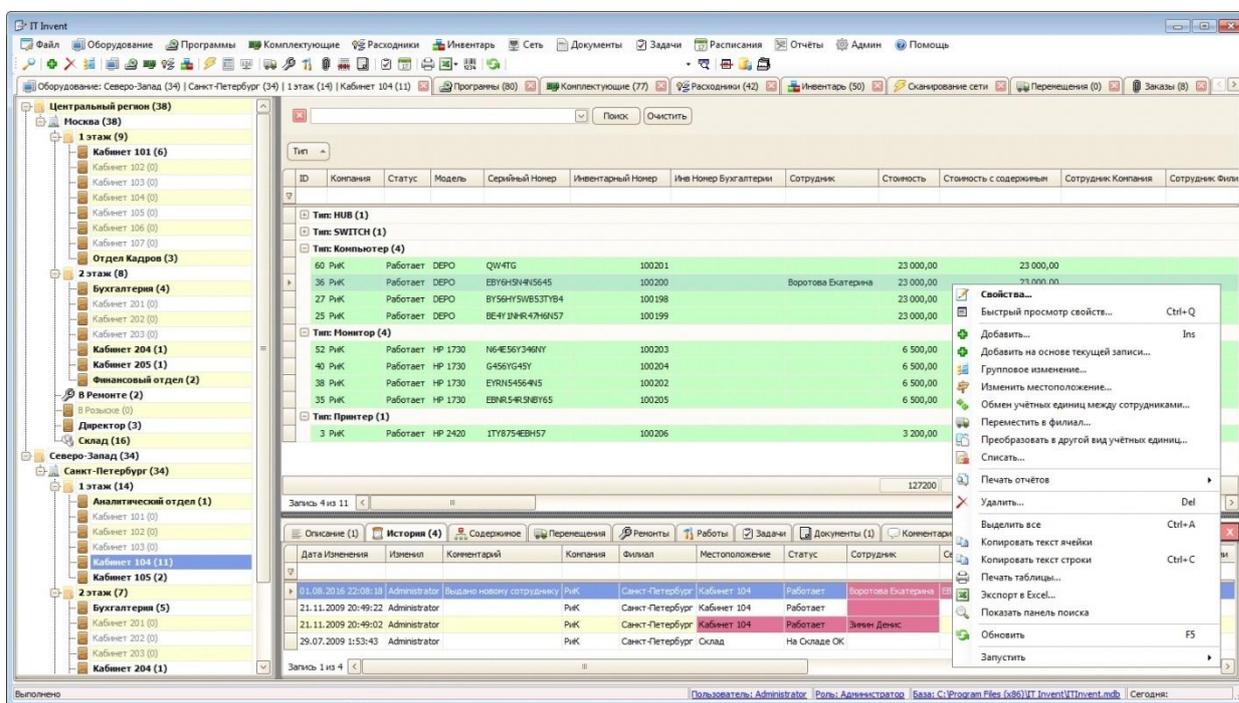


Рис. 1.9 Экран программы IT Invent

Основные функции программы:

- Автоматическая генерация и закрепление уникального инвентарного номера за каждым оборудованием либо комплектующим;
- возможность создать дополнительные свойства различных типов;
- ведение истории изменений по оборудованию;
- логическое связывание программ и комплектующих с оборудованием;
- ведение базы поставщиков, сервисных организаций и прочих партнеров;
- разграничение прав доступа для пользователей системы;
- Оповещения о событиях на различные почтовые сервисы сотрудников;

Программа Инвентаризация 2.0 (<http://itdim.com.ua>) предназначена для простого проведения инвентаризации. Ведет минимальный учет, реализует движение, списание итогового отчета. Есть поддержка различных фильтров для быстрого поиска (рис. 1.10).

Программа доступна и существует в бесплатном варианте.

Фамилия	Имя	Отчество	Телефон	Город
1 > Склад_				Киев

Тип	Модель	S/N	Цена	I/N	Дата счета	Номер счета	Дата прихода	Дата установки	Гарантия	Прошлый Юзер
1	Батарея для Ноутбук: PA3420U-1BR5 5200mAh	9001302181000	0	0			01.03.2011	09.12.2011	20.08.201	
2	Батарея для Ноутбук: A32-F5 5200mAh		0	0			13.01.2012			
3	Гарнитура к мобильно Nokia BH-207	0694657751402000282	0	0				27.06.2012		
4	Гарнитура к мобильно Nokia BH-207	069465781530100075	0	0				10.10.2011		
5	Гарнитура к мобильно Nokia BH-101	069424381...802006	0	0				07.10.2011		
6	Диктофон Olympus WS-450S	200162074	864	81970	25.10.2010	PHK-000014	25.10.2010	17.01.2012		
7	Диктофон Olympus WS-450S	200162371	864	81961	25.10.2010	PHK-000014	26.10.2010			
8	Компьютер (стационар)		0	0				06.03.2012		
9	Мобильный Nokia 5130		0	0				21.02.2012		
10	Мобильный Nokia 2700		0	0				12.04.2012		
11	Мобильный Nokia 5800		0	0				17.04.2012		

Рис. 1.10 Экран электронного журнала Инвентаризация

Характеристики сравнительного анализа исследуемых аналогов сведены в табл. 1.2

Табл. 1.2 Сравнительный анализ аналогов АИС ИТ - отдела

Система/ Характеристика	Программа Hardware Inspector	Программа IT Invent Full	Инвентаризация
Поддержка многопользовательской работы с базой данных в сети	+	+	+
Простота интеграции	+	+	+
Управление заявками на обслуживание ИТ-инфраструктуры	+	+	-
Составление отчётных документов	+	-	+
Реализация специфики бизнес-процессов ИТ -отдела ООО «Золотой берег»	-	-	-
Итого:	4	3	2

Как следует из таблицы 1.2, представленные ИТ-решения не позволяют в полной мере решить поставленные задачи.

В связи с этим предлагается разработать новую АИС, ориентированную на специфику ИТ-отдела ООО «Золотой берег».

1.6 Постановка задачи на разработку автоматизированной информационной системы

Исходя из анализа, проведённого нами в пункте 1.5 можно сделать вывод, что ни одна из рассмотренных уже существующих информационных систем в данной предметной области не удовлетворяет необходимым требованиям. Таким образом, еще раз подтверждается необходимость проектирования и разработки новой информационной системы для управления мероприятиями.

Сформулируем цель и основные требования к будущей информационной системе.

Целью создания информационной системы является предоставление удобного инструмента для автоматизации на управления заявками, автоматизация и упрощение рутинных процессов, возникающих в процессе получения и обработки заявки.

Для достижения поставленной цели были сформированы основные требования к будущей информационной системе:

1. Интуитивно-понятный интерфейс.
2. Формирование внутренней отчетности и возможность редактирования.
3. Обеспечить возможность работать с базой данных в сети.
4. Низкая цена владения и обслуживания автоматической информационной системы.

После того как была выполнена постановка задачи на разработку, описана цель ее создания и основные требования к будущей информационной системе, перейдем к разработке модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»

1.7 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»

Была построена контекстная модель «КАК ДОЛЖНО БЫТЬ», модель в данной ситуации показывает будущее предполагаемое состояние предметной области, и создана на основе контекстной модели «КАК ЕСТЬ» с устранением недостатков в существующей организации бизнес-процессов, а также, их улучшением и оптимизацией. Все это, достигается за счет устранения выявленных во время анализа контекстной модели «КАК ЕСТЬ» проблемных и спорных мест. В традиционном реинжиниринге рекомендуется проводить автоматизацию бизнес-процессов именно на основании модели «КАК ДОЛЖНО БЫТЬ», также, как и проектировать информационную систему. Что позволяет уменьшить риск проявления автоматизации как исключительно источника затрат из-за автоматизации несовершенных процессов. Контекстная модель «КАК ДОЛЖНО БЫТЬ» представлена на рисунке 1.11. По сравнению с моделью «КАК ЕСТЬ», добавился новый механизм – «Автоматизированная информационная система», который представляет собой информационную систему для приема и формирования отчета, а также его последующего редактирования если это будет нужно.

На основании вышеизложенного построена функциональная модель «КАК ДОЛЖНО БЫТЬ» исследуемого бизнес-процесса (рис. 1.11).

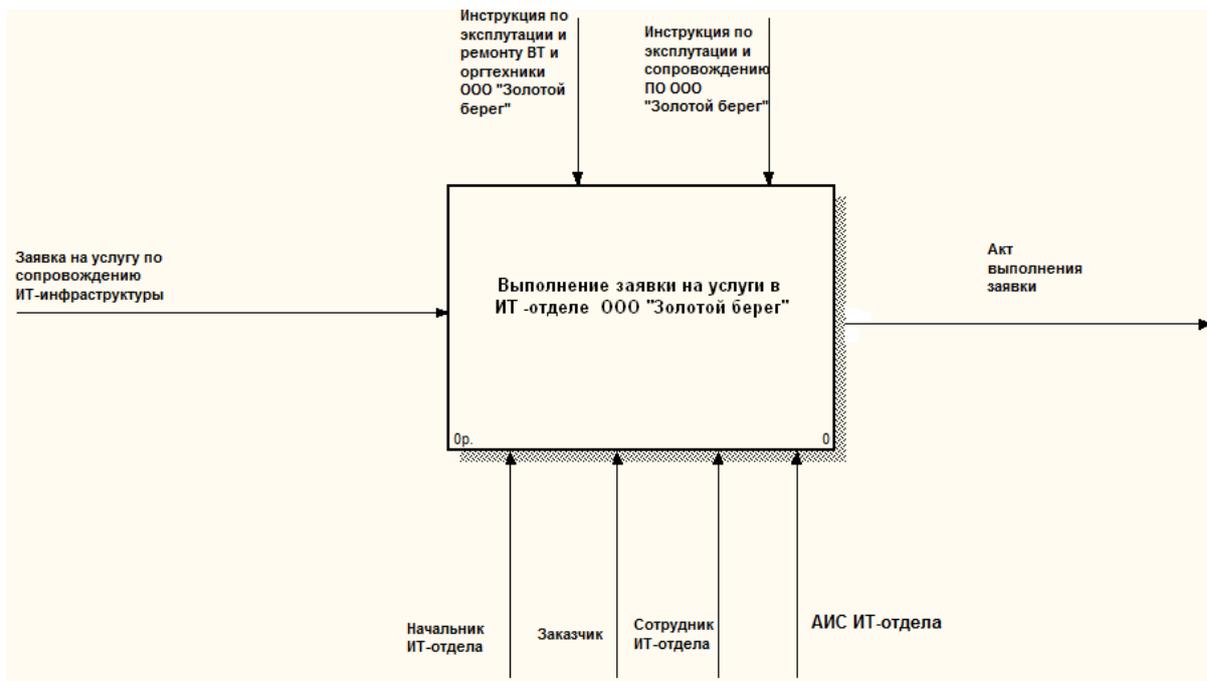


Рис. 1.11 Контекстная диаграмма (IDEF0) бизнес-процесса выполнения заявки на услуги по сопровождению ИТ - инфраструктуры ООО «Золотой берег» «КАК ДОЛЖНО БЫТЬ»

С помощью информационной системы начальник и сотрудники ИС-отдела смогут тратить меньше времени и затрачивать средств на принятие, формирование и обработку заявок.

Для работы с информационной системой необходимо будет пройти авторизацию сотрудника ИС-отдела, в ином случае доступ в информационной системы будет ограничен либо недоступен вовсе.

Для более наглядного отражения была представлена DFD диаграмма потоков данных бизнес-процесса (рис. 1.12).

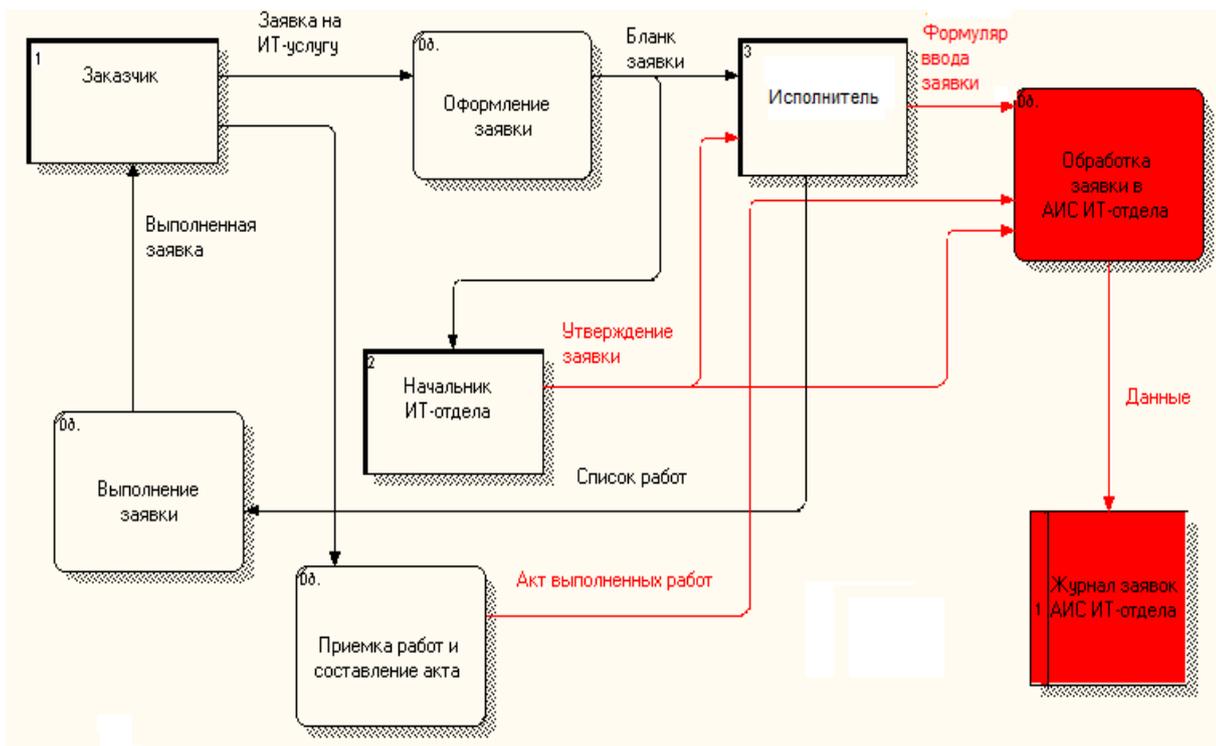


Рис. 1.12 Диаграмма потоков данных (DFD) бизнес-процесса выполнения заявки «КАК ДОЛЖНО БЫТЬ»

Для более наглядного отражения была представлена DFD диаграмма потоков данных бизнес-процесса (рис. 1.12).

На диаграммах выделены элементы, введенные в систему для достижения поставленной цели.

После того как была разработана контекстная модель бизнес-процесса «КАК ДОЛЖНО БЫТЬ», перейдем к заключению итогов второй главы.

Выводы к 1-й главе

Первая глава является описанием функционального моделирования предметной области. Была выбрана технология концептуального моделирования, также разработана модель бизнес-процесса «КАК ЕСТЬ», и сделан вывод о том, что часть процессов нуждается в автоматизации.

Следующим этапом стал сравнительный анализ существующих на данный момент информационных систем, для учета и инвентаризации оборудования, после анализа был сделан вывод что рассмотренные информационные системы не подходят по всем требованиям, следовательно был сделан вывод о том, что разработка удобной и не сильно требовательной и в материальном и техническом плане информационной системы необходима. Были сформированы и описаны требования к проектируемой информационной системе, после чего была поставлена задача на разработку новой информационной системы и разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ».

ГЛАВА 2 ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Обоснование модели данных

Хранимые в базе данные имеют определенную логическую структуру - иными словами, описываются некоторой моделью представления данных (моделью данных), поддерживаемой СУБД. К числу классических относятся следующие модели данных :

- иерархическая;
- сетевая;
- реляционная.

К достоинствам иерархической модели данных относятся эффективное использование памяти ЭВМ и неплохие показатели времени выполнения основных операций над данными. Иерархическая модель данных удобна для работы с иерархически упорядоченной информацией.

Недостатком иерархической модели является ее громоздкость в обработке информации с достаточно сложными логическими связями, необходимость использования той иерархии, которая была заложена в основу БД при ее проектировании, а также сложность понимания для обычного пользователя.

Достоинством сетевой модели данных является возможность эффективной реализации по показателям затрат памяти и оперативности. В сравнении с иерархической моделью сетевая модель предоставляет большие возможности в смысле допустимости образования произвольных связей.

Недостатком сетевой модели данных является высокая сложность и жесткость схемы БД, построенной на ее основе, а также сложность для понимания и выполнения обработки информации в БД обычным пользователем. Кроме того, в сетевой модели данных ослаблен контроль

целостности связей вследствие допустимости установления произвольных связей между записями.

Достоинство реляционной модели данных заключается в простоте, понятности и удобстве физической реализации на ЭВМ. Именно простота и понятность для пользователя явились основной причиной их широкого использования. Проблемы же эффективности обработки данных этого типа оказались технически вполне разрешимыми .

Основными недостатками реляционной модели являются отсутствие стандартных средств идентификации отдельных записей и сложность описания иерархических и сетевых связей.

Рассмотрев модели данных, мы склонны говорить, что наша система будет построена на основе реляционной модели. Поскольку РМД является наиболее простой и удобной формой для представления и формирования БД, а также универсальность и удобство обработки данных, которая осуществляется с помощью декларативного языка запросов SQL.

После определения модели данных переходим к этапу логического моделирования.

2.3 Разработка логической модели АИС

Логическая модель описывает деятельность предприятия посредством объектно-ориентированного проектирования (опираясь на методологию бизнес-моделирования и нотацию языка UML).

Цель логического моделирования - построить модель деятельности предприятия, которая является связующим звеном между бизнес-методиками и внедряемой АИС.

Конечной целью логического проектирования является разработка классов и спецификации объектов для использования на этапе реализации

программного приложения и отображения концептуальной модели АИС на логическую схему ее БД.

В свою очередь объектно-ориентированный подход использует объектную декомпозицию, при этом структура системы описывается в терминах объектов и связей между ними, а поведение системы - в терминах обмена сообщениями между объектами. Главными достоинствами визуальной модели являются ясность представления выбранного решения и относительная простота программной реализации.

В настоящее время для разработки логической модели систем используется Унифицированный язык моделирования UML (Unified Modeling Language) .

В методологии языка UML модель представляет собой совокупность диаграмм, описывающих различные аспекты структуры и поведения исследуемой системы. Однако при всем многообразии их типов при логическом моделировании ИС используются, как правило, диаграммы, составляющие ядро языка: диаграмма вариантов использования, диаграмма классов и диаграмма взаимодействия.

При ОО проектировании целесообразно использовать CASE- средства: StarUML, Visio и др.

В дипломной работе была использована условно-бесплатная версия пакета StarUML .

2.3.1 Диаграмма вариантов использования АИС

Диаграмма вариантов использования отражает организационный аспект системы.

Исполнитель (Действующее лицо) - личность, организация или система, взаимодействующая с АИС; различают внешнего исполнителя (который использует или используется системой, т.е. порождает прецеденты деятельности) и внутреннего исполнителя (который обеспечивает

реализацию прецедентов деятельности внутри системы). На диаграмме исполнитель представляется стилизованной фигуркой человека.

Прецедент-законченная последовательность действий, инициированная внешним объектом (личностью или системой), которая взаимодействует с ИС и получает в результате некоторое сообщение от ИС. На диаграмме представляется овалом с надписью, отражающей содержание действия.

Разработка диаграммы преследует следующие цели:

- определить общие границы моделируемой предметной области;
- сформулировать общие требования к функциональному поведению проектируемой системы;
- разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Варианты использования идентифицируются исходя из следующих соображений: каждый вариант использования представляет из себя некоторую функцию, которая в свою очередь выполняется системой в ответ на воздействие действующего лица (актера), и характеризует конкретный способ применения системы, либо диалог между актером и системой.

На рис. 2.1 представлена начальная диаграмма вариантов использования бизнес-процесса выполнения заявки на обслуживание ИТ - инфраструктуры ООО «Золотой берег».

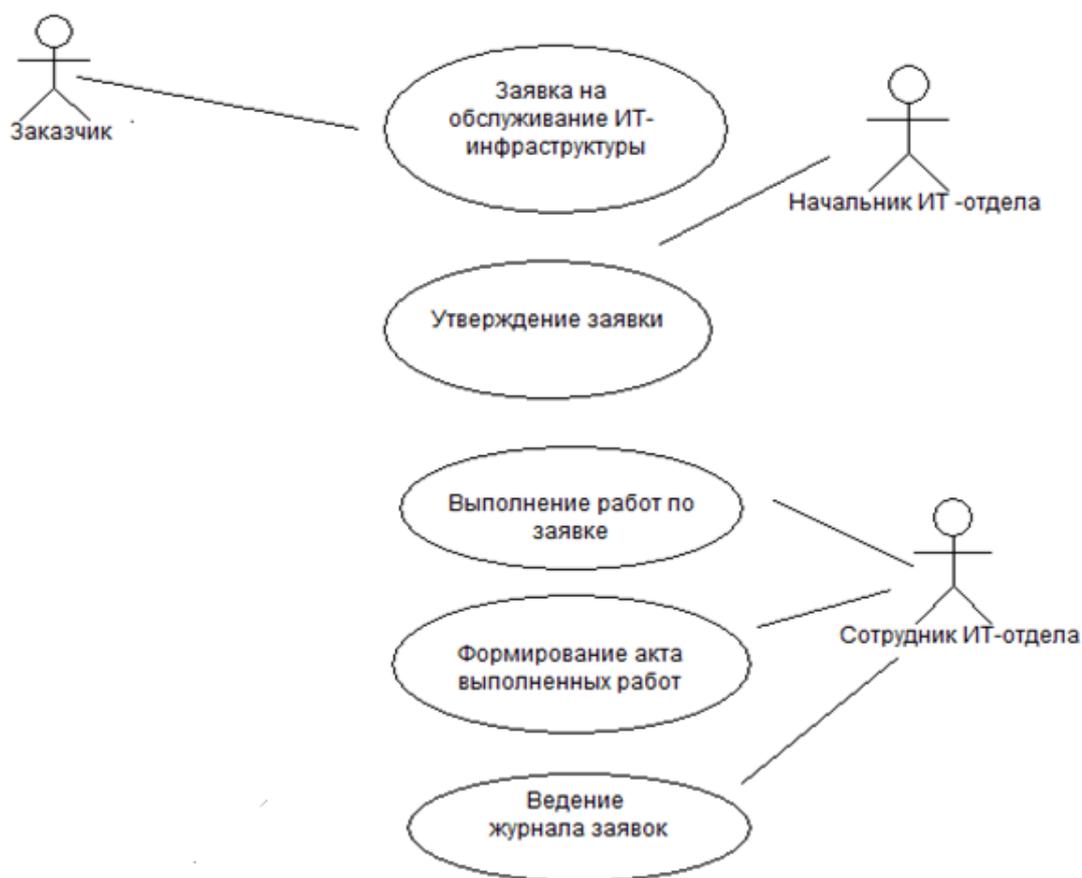


Рис. 2.1 Диаграмма вариантов использования выполнения заявки на обслуживание ИТ- инфраструктуры ООО «Золотой берег».

Как следует из диаграммы, в выполнении прецедента выполнения заявки участвуют следующие актеры:

Заказчик – оформляет заявку на обслуживание и акцептирует выполненные работы.

Начальник ИТ - отдела – утверждает заявку.

Сотрудник ИТ - отдела – обеспечивает выполнение работ по заявке, формирование акта выполненных работ, выполняет функции оператора АИС.

АИС ИТ - отдела – обеспечивает автоматизацию ведения журнала заявок и формирование отчетов.

2.3.2 Диаграмма классов АИС

На следующей стадии логического проектирования после уточнения диаграммы вариантов использования строится диаграмма классов системы .

Диаграмма классов служит для описания статического аспекта системы в терминологии классов объектно-ориентированного программирования. Обычно она отражают, различные взаимосвязи между отдельными сущностями предметной области, к примеру объекты или подсистемы, а также описывает их внутреннюю структуру и типы отношений.

Построение диаграммы классов АИС производится следующим образом:

- в описании классов выделяются кандидаты в классы – существительные, которые потенциально могут соответствовать классам (при этом следует помнить, что существительные могут относиться к объектам, ассоциациям или атрибутам классов);

- анализируются роли кандидатов в системе. Каждый класс должен выполнять некоторые действия и взаимодействовать с другими классами. Он должен иметь также уникальное имя, отражающее характер абстракции, представляемой данным классом.

С учетом этих рекомендаций составим диаграмму классов АИС ИТ - отдела (рис. 2.2).

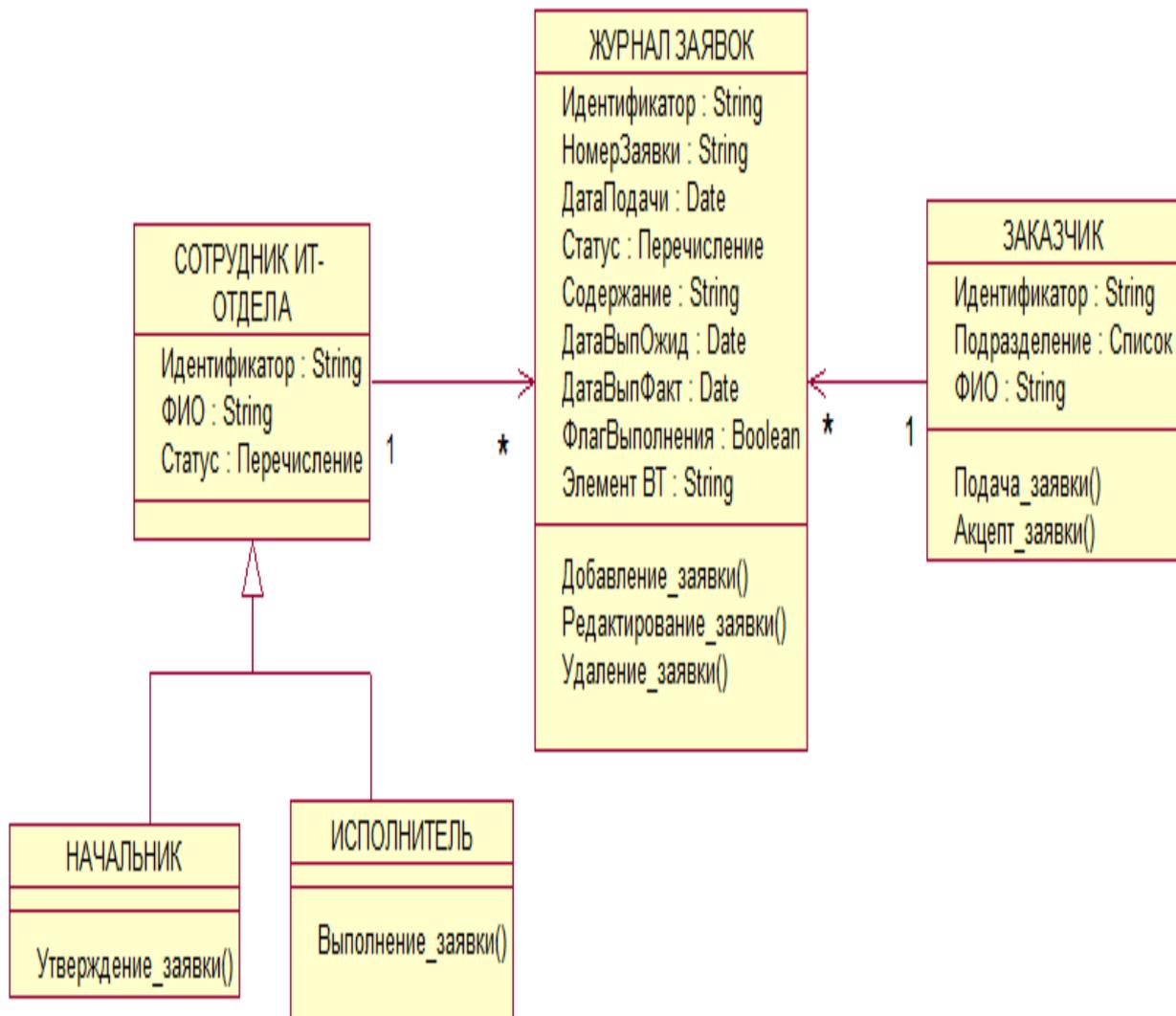


Рис. 2.2 Диаграмма классов АИС ИТ- отдела

Ниже представлена спецификация классов, объектов, атрибутов и операций АИС.

Как следует из диаграммы, основными классами АИС являются:

– **ЗАКАЗЧИК**

Атрибуты:

идентификатор: строка;

ФИО: строка;

подразделение: код из списка отделов ООО «Золотой берег».

Операции:

Подача_заявки()

Акцепт_заявки() – установка флага выполнения заявки в журнале заявок.

– **СОТРУДНИК ИТ-ОТДЕЛА**

Атрибуты:

идентификатор: строка;

ФИО: строка;

статус: начальник, ИТ-специалист.

Операции:

– **ИСПОЛНИТЕЛЬ**

Ведение_заявки(): ввод новой заявки, редактирование заявки, акцепт выполненной заявки;

– **НАЧАЛЬНИК**

Утверждение_заявки()

– **ЖУРНАЛ ЗАЯВОК**

Атрибуты:

идентификатор: строка;

номер заявки: строка;

дата подачи заявки: дата;

статус: принятая, отказная;

содержание заявки: строка;

дата выполнения ожидаемая: дата;

дата выполнения фактическая: дата;

флаг выполнения: логический.

Операции:

Добавление заявки();

Редактирование заявки();

Удаление заявки().

Связи между классами описываются следующим образом:

– между классами ЗАКАЗЧИК и ЖУРНАЛ ЗАЯВОК установлена ассоциация включения с множественностью «1- *» (каждый ЗАКАЗЧИК может подать одну или нескольких заявок);

– между классами СОТРУДНИК ИТ-ОТДЕЛА и ЖУРНАЛ ЗАЯВОК установлена ассоциация включения с множественностью «1- *» (каждый СОТРУДНИК ИТ-ОТДЕЛА может обработать одну или несколько заявок, при этом СОТРУДНИК ИТ-ОТДЕЛА со статусом «Начальник» имеет право на утверждение заявки);

– классы НАЧАЛЬНИК и ИСПОЛНИТЕЛЬ являются наследниками класса супертипа СОТРУДНИК ИТ-ОТДЕЛА

2.3.3 Требования к аппаратно-программному обеспечению автоматизированной информационной системы

- При выборе аппаратного обеспечения для реализации АИС, были учтены следующие требования к нему:
- Надежное функционирования ПО;
- скорость интернет соединения не менее чем 100 Мбит/с;
- возможность резервного копирования всей БД (ориентировочный размер БД – 3 Гб);
- возможность вывода информации на принтер;
- ориентация на существующую на предприятии ИТ- инфраструктуру.
- Минимальные требования к серверу СУБД и АИС:
- процессор – частота не менее 3.6 ГГц;
- оперативная память – не менее 8 Гб;
- свободное место на жестком диске - не менее 40 Гб;
- подключение к сети Интернет;
- операционная система - Windows.

Минимальные требования к ПК пользователей:

- процессор – частота не менее 2,8 ГГц;
- оперативная память – не менее 4 Гб;
- подключение к сети Интернет;
- ПК под управлением любой популярной ОС (Windows, Mac OS, Linux и др.);
- любой из популярных браузеров (Google Chrome, Mozilla Firefox, Opera и др.).
- Минимальные требования к мобильным устройствам пользователей:
- мобильное устройство под управлением любой популярной мобильной ОС (Android, iOS и др.);
- любой из популярных мобильных браузеров (Google Chrome, Mozilla Firefox и др.);
- оперативная память – не менее 2 Гб;
- подключение к сети Интернет.

После того как были сформулированы и описаны требования к аппаратно-программному обеспечению информационной системы, переходим к разработке логической модели данных.

2.4 Разработка логической модели данных АИС

Целью моделирования данных является обеспечение разработчика АИС концептуальной схемой данных БД в форме одной либо нескольких моделей, которые легко могут быть отображены в любую систему БД.

АИС ИТ- отдела, как учетная система, относится к категории OLTP – систем.

OLTP (Online Transaction Processing) – категория приложений и систем, предназначенных для ввода, структурированного хранения и обработки информации (операций, документов) в режиме реального времени.

Поэтому в АИС ИТ - отдела в качестве базовой модели данных используется реляционная модель с высокой степенью нормализации данных.

Главное преимущество реляционной модели данных, используемое в настоящей разработке - это практическая возможность объединения на логическом уровне проблемно-ориентированного программного кода и оперативных данных системы в рамках единой базы данных.

ERD - модель традиционно используется в структурном анализе и проектировании, однако, по существу, представляет собой подмножество объектной модели предметной области.

Одна из разновидностей модели «сущность-связь» используется в методе IDEF1X, входящем в семейство стандартов IDEF и реализованном в ряде распространенных CASE-средств (в частности, ERwin, Design/IDEF).

Методология IDEFIX подразделяется на уровни, соответствующие проектируемой модели данных системы. Каждый такой уровень соответствует определенной фазе проекта. Такой подход полезен при создании систем по принципу «сверху вниз».

Верхний уровень модели данных представлен логической моделью данных, нижний – физической моделью данных.

Процесс разработки модели данных по методу IDEF1X можно разбить на следующие этапы:

- выделение сущностей с помощью ERD- метода путем сопоставления с диаграммой классов;
- определение зависимостей между сущностями;
- задание первичных и альтернативных ключей;
- построение логической модели данных;

По результатам анализа диаграммы классов АИС ИТ-отдела (рис. 2.2) были выделены следующие сущности:

- ЗАКАЗЧИК;
- СОТРУДНИК ИТ-ОТДЕЛА;

- ОБОРУДОВАНИЕ (импортируемый справочник);
- ЖУРНАЛ ЗАЯВОК;
- СТАТУС СОТРУДНИКА ИТ-ОТДЕЛА (классификатор);
- СТАТУС ЗАЯВКИ (классификатор);
- ПОДРАЗДЕЛЕНИЕ ОБЩЕСТВА (импортируемый справочник).

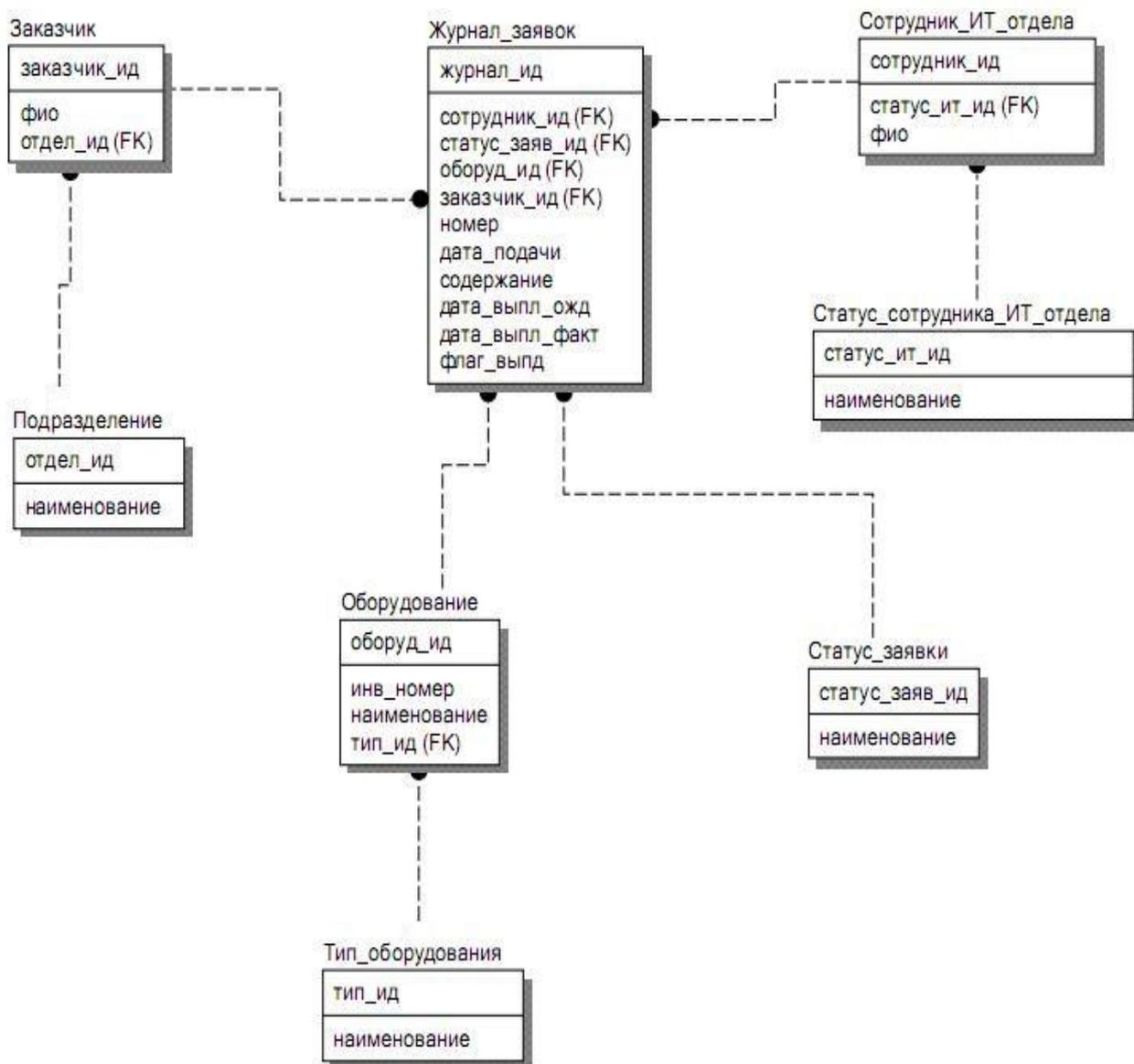


Рис. 2.3 Логическая модель данных АИС ИТ-отдела.

На рис. 2.3 представлена логическая модель данных АИС ИТ - отдела, построенная по методу IDEF1X.

Связи между сущностями опишем следующим образом:

- ЗАКАЗЧИК должен подать одну или несколько заявок в ЖУРНАЛ ЗАЯВОК (один ко многим, неидентифицирующая);
 - ЗАКАЗЧИК может, но не обязан быть сотрудником какого-либо ПОДРАЗДЕЛЕНИЯ общества (один ко многим, неидентифицирующая);
 - СОТРУДНИК ИТ - отдела обязан иметь статус (один ко многим, неидентифицирующая);
 - СОТРУДНИК ИТ - отдела может выполнять одну или несколько заявок из ЖУРНАЛА ЗАЯВОК (один ко многим, неидентифицирующая);
 - Каждая заявка из ЖУРНАЛА ЗАЯВОК может иметь СТАТУС (один ко многим, неидентифицирующая);
 - ОБОРУДОВАНИЕ может, но не обязано иметь ТИП ОБОРУДОВАНИЯ (один ко многим, неидентифицирующая);
- Разработанная логическая модель данных является основой для построения физической модели данных АИС .

Выводы ко 2-й главе

Вторая глава представляет собой описание логического проектирования информационной системы. Был сделан выбор технологии логического моделирования, после чего была построена логическая модель информационной системы, включающая в себя диаграмму вариантов использования и диаграмму классов, а также, было выполнено построение концептуальной и логической модели базы данных.

ГЛАВА 3 ФИЗИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОЕКТНЫХ РЕШЕНИЙ ПО АВТОМАТИЗАЦИИ

3.1 Обоснование выбора средств для реализации АИС

3.1.1 Разработка типа архитектуры АИС

Базовыми принципами реализации документально - фактографической АИС являются :

– комплексность, подразумевающая решение задач автоматизации на основе комплексного подхода, благодаря которому обеспечивается информационное взаимодействие всех основных подразделений Общества, участвующих в процессе выполнения заявки на обслуживание ИТ-инфраструктуры;

– принцип организации единой базы данных, содержащей всю информацию по заявкам на обслуживание ИТ - инфраструктуры;

– принцип масштабируемости, обеспечивающий произвольное наращивание количества автоматизированных рабочих мест, а также возможность увеличения производительности АИС за счет добавления новых программных и/или аппаратных ресурсов;

– принцип гибкости, подразумевающий возможность настройки АИС в соответствии с текущими требованиями;

– принцип безопасности, подразумевающий защиту БД от несанкционированного доступа и др.

При выборе архитектуры системы следуют учесть также следующие требования заказчика:

- уменьшение сетевого трафика;
- простота интеграции АИС с существующей на предприятии ИТ-инфраструктуру;
- простота бизнес - логики обработки данных;
- простота в использовании и сопровождении;

- минимальные затраты на приобретение и эксплуатацию.

На основе этого были рассмотрены и сравнены известные модели архитектуры построения ИС.

Архитектура «клиент-сервер» может быть реализована в следующих вариантах:

Двухзвенная архитектура «клиент - сервер»:

Первый уровень -это клиентские компьютеры с прикладными программами, которые позволяют пользователям обращаться по сети к базе данных.

Второй уровень- сервер баз данных (СБД), также участвующий в обработке данных.

Благодаря двухуровневой архитектуре снижается нагрузка на информационную сеть, поскольку передаются только запросы и ответы на них.

Трехзвенная архитектура «клиент - сервер»:

Первый уровень – клиентская программа. Клиент посылает запросы на выполнение нужных действий. Вся логика системы реализуется на других уровнях, поэтому требования к клиентской машине минимальные. Такие клиентские программы называют тонкими клиентами.

Второй уровень – сервер приложений – программное обеспечение промежуточного уровня. Синхронизирует работу компонентов всей системы и организует связь между ними. Обычно это сетевые службы или приложения.

Третий уровень – СБД. Данный сервер не работает напрямую с клиентскими программами, что в свою очередь повышает безопасность информации в системе.

Результаты сравнительного анализа архитектур построения АИС показаны в табл. 3.1.

Табл. 3.1 Сравнительный анализ архитектур построения АИС.

Архитектура/ Требование	Файл-сервер	Клиент-сервер, двухзвенная архитектура	Клиент-сервер трехзвенная архитектура
Уменьшение расхода сетевого трафика и его оптимизация	-	+	+
Простота интеграции с существующей ИТ-инфраструктурой	+	+	-
Простая бизнес- логика обработки данных	+	+	+
Простота в использовании и сопровождении	+	+	-
Минимальные затраты на приобретение и эксплуатацию	+	+	-
Итого	6	8	3

По результатам анализа предлагается использовать при реализации АИС ИТ- отдела двухзвенную архитектуру «клиент-сервер» (рис.3.1).



Рис. 3.1 Структурная схема АИС с двухзвенной архитектурой «клиент-сервер».

На стороне клиента выполняется код приложения, куда входят компоненты, в числе которых поддержка интерфейса с конечным пользователем, производство отчетов, либо позволяющие выполнять другие функции, в зависимости от требования заказчика либо специфики предметной области.

Клиентская часть приложения взаимодействует с клиентской частью программного обеспечения управления базами данных, которая, фактически, является индивидуальным представителем СУБД для приложения.

Одним из главных преимуществ архитектуры клиент-сервер является ее масштабируемость и способность к развитию, поскольку увеличение масштаба информационной системы не порождает принципиальных проблем. Даже при замене аппаратуры сервера практически не затрагивается прикладная часть информационной системы.

3.1.2 Выбор системы управления базами данных

Важнейшая задача компьютерных систем — хранение и обработка данных. Для ее решения были приложены усилия, которые привели к появлению специализированного программного обеспечения — систем управления базами данных (СУБД).

СУБД позволяют структурировать, систематизировать и организовать данные для их компьютерного хранения и обработки.

Современные СУБД содержат:

- инструментарий для проектирования и реализации выбранной реляционной модели БД;
- набор средств для поддержки таблиц и отношений между связанными таблицами;
- развитый пользовательский интерфейс, который позволяет вводить и модифицировать информацию, выполнять поиск и анализ данных;

- средства программирования высокого уровня, с помощью которых создаются собственные приложения;
- выбирать информацию по запросу и выполнять различные вычисления в процессе подготовки отчетов.

Для реализации АИС в архитектуре «клиент-сервер» в качестве СУБД применяются так называемые сервера баз данных (СБД): MS SQL Server, Oracle, MySQL и др.

ИТ - инфраструктура ООО «Золотой берег» создавалась и развивалась с ориентацией на семейство операционных систем Windows корпорации Microsoft, широко распространенных в системах автоматизации производственно-хозяйственной деятельности российских предприятий малого и среднего бизнеса.

В этой связи использование СУБД MS SQL Server 2008 в качестве СУБД является требованием ООО «Золотой берег».

Microsoft SQL Server 2008 –представляет из себя высокую в плане производительности платформу для обработки или анализа данных для бизнеса любого масштаба. Оно отвечает всем современным стандартам и требованиям, удобна для работа с данными разных типов и обладает удобной поддержке при разработки приложений. Позволяет хранить не только структурированную, или неструктурированную информацию, к примеру как отчеты либо изображения в любом формате. Также в наличие огромное количество встроенных служб ,которые позволяют совершать такие действия как :составление запроса, поиск и синхронизация документов или других данных, составление отчетов и анализ данных.

Главным достоинством MS SQL Server является возможность использование мощного языка программирования T-SQL для разработки хранимых процедур и триггеров, тесная интеграция с Windows NT и семейством продуктов BackOffice – общая модель защиты, основой защиты которой является Windows NT.

Администрирование MSDE будет производиться при помощи утилиты osql, клиентских утилит от MS SQL Server, либо из MS Access 2008.

Клиентская часть MS SQL Server реализована на платформе Win32. В стандартный комплект входят драйверы, работающие под управлением Windows XP и Windows 2000.

Все это делает MS SQL Server удобным инструментарием привлекательным решением для реализации БД на платформе Windows 2000/XP.

С учетом вышеизложенного в качестве СБД выбираем MS SQL Server 2008.

3.1.3 Выбор среды разработки программного обеспечения

При разработке программного обеспечения в виде клиентского приложения на основе спиральной модели его жизненного цикла важно учитывать следующие критерии:

- поддержка технологии быстрой разработки приложений RAD (Rapid Application Development), основа которой, будет построена на объектно-ориентированной технологии программирования;
- поддержка архитектуры клиент-сервер;
- принадлежность средства разработки к линейке продуктов корпорации Microsoft;
- преимущество используемого на предприятии ПО.

В ООО «Золотой берег» в качестве средства реализации прикладных ИТ - решений используется среда разработки СБД MS Visual FoxPro 9.0 (VFP) .

Среди достоинств VFP можно выделить наличие встроенного объектно-ориентированного языка для разработки приложений, языка SQL, а также возможность легкой интеграции с MS SQL Server.

Объектно-ориентированный язык программирования VFP является мощным, гибким и удобным в использовании инструментом для решения огромного количества задач.

Его возможности расширены с помощью ряда библиотек и элементов ActiveX, что позволяет создать на его базе среду разработки приложений для автоматизации страхового бизнеса, в том числе поддерживающие двухзвенную архитектуру «клиент-сервер» (рис.3.2).

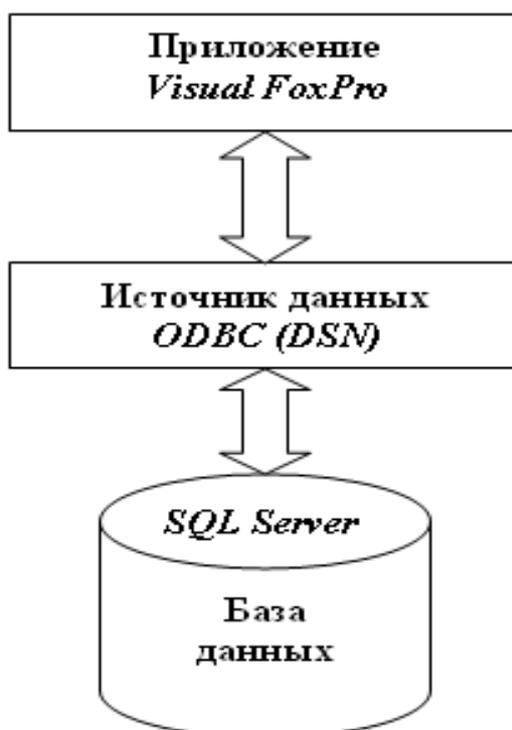


Рис.3.2 Структурная схема АИС ИТ-отдела

Интеграция приложения с СБД осуществляется с помощью интерфейса ODBC (Open Data base Connectivity), а так же при помощи специальной технологии «сквозных SQL-запросов».

Для соблюдения бизнес-логики системы, предусматривающей атомарность операции ввода нового страхового полиса, в АИС используется механизм транзакций СБД MS SQL Server 2008 на уровне хранимых процедур.

Для обеспечения целостности данных использованы диаграммы БД, которые строятся с помощью дизайнера диаграмм MS SQL Server 2008.

Имеется возможность включения средств автоматизации с применением технологии OLE, в качестве объектов которой используется книга Excel и шаблоны документа Word.

Код, используемый для вставки OLE-объектов, строится на базе языка VBA для MS Office.

Учитывая все вышесказанное мы принимаем решение о том, что среда разработки VFP 9.0 будет основным средством для разработки клиентского приложения АИС ИТ-отдела.

3.2. Разработка физической модели данных автоматизированной информационной системы

В отличие от логической модели данных, физическая модель данных отображает структуру базы данных на примере конкретной СУБД, был сделан выбор СУБД для информационной системы. Разработка физической модели данных будет осуществлено при помощи CASE-средства StarUML.

Для создания базы данных информационной системы будет использован инструмент MySQL Workbench.

MySQL Workbench – представляет собой инструмент для визуального проектирования баз данных, который включает в себя проектирование, моделирование, создание и эксплуатацию базы данных для СУБД MySQL.

Определившись с инструментами для разработки физической модели данных и создания базы данных, перейдем к разработке физической модели данных.

Разработанная физическая модель данных представлена на рис 3.3.

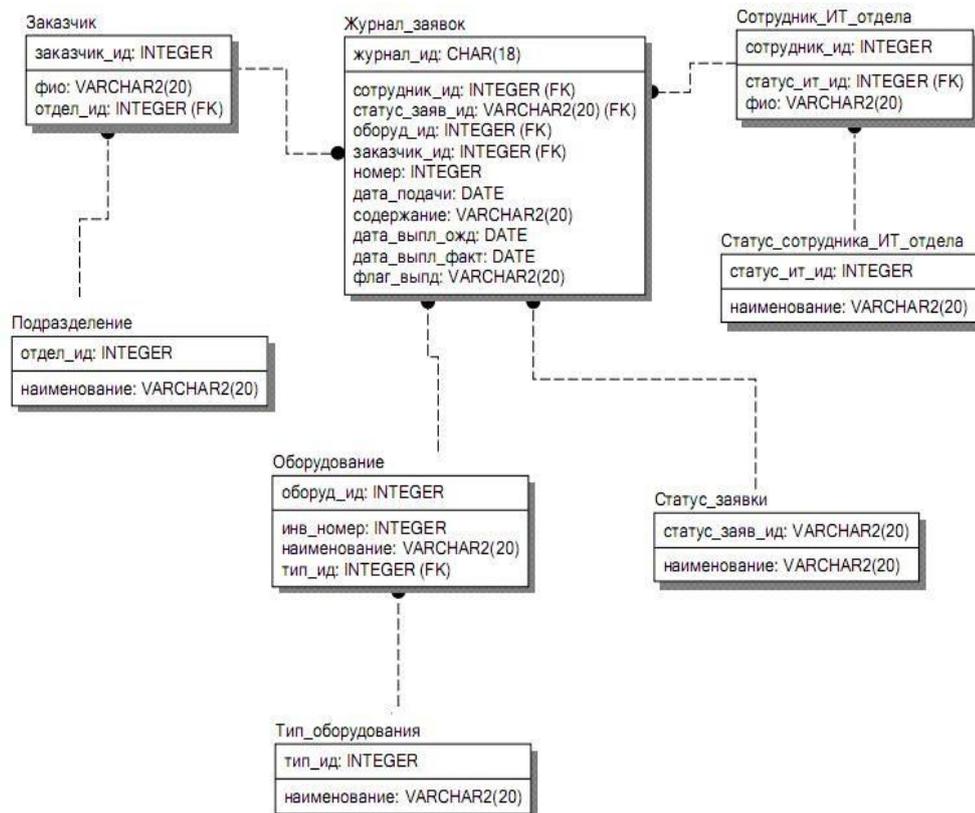


Рисунок 3.3 – Физическая модель данных

Физическая модель данных содержит в себе 8 сущностей: «Журнал_заявок», «Заказчик», «Подразделение», «Сотрудник_ИТ_отдела», «Статус_сотрудника_ИТ_отдела», «Оборудование», «Статус_заявки», «Тип_оборудования».

Все сущности связаны отношением «один-ко-многим».

После того как была построена физическая модель данных, перейдем к описанию функциональности информационной системы.

3.3. Структурная схема программного обеспечения АИС

Структурная схема программного приложения представлена на рис.

3.4



Рис. 3.4 Структурная схема программного приложения АИС

Здесь:

Модуль «Журнал учета заявок» - предназначен ведения и учета заявок на сопровождение ИТ - инфраструктуры предприятия.

Модуль «Справочники» - предназначен для ведения справочников АИС.

Модуль «Сервис и настройки» - предназначен настройки параметров АИС.

Модуль «Отчетность» - предназначен для формирования отчетности.

Модуль «БД MS SQL Server» - предназначен для поддержки бизнес-логики обработки данных. Размещен в блоке хранимых процедур СУБД MS SQL Server (прил. А).

Взаимодействие между модулями обеспечивается с помощью системы управления и интерфейса АИС.

В соответствии с предлагаемой схемой реализовано клиентское приложение АИС.

Основой разработки программ VFP является проект (Project).

Проект выглядит на экране как особое окно, содержащие списки всех его компонентов, входящих в состав пользовательского приложения, для обеспечения легкого доступа к ним. В VFP окно проекта разбито на отдельные страницы по видам компонентов проекта.

Во вкладке Classes содержатся ссылки на библиотеки ОО классов компонентов VFP (рис. 3.5)

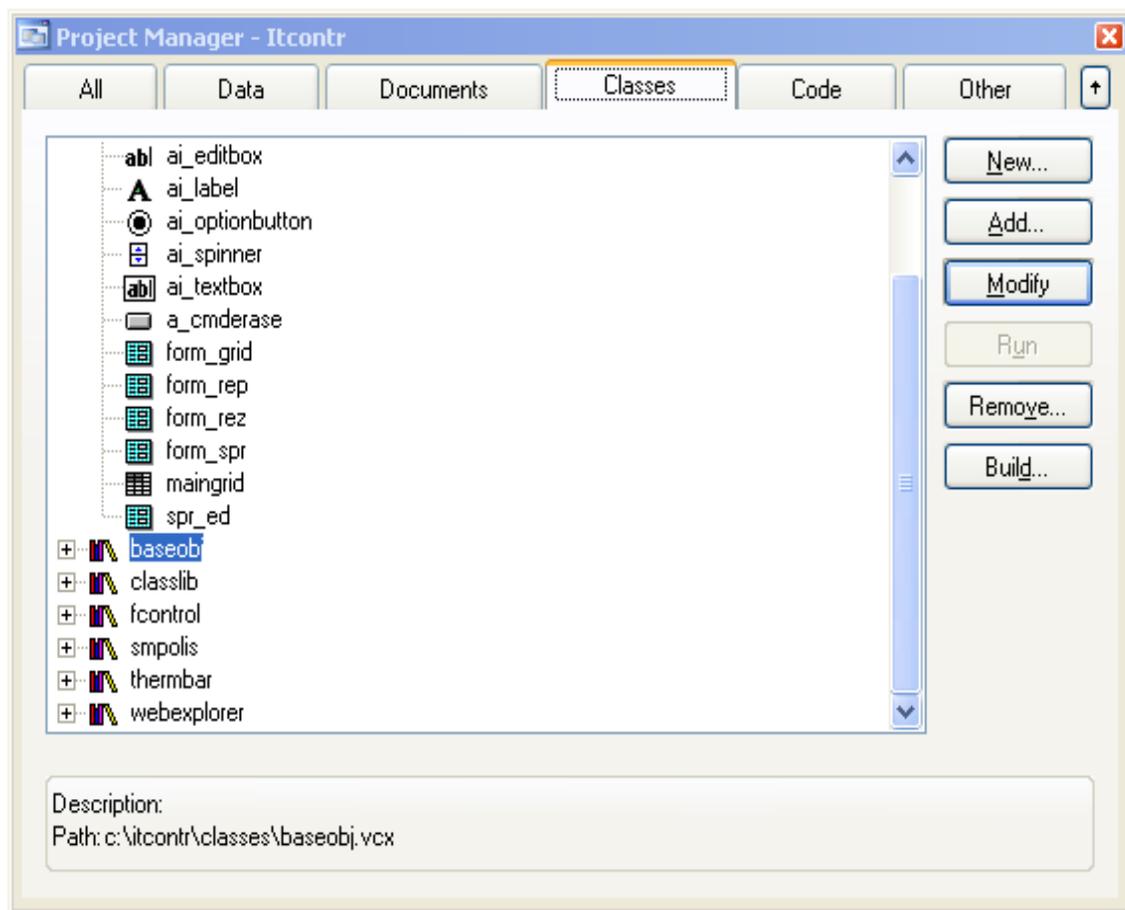


Рис. 3.5 Экран проекта приложения АИС

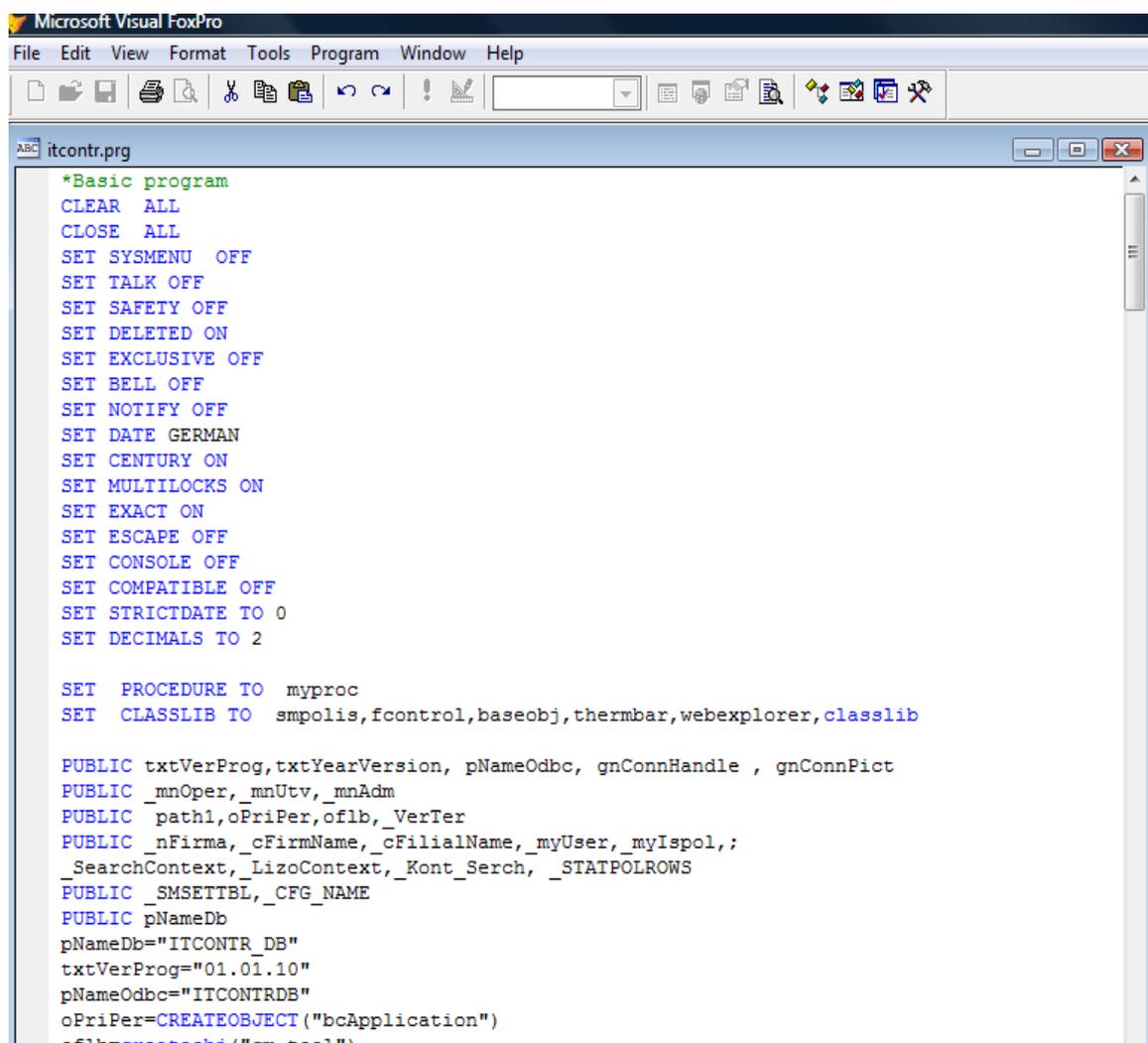
При разработке клиентского приложения в его проекте использованы стандартные библиотеки классов среды разработки VFP 9.0 (рис. 3.6)

В VFP имеются в наличие все необходимые инструменты для фильтрации или обработки данных, а также для управления объемами информации.

В VFP основными объектами являются таблицы, запросы, формы, отчеты, программы и классы.

Структурно клиентское приложение представляет классическое приложение VFP и состоит из следующих компонентов

главная программа – ITCONR.PRG;



```
*Basic program
CLEAR ALL
CLOSE ALL
SET SYSMENU OFF
SET TALK OFF
SET SAFETY OFF
SET DELETED ON
SET EXCLUSIVE OFF
SET BELL OFF
SET NOTIFY OFF
SET DATE GERMAN
SET CENTURY ON
SET MULTLOCKS ON
SET EXACT ON
SET ESCAPE OFF
SET CONSOLE OFF
SET COMPATIBLE OFF
SET STRICTDATE TO 0
SET DECIMALS TO 2

SET PROCEDURE TO myproc
SET CLASSLIB TO smpolis,fcontrol,baseobj,thermbar,webexplorer,classlib

PUBLIC txtVerProg,txtYearVersion, pNameOdbc, gnConnHandle , gnConnPict
PUBLIC _mnOper,_mnUtv,_mnAdm
PUBLIC path1,oPriPer,oflb,_VerTer
PUBLIC nFirma,_cFirmName,_cFilialName,_myUser,_myIspol,;
_SearchContext,_LizoContext,_Kont_Serch,_STATPOLROWS
PUBLIC _SMSETTBL,_CFG_NAME
PUBLIC pNameDb
pNameDb="ITCONTR_DB"
txtVerProg="01.01.10"
pNameOdbc="ITCONTRDB"
oPriPer=CREATEOBJECT("bcApplication")
oflb=createobj("sm_tool")
```

Рис. 3.6 Экран разработки главной программы

Главная программа разработана на языке xBase (вкладка Code). Она состоит из набора функций и процедур, логически связанных между собой.

блок меню – ITC_MENU.MPR;

Меню разрабатывается во вкладке Others с помощью встроенного Дизайнера меню, который позволяет создавать обычные и контекстные меню.

Главное меню программы состоит из набора подменю, каждое из которых содержит команды активизации соответствующей экранной формы.

Управление меню осуществляется с помощью менеджера управления отдельными опциями.

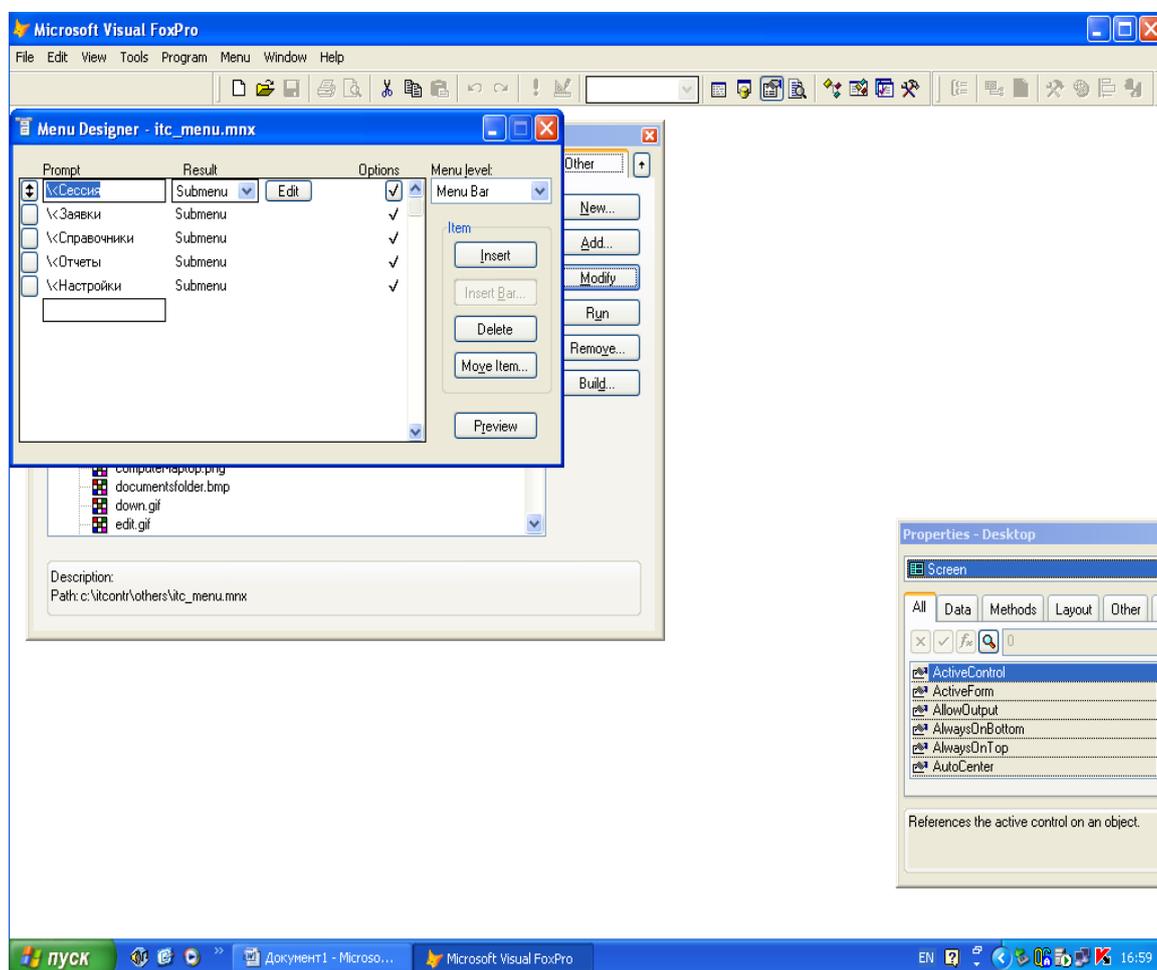


Рис. 3.7 Экран разработки меню
блок экранных форм проекта (вкладка Documents- FORMS)

Форма - объект, предназначенный для ввода данных, отображения их на экране либо же управления работой приложения. Формы используются для того, чтобы реализовать требования пользователя к представлению данных таблиц либо наборов записей запросов. С помощью форм можно в ответ на некоторое действия со стороны пользователя запустить функцию, процедуру, метод формы или класса.

Для разработки форм используется встроенный Дизайнер формы, позволяющий визуально создавать и модифицировать формы и формировать наборы форм. Набор форм состоит из одной или более форм, которые могут редактироваться как отдельный модуль. Формы и наборы формы – это объекты со своими собственными свойствами, событиями и методами.

Когда окно Дизайнер формы активно, VFP отображает меню формы, меню компонентов и окно свойств.

На форме можно добавлять и модифицировать элементы управления.

В VFP можно создать новые классы форм других компонентов и создавать на их основе новые объекты.

Экранные формы проекта разработаны на базе следующих классов форм:

Класс – журнальная форма (рис. 3.8);

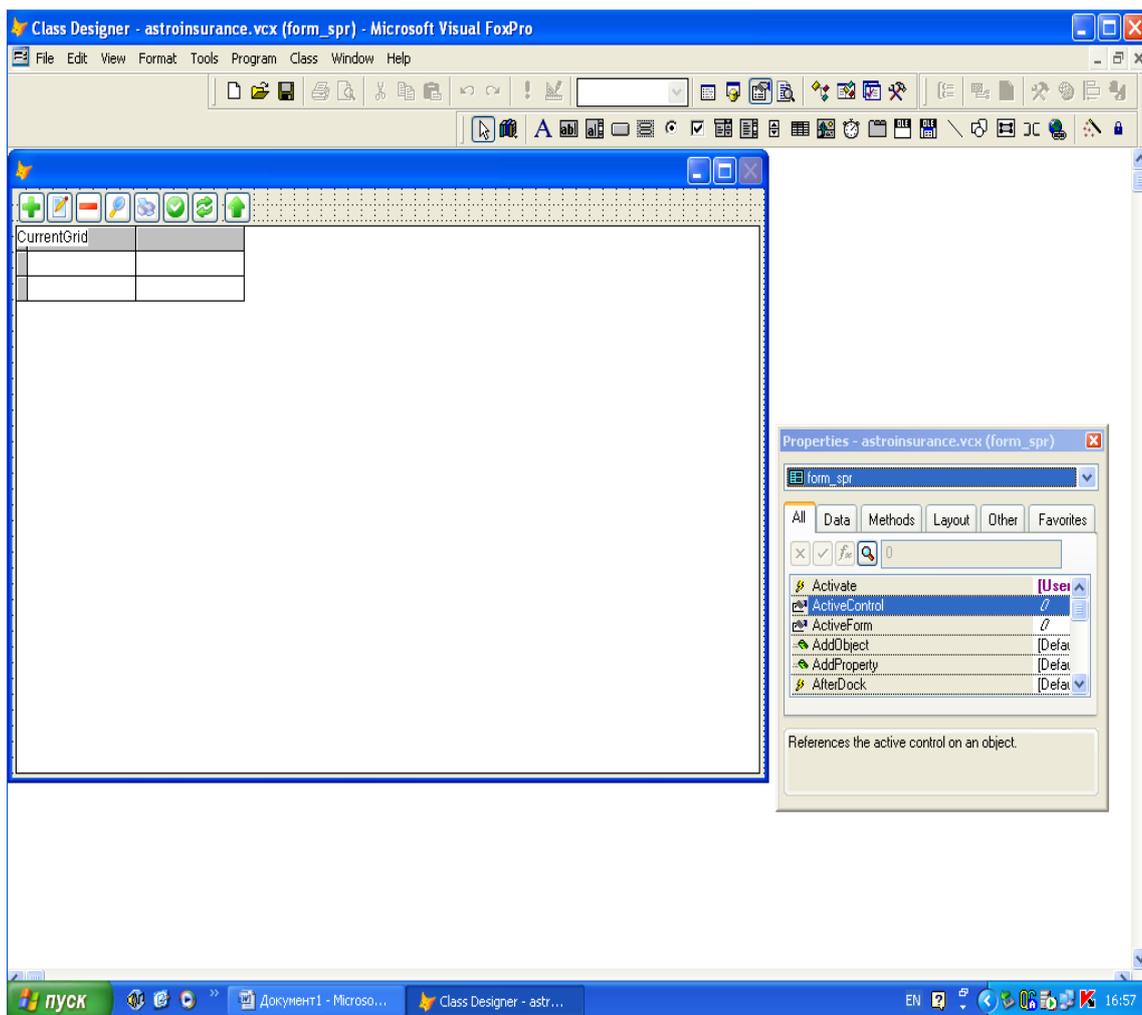


Рис. 3.8 Экран разработки формы журнала заявок

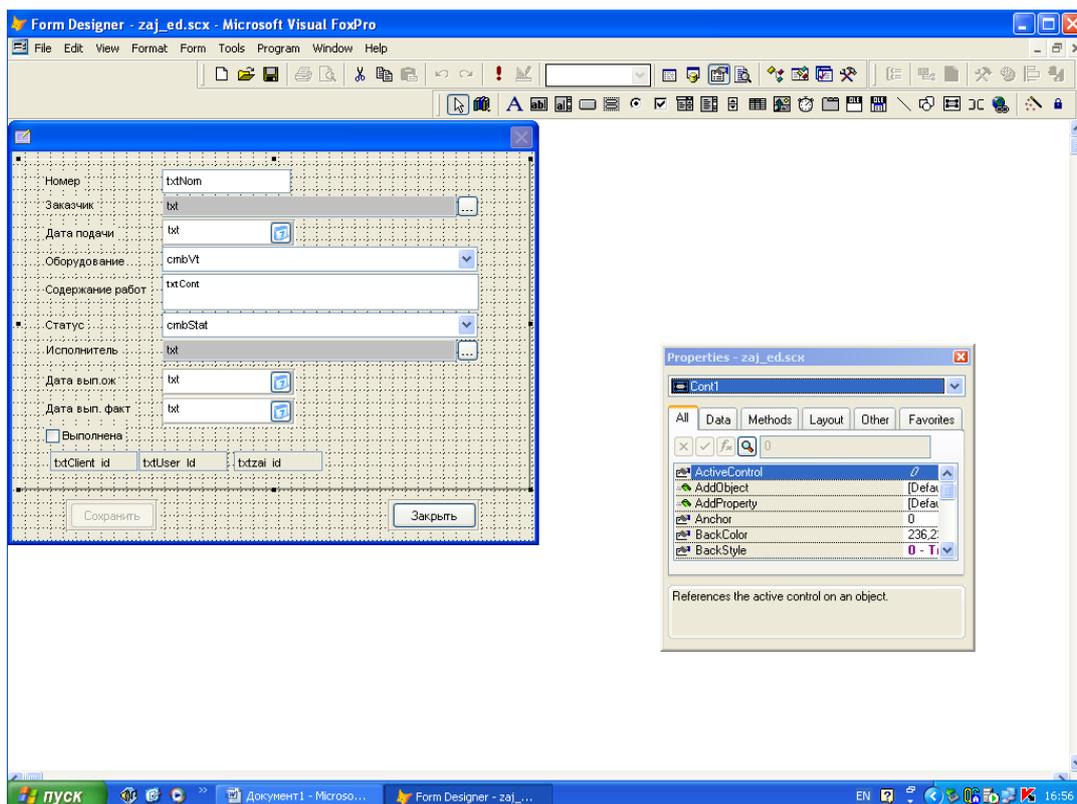


Рис. 3.9 Экран разработки формы ввода заявок

Класс – форма отчета (рис. 3.10)

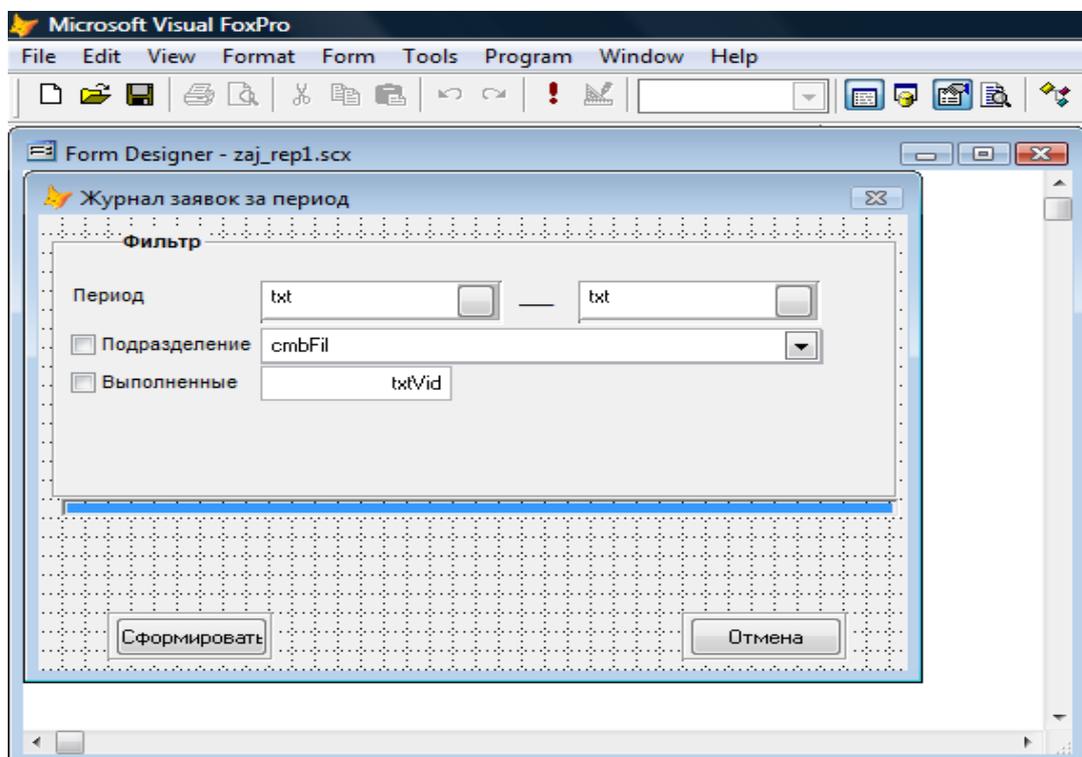


Рис. 3.10 Экран разработки формы отчета

блок рабочих отчетов – (вкладка Documents- REPORTS)

Отчёт – используется для вывода определенного набора информации ,либо же предназначены для форматирования, вычисления итогов и печати важных данных.

Отчеты создаются с помощью Конструктор отчетов (рис.3.10), позволяющий создавать и настраивать отчеты в формате VFP.

Одной из важнейших характеристик отчета является возможность группировать важную информацию для получения итоговых данных для групп и всего отчета. При формировании отчетов можно задать фильтры для выборки нужных данных либо сформировать отчет на основе данных SQL-запроса или представления данных (View).

Наиболее просто для разработки основы отчета воспользоваться Мастером отчетов с последующей модификацией и дополнением отчета в Конструкторе. Это можно увидеть на (рис. 3.11)

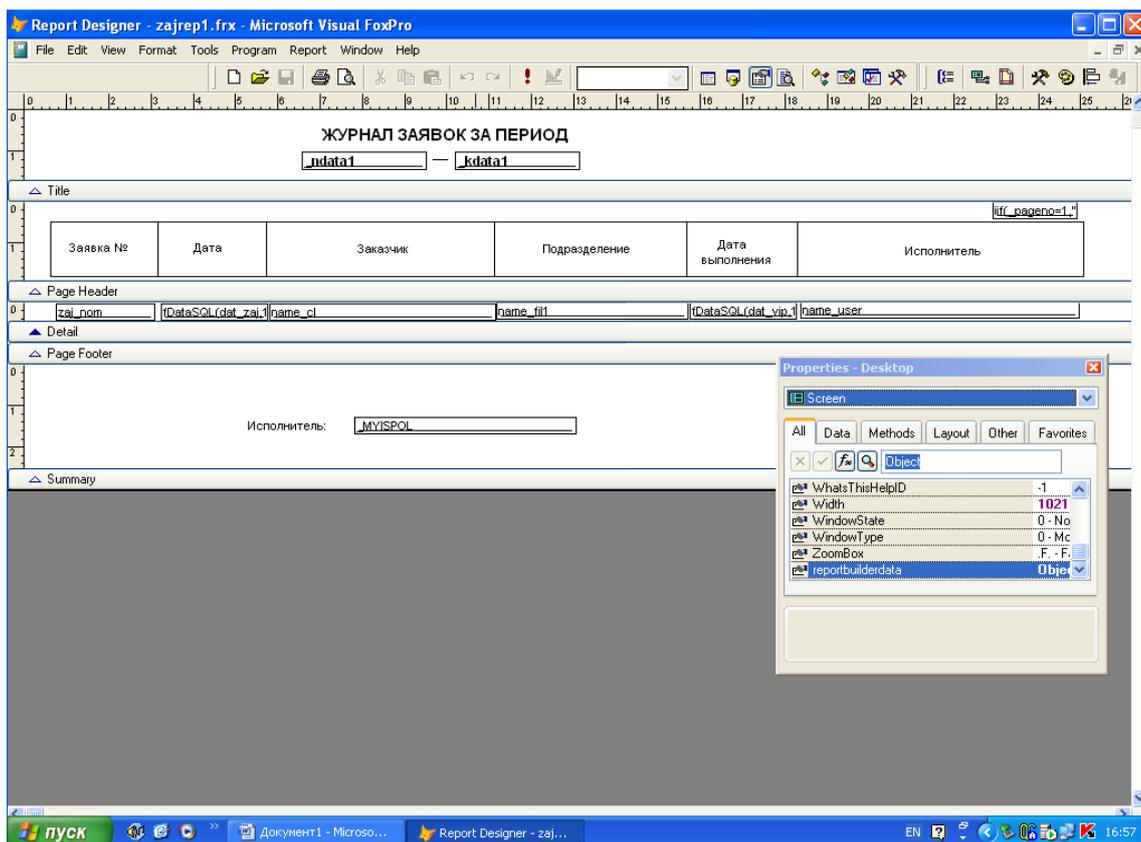


Рис. 3.11 Экран разработки отчета

Физически клиентское приложение представляет собой файл ITCONR.EXE.

Для создания EXE-файла используется специальное диалоговое окно (рис. 3.12).

Для нормальной работы программы необходимо установить в системе библиотеки VFP 9.0.

Для этого создается инсталляционная программа.

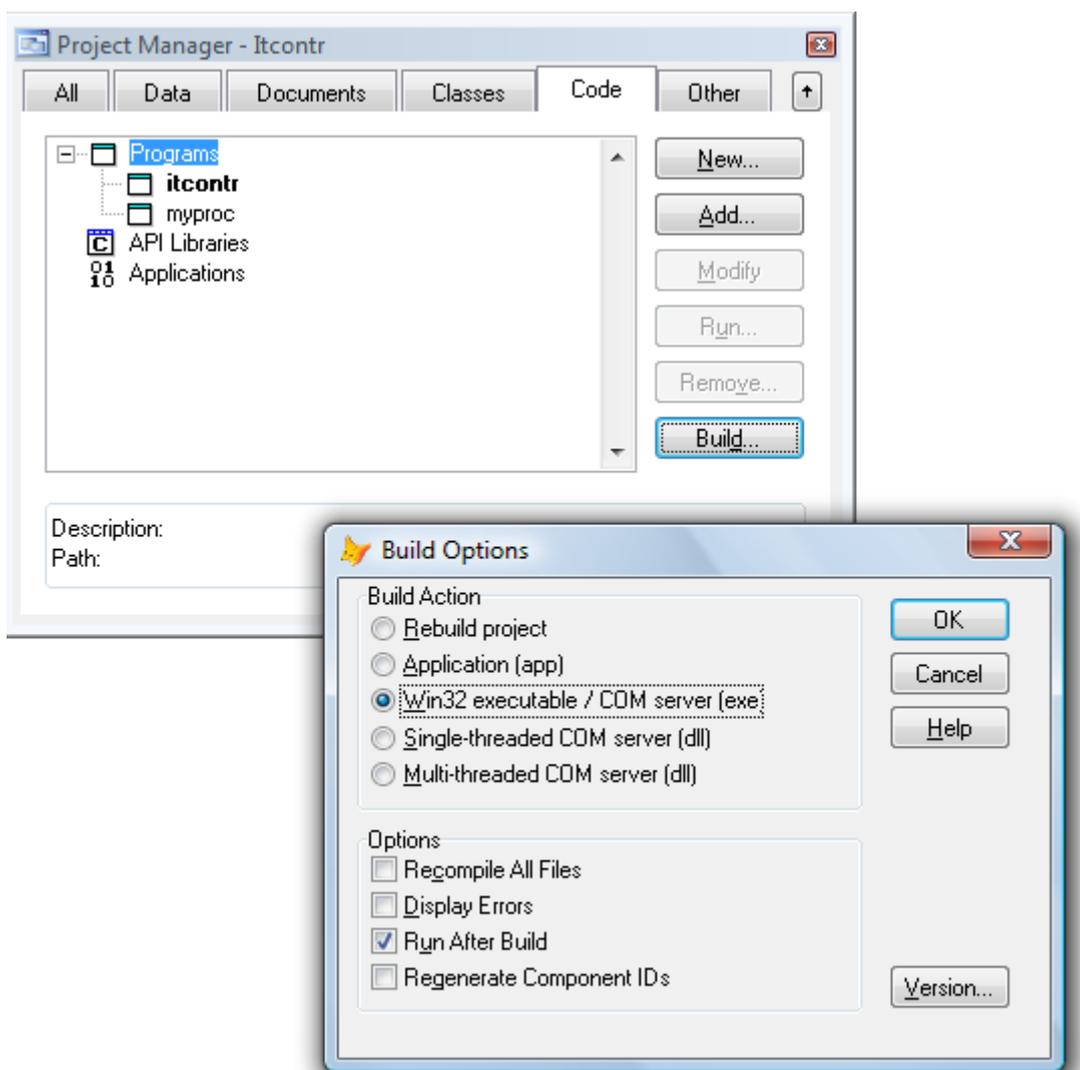


Рис. 3.12 Экран создания EXE-файла

Фрагменты программного кода приложения АИС, экранные формы и инструкция пользователя приведены в приложении А.

3.4 Тестирование АИС

Тестирование АИС является завершающим этапом ее разработки.

Если рассматривать все этапы тестирования, то можно заметить, что и этап анализа, и этап проектирования содержат элементы работы тестеров, например, для получения сразу нескольких удачных или удовлетворительных обоснований выбора того или иного решения, а также для оценки критериев качества получаемой системы.

Данные полученные в результате тестирования, и их характеристики, переданные группами тестирования, могут отображать, что выбор подхода реализации, на этапе определения стратегии возможно является неверным либо то, что выбранные средства реализации не удовлетворяют тем или иным требованиям АИС.

Такие же данные могут быть получены относительно выбора аппаратной платформы и операционной системы. Получение подобных результатов инициирует изменение данных, полученных на этапе определения стратегии, например пересчитывается смета затрат на проект.

Как правило, на этапе анализа проекта привлекаются группы тестирования, например для получения сравнительных характеристик предполагаемых к использованию аппаратных платформ, операционных систем, СУБД, иного окружения. Кроме того, на данном этапе определяется план работ по обеспечению надежности информационной системы и ее тестирования.

Для любых проектов целесообразным является привлечение тестеров на ранних этапах разработки, в частности на этапе анализа и проектирования. Если проектное решение оказалось неудачным и это обнаружено слишком поздно - на этапе разработки или, что еще хуже, на этапе внедрения в эксплуатацию, - то исправление ошибки проектирования может обойтись очень дорого.

Выводы к 3-й главе

Третья, заключительная, глава представляет собой описание физического проектирования информационной системы. Был сделан выбор архитектуры информационной системы, была выбрана технология разработки программного обеспечения, после чего был сделан выбор СУБД и осуществлена разработка физической модели данных под выбранную СУБД, рассмотрен основной функционал разработанной информационной системы, в заключении было проведено тестирование разработанной информационной системы.

ЗАКЛЮЧЕНИЕ

В условиях кризиса и усиливающейся конкуренции между предприятиями невозможно добиться успеха без создания и развития гибкой и масштабируемой ИТ - информационных технологий.

В этой связи возрастает заинтересованность каждого предприятия в развитии собственной ИТ - службы, на которую возлагаются обязанности по сопровождению ИТ - инфраструктуры и оказанию ИТ- услуг подразделениям предприятия, оформляемых в виде соответствующих заявок.

Настоящая дипломная работа посвящена актуальной теме разработки автоматизации ИТ - отдела коммерческого предприятия ООО «Золотой берег».

Объектом исследования дипломной работы являются бизнес - процессы инвентаризации и сопровождения аппаратного и программного обеспечения в отделе информационных технологий ООО «Золотой берег».

Предметом исследования является автоматизированная информационная система (АИС) отдела информационных технологий ООО «Золотой берег».

Область применения - информационно-технические подразделения ООО «Золотой берег» занимающиеся учетом оборудования и внедрением современных технологий на предприятии.

Дипломная работа выполнена на основе материалов преддипломной практики, проведенной в ООО «Золотой берег», которое занимается продажами кондитерских изделий.

В работе использована методология трехуровневого проектирования фактографических АИС.

В ходе решения вышеперечисленных задач и проведенных исследований были получены следующие результаты:

- произведен анализ предметной области и на основе структурного подхода разработана концептуальная модель АИС;
- на базе методологии UML разработана логическая модель АИС;
- реализовано программное обеспечение АИС.

АИС реализована в двухзвенной архитектуре «клиент-сервер» в среде VFP 9.0.

В качестве СБД использована СУБД MS SQL 2008.

К достоинствами АИС можно отнести простоту в эксплуатации и сопровождения, а также низкую стоимость владения.

Разработанная АИС прошла успешное тестирование и в настоящее время проходит опытную эксплуатацию в ИТ - отделе ООО «Золотой берег».

Предлагаемая АИС может быть использована для автоматизации ИТ-отделов предприятий малого бизнеса.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Мусина, Т.В. VisualFoxPro 7.0. Учебный курс [Электронный ресурс]:учебник /Т.В.Мусина,В.А. Пушенко.-М.:Век +,BookStar,2005.-736с.- ISBN 978-57931-0556-9.

2. Данелян, Т.Я. Экономические информационные системы (ЭИС) предприятий и организации [Электронный ресурс]:монография/Т.Я.Данелян.- М.:ЮНИТИ,2015.-284с.-ISBN 978-5-238-02736-4.

3. Илюшчкин, В.М. Основы использования и проектирования баз данных[Электронный ресурс]:учебное пособие для вузов/В.М.Илюшечкин.- М.:Высшее образование,2012.-224с.-ISBN 978-5-9692-0253-5.

4. Репин, В.В. Бизнес-процессы.Моделирование,внедрение,управление [Электронный ресурс]:учебник /В.В.Репин.-М.:Манн,Иванов и Фербер,2013.- 512с.-ISBN 978-5-91657-521-7.

5. Вахрушина, М.А. Учет на предприятиях малого бизнеса[Электронный ресурс]:Учебное пособие / М.А. Вахрушина, Л.В.Пашкова.-М.:ИНФРА-М,2012.-381с.-ISBN 978-5-9558-0130-8.

6. Лукин, В.Н. Введение в проектирование баз данных [Электронный ресурс]:Учебное пособие / В.Н. Лукин.-М.:Вузовская книга,2015.-144с.-ISBN 978-5-9502-0761-7.

7. Афонин, В.В. Моделирование систем[Электронный ресурс]:Учебно-практическое пособие /В.В. Афонин, С.А.Федосян.- М.:Бином,2016.-231с.-ISBN 978-5-9963-0352-6.

Электронные ресурсы

8. IDEF0 /[Электронный ресурс]:<https://ru.wikipedia.org/wiki/IDEF0>, (дата обращения: 29.04.2017).

9. Знакомство с нотацией IDEF0 и пример ее использования
/[Электронный ресурс]:<https://habrahabr.ru/post/322832/>, (дата обращения:
30.04.2017).

10. Концептуальная модель
/[Электронный ресурс]:https://ru.wikipedia.org/wiki/Концептуальная_модель, (дата
обращения: 25.04.2017).

11. Создание логической модели
/[Электронный ресурс]:<http://itteach.ru/bpwin/sozдание-logicheskoy-modeli>, (дата обращения:
20.04.2017).

12. MicrosoftSQLServer/[Электронный
ресурс]:https://ru.wikipedia.org/wiki/Microsoft_SQL_Server, (дата обращения:
15.04.2017).

Литература на иностранном языке

13. Seidl M., Scholz M., Huemer C., Kappel G., UML @ Classroom, 2015.

14. Itzik Ben-Gan, T-SQL Fundamentals for Microsoft SQL Server 2012
and SQL Azure, Microsoft Corporation, 2012.

15. Pramod J. Sadalage, Martin Fowler, NOSQL Distilled-A Brief Guide to the
Emerging World of Polyglot Persistence, Addison-Wesley, 2012.

16. Schwartz B., and others. High Performance MySQL.-3th Edition, 2012.

17. Alan Mark Davis. Just Enough Requirements Management: Where
Software Development Meets Marketing.-Dorset House, 2015.

ПРИЛОЖЕНИЕ А

Фрагменты программного кода АИС

Коды T-SQL для создания таблиц базы данных

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[SM_IDENTS]') and  
OBJECTPROPERTY(id, N'IsUserTable') = 1)
```

```
drop table [dbo].[SM_IDENTS]
```

```
GO
```

```
CREATE TABLE [dbo].[SM_IDENTS] (
```

```
    [Name_tbl] [varchar] (25) COLLATE Cyrillic_General_CI_AS NOT NULL ,
```

```
    [code_max] [int] NOT NULL
```

```
) ON [PRIMARY]
```

```
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[SPR_CLIENTS]') and  
OBJECTPROPERTY(id, N'IsUserTable') = 1)
```

```
drop table [dbo].[SPR_CLIENTS]
```

```
GO
```

```
CREATE TABLE [dbo].[SPR_CLIENTS] (
```

```
    [client_id] [char] (13) COLLATE Cyrillic_General_CI_AS NOT NULL ,
```

```
    [name_cl] [char] (100) COLLATE Cyrillic_General_CI_AS NULL ,
```

```
    [filial_id] [char] (3) COLLATE Cyrillic_General_CI_AS NULL ,
```

```
    [Time_Stamp] [timestamp] NOT NULL
```

```
) ON [PRIMARY]
```

```
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[SPR_USERS]') and  
OBJECTPROPERTY(id, N'IsUserTable') = 1)
```

```
drop table [dbo].[SPR_USERS]
```

```
GO
```

```

CREATE TABLE [dbo].[SPR_USERS] (
    [user_id] [char] (13) COLLATE Cyrillic_General_CI_AS NOT NULL ,
    [name_user] [varchar] (50) COLLATE Cyrillic_General_CI_AS NOT NULL ,
    [login] [varchar] (10) COLLATE Cyrillic_General_CI_AS NOT NULL ,
    [filial_id] [varchar] (3) COLLATE Cyrillic_General_CI_AS NOT NULL ,
    [profi_id] [smallint] NULL ,
    [time_stamp] [timestamp] NOT NULL ,
    [f_block] [bit] NULL ,
    [f_shef] [bit] NULL
) ON [PRIMARY]

GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[FK_SPR_USERS_ZURZAJIS]') and OBJECTPROPERTY(id,
N'IsForeignKey') = 1)

ALTER TABLE [dbo].[SPR_USERS] DROP CONSTRAINT FK_SPR_USERS_ZURZAJIS

GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[ZURZAJIS]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)

drop table [dbo].[ZURZAJIS]

GO

CREATE TABLE [dbo].[ZURZAJIS] (
    [zaj_id] [char] (13) COLLATE Cyrillic_General_CI_AS NOT NULL ,
    [zaj_nom] [char] (15) COLLATE Cyrillic_General_CI_AS NULL ,
    [client_id] [char] (13) COLLATE Cyrillic_General_CI_AS NULL ,
    [user_id] [char] (13) COLLATE Cyrillic_General_CI_AS NULL ,
    [content] [varchar] (500) COLLATE Cyrillic_General_CI_AS NULL ,
    [status] [int] NULL ,
    [dat_zaj] [datetime] NULL ,
    [dat_oz] [datetime] NULL ,

```

```

        [dat_vip] [datetime] NULL ,
        [f_vip] [bit] NULL ,
        [vt_id] [int] NULL ,
        [Time_Stamp] [timestamp] NULL
    ) ON [PRIMARY]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[ROLES]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)

drop table [dbo].[ROLES]

GO

CREATE TABLE [dbo].[ROLES] (
        [profi_id] [smallint] NOT NULL ,
        [name_pro] [char] (50) COLLATE Cyrillic_General_CI_AS NULL
) ON [PRIMARY]
GO

```

Фрагменты программного кода приложения

Главная процедура приложения

```

CLEARALL
CLOSEALL
SET SYSMENU OFF
SET TALK OFF
SET SAFETY OFF
SET DELETED ON
SET EXCLUSIVE OFF
SET BELL OFF
SET NOTIFY OFF
SET DATE GERMAN
SET CENTURY ON
SET MULTLOCKS ON
SET EXACT ON
SET ESCAPE OFF
SET CONSOLE OFF
SET COMPATIBLE OFF
SET STRICTDATE TO 0

```

```
SET DECIMALS TO 2
```

```
SET PROCEDURE TO myproc
```

```
SET CLASSLIB TO smpolis,fcontrol,baseobj,thermbar,webexplorer,classlib
```

```
PUBLIC txtVerProg,txtYearVersion, pNameOdbc, gnConnHandle , gnConnPict
```

```
PUBLIC _mnOper,_mnUtv,_mnAdm
```

```
PUBLIC path1,oPriPer,oflb,_VerTer
```

```
PUBLIC _nFirma,_cFirmName,_cFilialName,_myUser,_myIspol,;
```

```
_SearchContext,_LizoContext,_Kont_Serch, _STATPOLROWS
```

```
PUBLIC _SMSETTBL,_CFG_NAME
```

```
PUBLIC pNameDb
```

```
pNameDb="ITCONTR_DB"
```

```
txtVerProg="01.01.10"
```

```
pNameOdbc="ITCONTRDB"
```

```
oPriPer=CREATEOBJECT("bcApplication")
```

```
oflb=createobj("sm_tool")
```

```
WITH oPriPer
```

```
.cMainMenu="itc_menu.mpr"
```

```
.cMainWinTitle="АИСИТ-отдела"+" "+ txtVerProg
```

```
.cMainWinIcon="Z_PC024.ICO"
```

```
ENDWITH
```

```
WITH _screen
```

```
.Caption=oPriPer.cMainWinTitle
```

```
.BackColor=rgb(128,128,128)
```

```
.BorderStyle=2
```

```
.Closable=.F.
```

```
.MaxButton=.T.
```

```
.MinButton=.T.
```

```
.Movable=.T.
```

```
.Icon=oPriPer.cMainWinIcon
```

```
.windowstate=2
```

```
ENDWITH
```

```
_screen.visible=.t.
```

```
***проверказапуска***
```

```
cEvent='INCONTR.EXE'
```

```
DECLARE INTEGER OpenEvent IN Win32Api ;
```

```
INTEGER, INTEGER, STRING
```

```
DECLARE INTEGER CreateEvent IN Win32Api ;
```

```
INTEGER, INTEGER, INTEGER, STRING
```

```
IF OpenEvent(2, 0, cEvent) != 0
```

```
=MESSAGEBOX("Приложениеужезапущено", 0 + 16 + 0, "")
```

```
CLEAR EVENTS
```

```
QUIT
```

```
ENDIF
```

```
CreateEvent(0, 0, 0, cEvent)
```

```
oflb.dock(0)
```

```
oflb.show()
```

```
oPriPer.do()
```

```
CLEAR DLLS
```

```
CLEAR ALL
```

RELEASE ALL EXTENDED

```
FUNCTION f_Quit
#DEFINE MSGBOX_YES          6
#DEFINE C_MSGBOX1          36
#DEFINE C_TITLE            oPriPer.cmainwintitle
LOCAL lCur,lSql
IF MESSAGEBOX("Завершить работу?",C_MSGBOX1,C_TITLE) = MSGBOX_YES
CLEAR EVENTS
QUIT
ENDIF
ENDFUNC
```

```
PROCEDURE f_OpenDbs
LOCAL _fDbf1,_Dbf1,_lCur,_cnt1,_lFlag

_Dbf1="ITCONRDB"
_fDbf1=SYS(5)+SYS(2003)+"\"+_Dbf1+".dbf"
IF _pubDbs=1
IF !FILE(_fDbf1)
=MESSAGEBOX("Не найден файл "+_fDbf1,48,oPriPer.cmainwintitle)
=f_CloseApp()
ELSE
_lCur=CursorName()
USE &_fDbf1 IN 0 ALIAS &_Dbf1 EXCLUSIVE
SELECT (_Dbf1)
COUNT TO _cnt1
DO CASE
CASE _cnt1=0
=MESSAGEBOX("Нет зарегистрированных баз данных! ",48,oPriPer.cmainwintitle)
=f_CloseApp()
CASE _cnt1=1
SELECT (_Dbf1)
GO TOP
pNameDb=ALLTRIM(Name_Db)
pNameOdbc=ALLTRIM(Name_Odbc)
pNamePict=ALLTRIM(Name_OdbcP)
OTHERWISE
pNameDb=""
pNameOdbc=""
pNamePict=""
ENDCASE

USE IN (_Dbf1)
ENDIF
ENDIF
ENDPROC
```

```
PROCEDURE f_CloseApp
close all
clea events
```

```
cancel
quit
ENDPROC
```

```
*****
*      CursorName
*      Создает имя уникальное курсора
*****
FUNCTION CursorName
LOCAL lcName
lcName=SYS(2015)
DO WHILE USED(lcName)
    lcName=SYS(2015)
ENDDO
RETURN lcName
```

```
FUNCTION _Showpanel
oflb.refrtoolbar(.t.,.t.)
ENDFUNC
```

```
FUNCTION _Hidepanel
oflb.refrtoolbar(.f.,.f.)
ENDFUNC
```

```
DEFINE CLASS HeadColumn AS Header
Visible=.T.
ForeColor=0
FontBold=.f.
PROCEDURE Click
LOCAL pch
pch=thisform.p_cHead
IF pch
thisform.CurrentGrid.SetAll("Picture","", "HeadColumn")
this.Picture="tdn.bmp"
thisform.Refresh
thisform.sql_order=ALLTRIM(this.Tag)
thisform.Grid_Refresh
ELSE
header::click()
ENDIF
ENDPROC
ENDDEF
```

```
DEFINE CLASS myText1 AS TextBox
BackColor=RGB(255,255,255)
DisabledBackColor=RGB(255,255,255)
DisabledForeColor=RGB(128,128,128)
ForeColor=RGB(0,0,0)
SelectedBackColor=RGB(10,36,106)
SelectedForeColor=RGB(255,255,255)
SelectOnEntry=.F.
```

```
BorderStyle=0
Margin=0
Visible=.T.
```

```
PROCEDURE GotFocus
TextBox::GotFocus
ENDPROC
```

```
PROCEDURE DblClick
LOCAL lObj
lObj=this.Tag
IF &lObj.Commandgroup1.cmdEdit.Enabled
&lObj.Commandgroup1.cmdEdit.Click
ENDIF
ENDPROC
```

```
PROCEDURE KeyPress
LPARAMETERS nKeyCode, nShiftAltCtrl
LOCAL lObj
lObj=this.Tag
IF nKeyCode=13 AND nShiftAltCtrl=0
IF &lObj.Commandgroup1.cmdEdit.Enabled
&lObj.Commandgroup1.cmdEdit.Click
ENDIF
ENDIF
ENDDEFINE
```

```
PROCEDURE MsgErr
LPARAMETERS _pText, _pZnak,m
```

```
m1=messagebox(ALLTRIM(_pText),_pZnak,oPriPer.cmainwintitle)
RETURN m1
```

```
DEFINE CLASS TextGrid AS TextBox
BackColor=RGB(255,255,255)
DisabledBackColor=RGB(255,255,255)
DisabledForeColor=RGB(128,128,128)
ForeColor=RGB(0,0,0)
SelectedBackColor=RGB(10,36,106)
SelectedForeColor=RGB(255,255,255)
SelectOnEntry=.F.
BorderStyle=0
Margin=0
Visible=.T.
```

```
PROCEDURE GotFocus
TextBox::GotFocus
ENDPROC
```

```
PROCEDURE DblClick
IF ThisForm.P_AddEdit=2 AND !EMPTY(ALLTRIM(ThisForm.p_FormFon))
ThisForm.Commandgroup1.cmdSelect.Click
```

```

ELSE
IF ThisForm.Commandgroup1.cmdEdit.Enabled
ThisForm.Commandgroup1.cmdEdit.Click
ENDIF
ENDIF
ENDPROC
    ENDDDEFINE

```

Фрагменты бизнес-логики АИС

Хранимая процедура добавления заявки

CREATEPROCEDUREspk_AddZaj

```

    @zaj_nom CHAR(15), @client_id CHAR(13),@user_id CHAR(13), @content
CHAR(200), @status INT,
    @dat_zaj DATETIME, @dat_oz DATETIME,@dat_vip DATETIME,@f_vip
BIT,@vt_id INT, @Filial_Id CHAR(3)
AS
DECLARE @lnNum INT, @StrNext_IdCHAR(13), @F_Upd BIT
SET @F_Upd=1

EXEC Sm_NewId 'zurzajs',@Filial_Id, @StrNext_Id OUT

INSERT INTO zurzajs
(zaj_nom,client_id,user_id,content,status,dat_zaj,dat_oz,dat_vip,f_vip,vt_id,zaj_id )
VALUES
(@zaj_nom,@client_id,@user_id,@content,@status,@dat_zaj,@dat_oz,@dat_vip,@f_vip,@vt
_id, @StrNext_Id)
SELECT zurzajs.*, 'F_Update'=@F_Upd,'Sm_Err'=0, 'Name_Err'="" FROM zurzajs
WHEREzaj_id=@StrNext_Id
GO

```

Хранимая процедура редактирования заявки

CREATEPROCEDUREspk_UpdZaj

```

    @zaj_nom CHAR(15), @client_id CHAR(13),@user_id CHAR(13), @content
CHAR(200), @status INT,
    @dat_zaj DATETIME, @dat_oz DATETIME,@dat_vip DATETIME,@f_vip
BIT,@vt_id INT, @zaj_idCHAR(13), @Time_Stamp TIMESTAMP

```

```

AS
DECLARE @Time_Stamp2 TIMESTAMP, @F_Upd BIT
SELECT @Time_Stamp2=Time_Stamp FROM zurzajs WHEREzaj_id=@zaj_id
BEGIN
IF @Time_Stamp2=@Time_Stamp
BEGIN
UPDATE zurzajs SETzaj_nom= @zaj_nom, client_id=@client_id,user_id=@user_id,
content= @content, status=@status,
dat_zaj=@dat_zaj, dat_oz=@dat_oz,dat_vip=@dat_vip,f_vip=@f_vip,vt_id=@vt_id
WHERE zaj_Id=@zaj_Id ANDTime_Stamp=@Time_Stamp
SELECT @F_Upd=1
END
ELSE
SELECT @F_Upd=0
END
SELECT zurzajs.*, 'F_Update'=' @F_Upd,'Sm_Err'=0, 'Name_Err'=' FROM zurzajs
WHEREzaj_id=@zaj_Id
GO

```

Хранимая процедура удаления заявки

```

CREATEPROCEDUREspk_DelZaj
@zaj_IdCHAR(13)
AS
DELETE FROM zurzajs WHEREzaj_id=@zaj_Id
GO

```