



МИНОБРНАУКИ РОССИИ

**федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования**

«Московский государственный технологический университет «СТАНКИН»

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

ЛАБОРАТОРНАЯ РАБОТА 4.

**РАЗРАБОТКА ПРОГРАММНОЙ И ТЕХНОЛОГИЧЕСКОЙ АРХИТЕКТУР
ИНФОРМАЦИОННОЙ СИСТЕМЫ**

Перед началом работы необходимо ознакомиться с материалами лекции «Информационные системы и технологии», посвященной дизайну архитектур информационных систем.

Лабораторная работа 4

ПОСТРОЕНИЕ ПРОГРАММНОЙ И ТЕХНОЛОГИЧЕСКОЙ АРХИТЕКТУР ИНФОРМАЦИОННОЙ СИСТЕМЫ

ЧАСТЬ 1. РАЗРАБОТКА ПРОГРАММНОЙ АРХИТЕКТУРЫ ИС

1.1 Цель работы

Разработать программную модель ИС:

- Функциональная архитектура приложения (дерево автоматизируемых функций)
- Структура диалога приложения
- Дерево программных модулей
- Поведенческая модель (алгоритм функционирования системы)



Разработку программной модели информационной системы необходимо выполнить в строгой синхронизации с моделями IDEF0, DFD, ERA, разработанными на 1 и 2 лабораторных работах по дисциплине.

1.2 Основные теоретические положения

Архитектура автоматизируемых бизнес-процессов(IDEF0, DFD) определяет функциональную архитектуру приложения (*совокупность функциональных подсистем и процедур*) в виде набора операций, функций, задач обработки информации, обеспечивающих реализацию бизнес-процессов.

Функциональная архитектура приложения может быть представлена деревом функций предметной области.

Дерево автоматизируемых функций.

Следует привести иерархию функций управления и обработки данных, которые призваны автоматизировать разрабатываемый программный продукт. При этом можно выделить и детализировать два подмножества функций:

- реализующих служебные функции (например, проверки па- роля, ведения календаря, архивации баз данных и др.)
- реализующих основные функции ввода первичной информации, обработки, ведения справочников, ответов на запросы и др. (рис. 1).

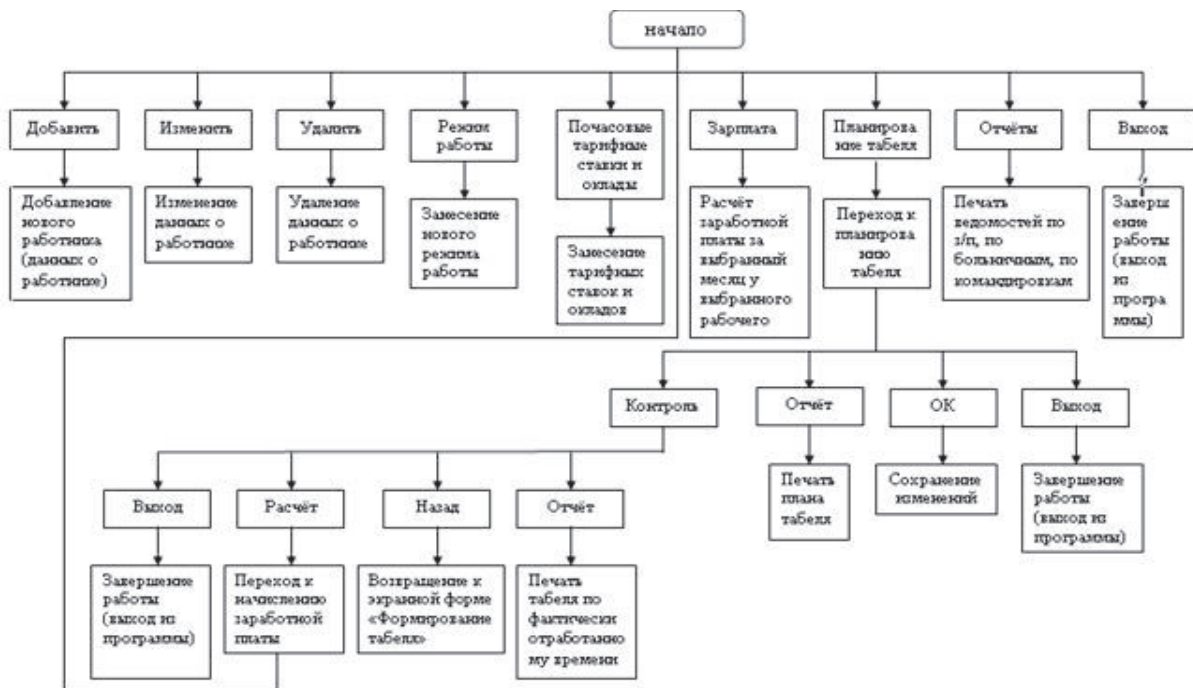


Рис. 1. Пример функциональной архитектуры

Разработка структуры диалога приложения.

В этом пункте следует выбрать способ описания диалога. Как правило, применяется два способа описания диалога.

- Первый предполагает использование табличной формы описания.
- Второй использует представление структуры диалога в виде орграфа, вершины которого могут быть перенумерованы, а описание его содержания в соответствии с нумерацией вершин, либо в виде экранов, если сообщения относительно просты, либо в виде таблицы. Пример приведен на рис.2.

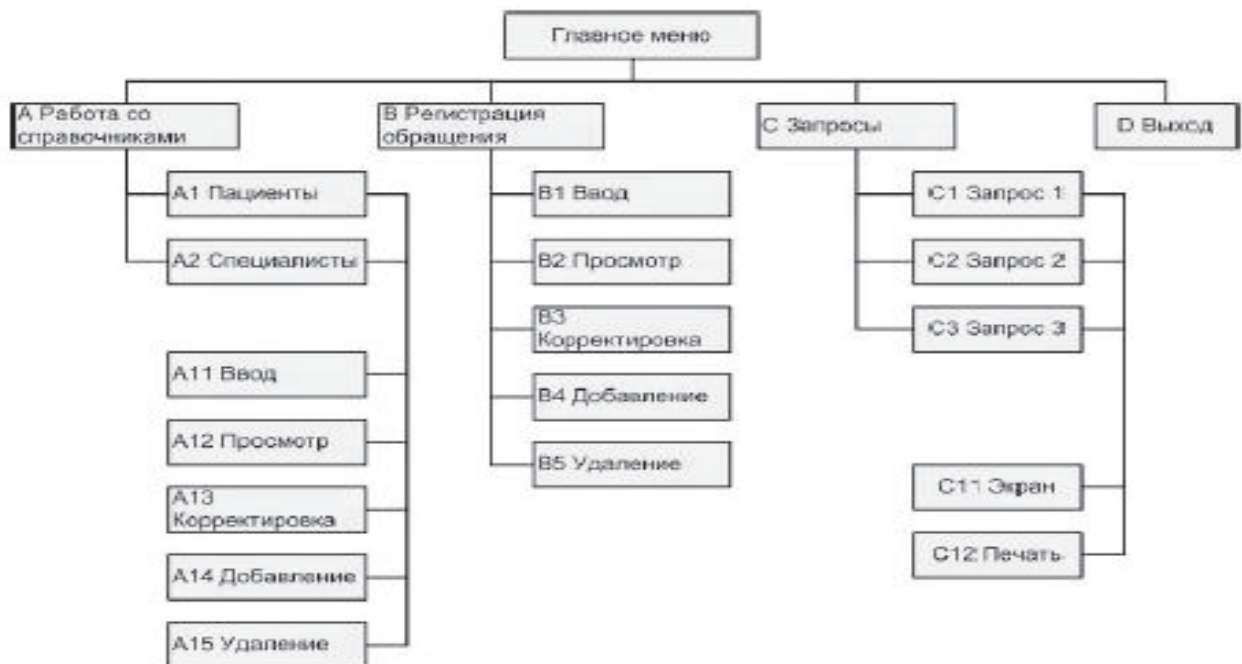


Рис. 2. Пример структуры диалога

Дерево программных модулей.

На основе результатов, полученных выше, строится дерево программных модулей, отражающих структурную схему пакета, содержащей программные модули различных классов:

- выполняющие служебные функции;
- управляющие модули, предназначенные для загрузки меню и передачи управления другому модулю;
- модули, связанные с вводом, хранением, обработкой и выдачей информации.

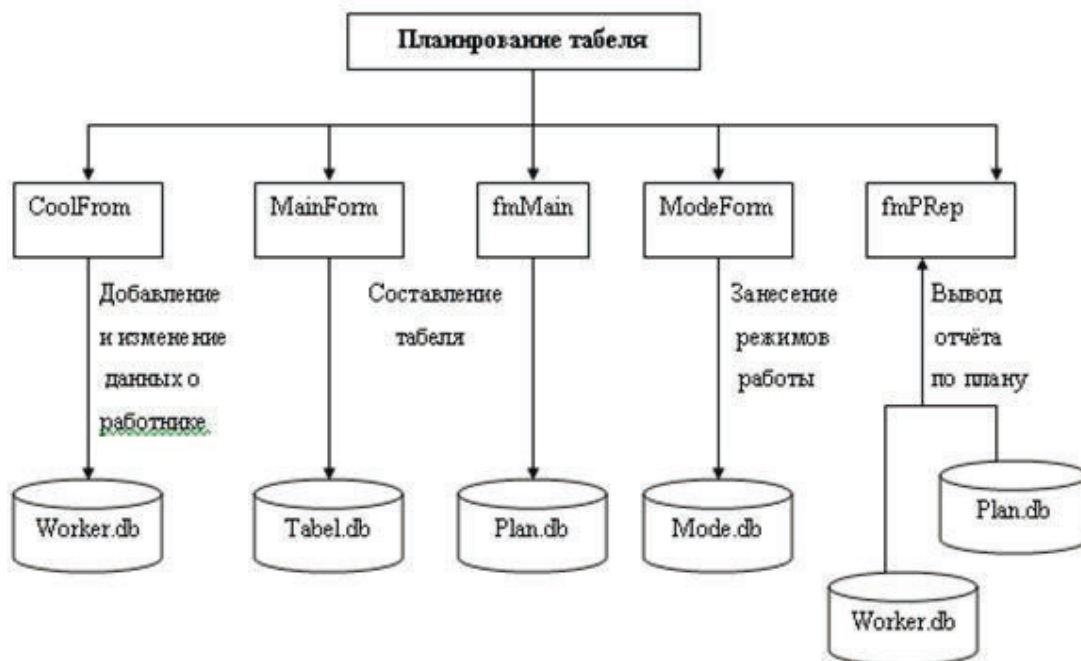


Рис. 3. Дерево программных модулей

Схема взаимосвязи программных модулей и информационных файлов

Схема отражает взаимосвязь программного и информационного обеспечения ИС. Пример на рис.3. В данном пункте необходимо для каждого модуля указать идентификатор и выполняемые функции, например, в виде:

Идентификатор Модуля	Выполняемые модулем функции
MainForm	Выбор пунктов главного меню
NewAppl	Поступление заявки на получение кредита
Add	Занесение клиента в базу данных

Поведенческая модель системы.








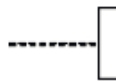

Наиболее известными и популярными способами и методологиями разработки поведенческих моделей являются:

- Блок-схемы алгоритмов;
- EPC –диаграммы (методология ARIS);
- Методология BPMN.

Для разработки блок-схем алгоритмов используются обозначения :

Описание символов
в схемах представления информационных технологий

№ п/п	Символ	Наименование	Примечания
1. СИМВОЛЫ ДАННЫХ			
<i>Основные</i>			
1.1		Данные	Данные, носитель которых не определен
1.2		Запоминающее устройство (ЗУ)	Данные, хранимые в виде, для автоматической обработки, носитель не определен
<i>Специфические</i>			
		ОЗУ	Данные, хранящиеся в ОЗУ (оперативная память)
1.4		ЗУ с последовательным доступом	Данные, хранящиеся на магнитной ленте
1.5		ЗУ с прямым доступом	Данные, хранящиеся на жестких или гибких магнитных дисках, CD, DVD, ZIP и т.д.
1.6		Документ	Данные, представляемые не в компьютерном виде (на бумаге, на пленках и т.д.)
1.7		Ручной ввод	Данные, вводимые вручную с помощью клавиатуры, мыши, светового пера и т. д.
1.8		Карта	Данные на перфокартах, магнитных картах, картах со считываемыми метками и т.д.
1.9		Бумажная лента	Данные на бумажной ленте
1.10		Дисплей	Данные, отображаемые на экране монитора, сигнальные индикаторы и т.д.
2. СИМВОЛЫ ПРОЦЕССА			
<i>Основной</i>			
2.1		Процесс	Элементарная (атомарная) операция по обработке данных (например, $n := n + 1$)
<i>Специфические</i>			
2.2		Предопределенный процесс (процедура)	Процесс, состоящий из операций, описанных в другом месте (на другой схеме)
2.3		Ручная операция	Операция, выполняемая вручную
2.4		Подготовка	Подготовительные операции, выполняемые с целью модификации последующих операций (цикл с параметром [For-To-Do])
2.5		Решение	Условная операция (операторы If-Else или Switch-Case)
2.6		Параллельные действия	Параллельное выполнения двух и более операций
2.7		Границы цикла	Начало и конец цикла. записывается в начале или конце, в зависимости от того, где осуществляется его проверка (циклы с пред- или постусловием)

3. СИМВОЛЫ ЛИНИЙ				
Основной				
3.1		Линия	Поток данных или управления	
Специфические				
3.2		Канал связи	Передача данных по каналу связи	
3.3		Пунктирная линия	Альтернативная связь между двумя и более символами или для обводки комментируемого участка схемы	
4. СПЕЦИАЛЬНЫЕ СИМВОЛЫ				
4.1	ГОСТ		Соединитель	Используется для обрыва линий и их продолжения в другом месте. Обычно используется для обозначения взаимосвязанных частей схемы на разных листах. Внутри соединителя пишется номер соединения
	ИСО			
4.2			Терминатор	Выход во внешнюю среду или вход из внешней среды (начало и конец процесса обработки данных [в этом случае внутри пишут "начало" или "конец"]), источник или пункт назначения данных, начало и
				конец работы предопределенного процесса)
4.3			Получатель – отправитель	По функциональному назначению аналогичен символу "Терминатор"
4.4			Комментарий	Символ используется для внесения пояснительных записей
4.5			Пропуск	Пропуск символа или группы символов

Некоторые примеры реализации функциональных архитектур:

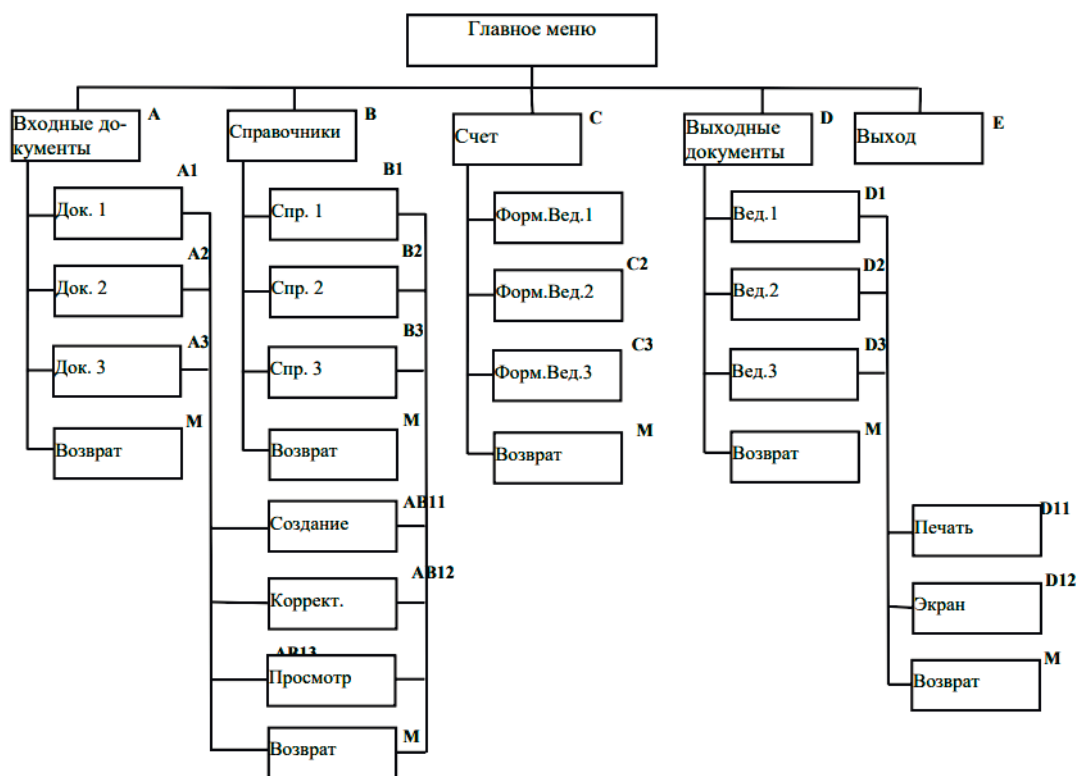


Схема работы системы

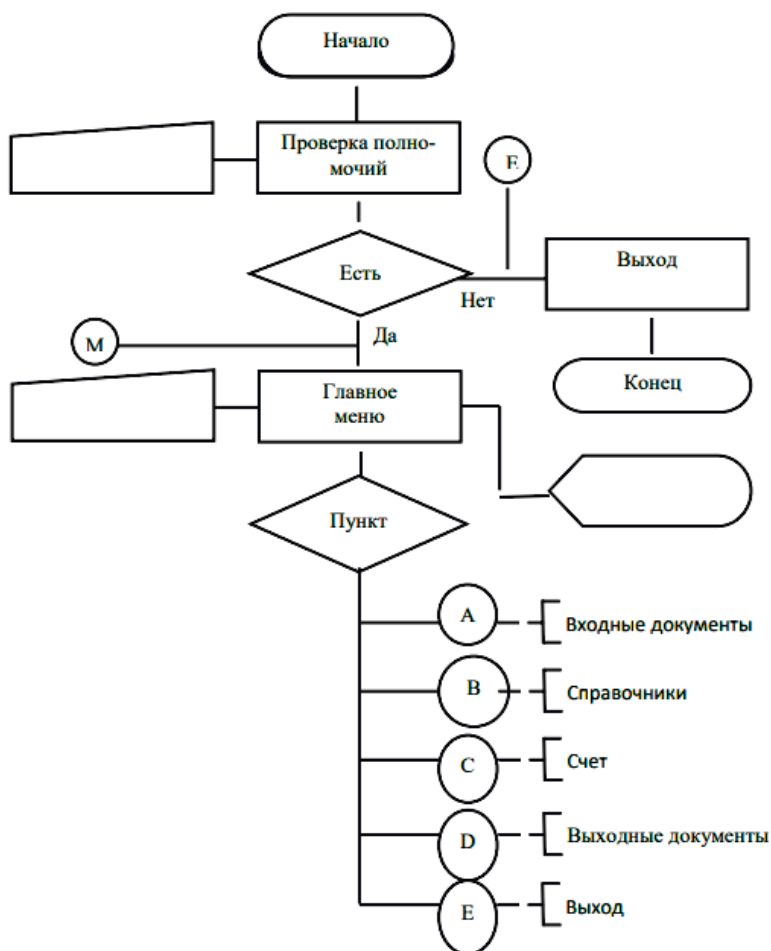


Схема работы системы (продолжение)

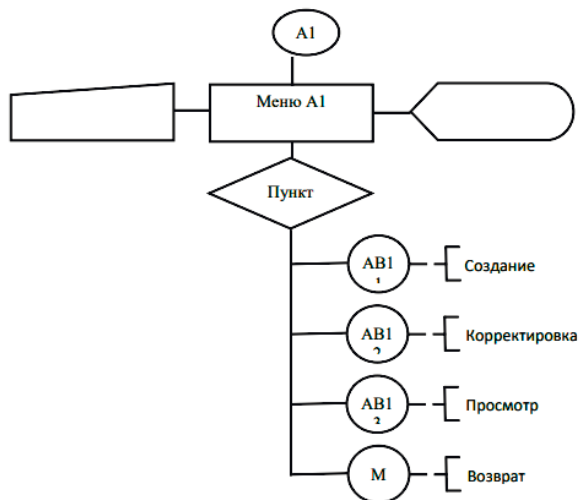
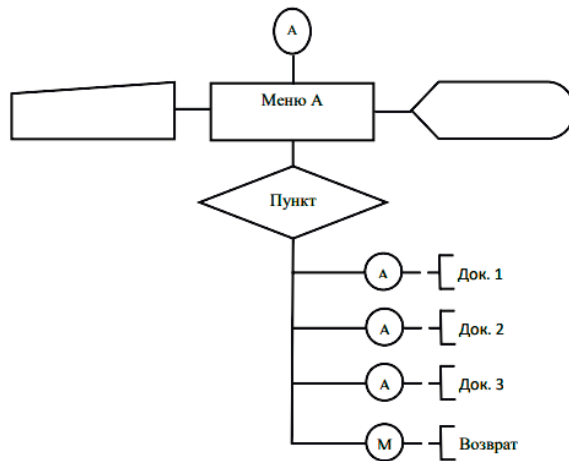


Схема работы системы (продолжение)

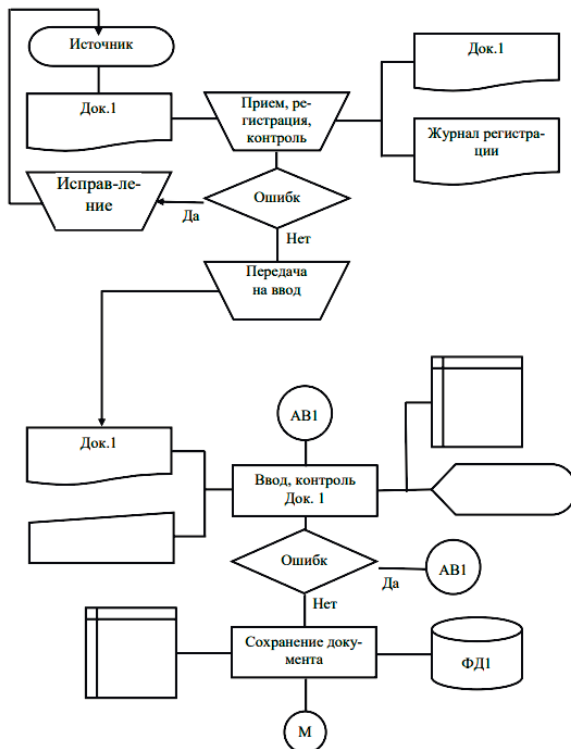


Схема работы системы (продолжение)

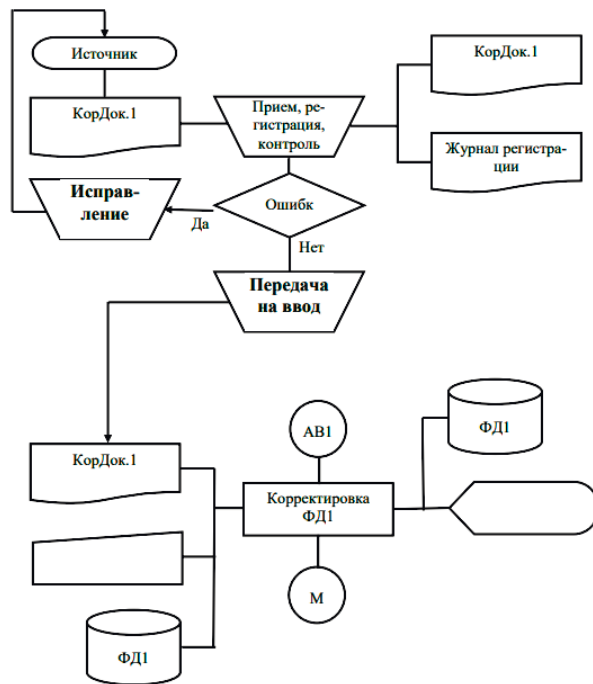


Схема работы системы (продолжение)

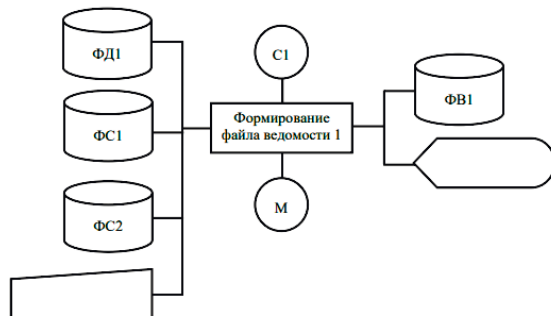
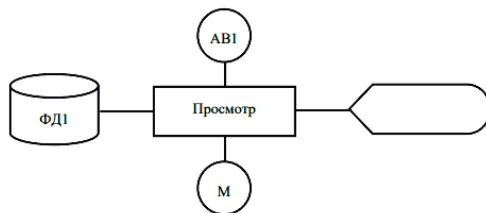


Схема данных

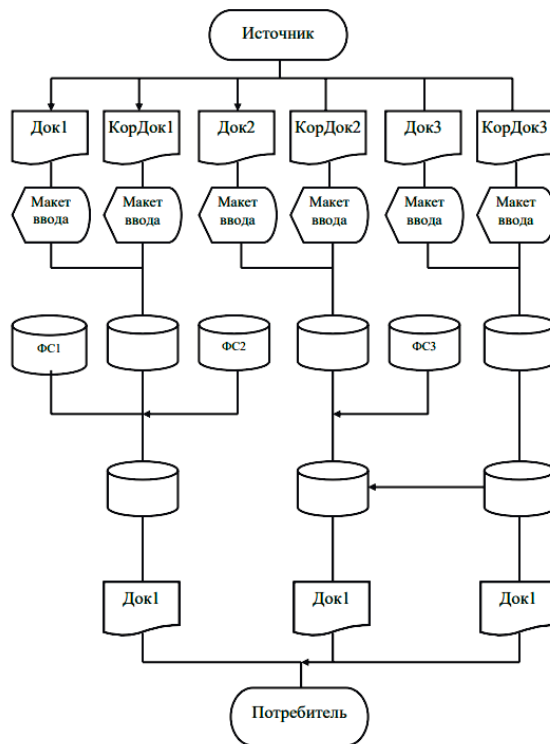


Схема взаимодействия программ

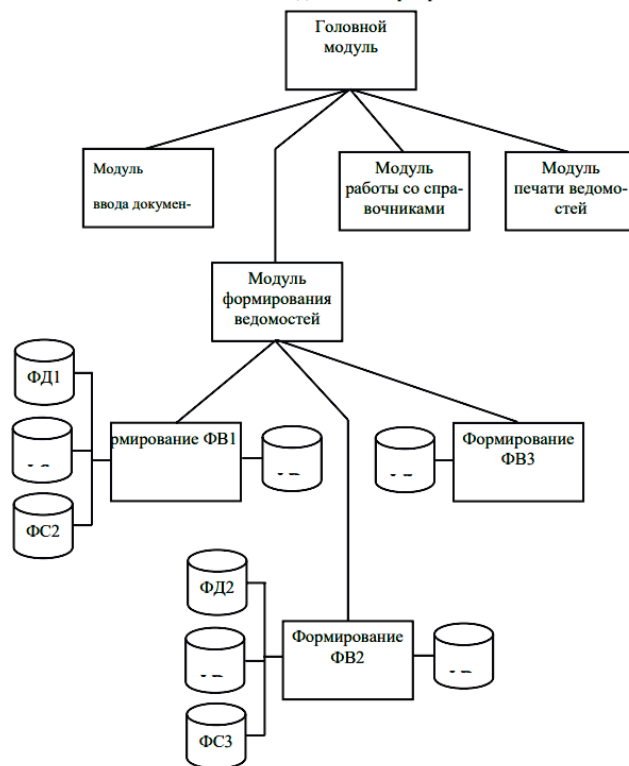
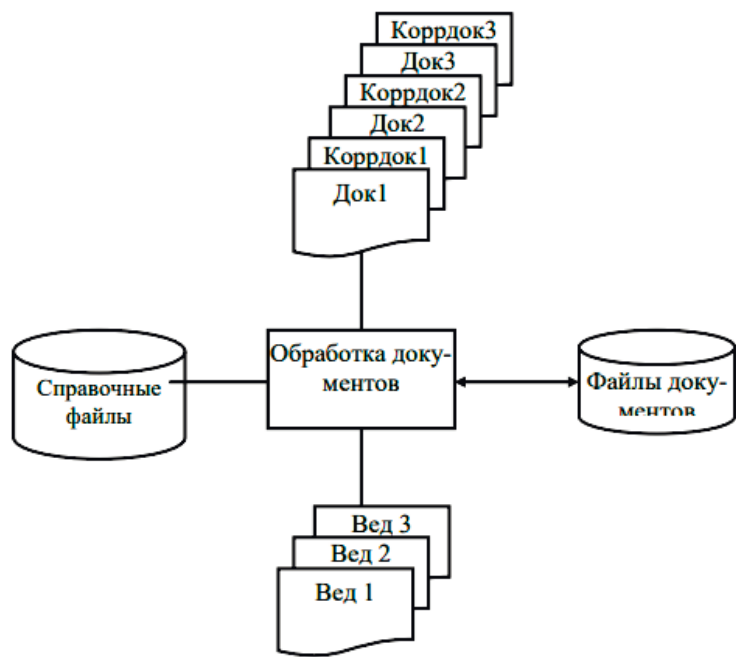


Схема ресурсов системы



ЧАСТЬ 2. РАЗРАБОТКА ТЕХНОЛОГИЧЕСКОЙ АРХИТЕКТУРЫ ИС

2.1 Цель работы

Получить навыки и освоить правила описание инфраструктуры и системных компонентов, требующихся для поддержки архитектуры приложений и информационной архитектуры.



Разработку технологической модели информационной системы необходимо выполнить в строгой синхронизации с моделями IDEF0, DFD, ERA, разработанными на 1 и 2 лабораторных работах по дисциплине.

2.2 Основные теоретические положения

Технологическая архитектура представляет собой логическое, не привязанное к конкретным производителям описание инфраструктуры и системных компонентов, требующихся для поддержки архитектуры приложений и информационной архитектуры.

В частности, она включает в себя следующие элементы:

- компьютеры и серверы;
- операционные системы;
- сетевые компоненты;
- принтеры;
- модемы и т.д. и т.п.

С точки зрения программно-аппаратной реализации можно выделить ряд типовых архитектур ИС.

Традиционные архитектурные решения основаны на использовании выделенных файл-серверов, 2-уровневой архитектуры серверов баз данных, 3-уровневой архитектуры с сервером приложений.

Существуют также варианты архитектур информационных систем, *базирующихся на технологии Internet* (Intranet-приложения). Разновидностью архитектуры информационной системы основывается на концепции "хранилища данных" (Data Warehouse) - интегрированной информационной среды, включающей разнородные информационные ресурсы. Для построения глобальных распределенных информационных приложений используется архитектура интеграции информационно-вычислительных компонентов на основе объектно-ориентированного подхода.

Технологическая архитектура рассматривает следующие компоненты:

- архитектуру инфраструктуры приложений;
- архитектуру управления;
- сетевую архитектуру;
- архитектуру хранения;
- архитектуру безопасности.

Работы по разработке технологической архитектуры должны начинаться с обследования имеющейся ИТ-инфраструктуры предприятия (учреждения) и определения её соответствия требованиям архитектуры приложений. Архитектура приложений представляет собой архитектуру всех бизнес-приложений для автоматизированных служб предприятия, в том числе описание механизма реализации в этой структуре функциональных требований организации.

Эксплуатационные требования к программной системе включают требования к безопасности, надежности, производительности и возможности взаимодействия с другим программным обеспечением. Например, служба доступна только авторизованным подписчикам, служба должна функционировать надлежащим образом 99,999 % времени.

Эффективная технологическая архитектура обеспечивает безопасность, доступность и надежность ПО, а хорошо спроектированной структуре приложения, в частности

соответствующей требованиям бизнес-процесса, должна соответствовать технологическая конфигурация и сетевое оборудование для обеспечения передачи данных. То есть архитектура приложений и технологическая архитектура неразрывно связаны друг с другом.

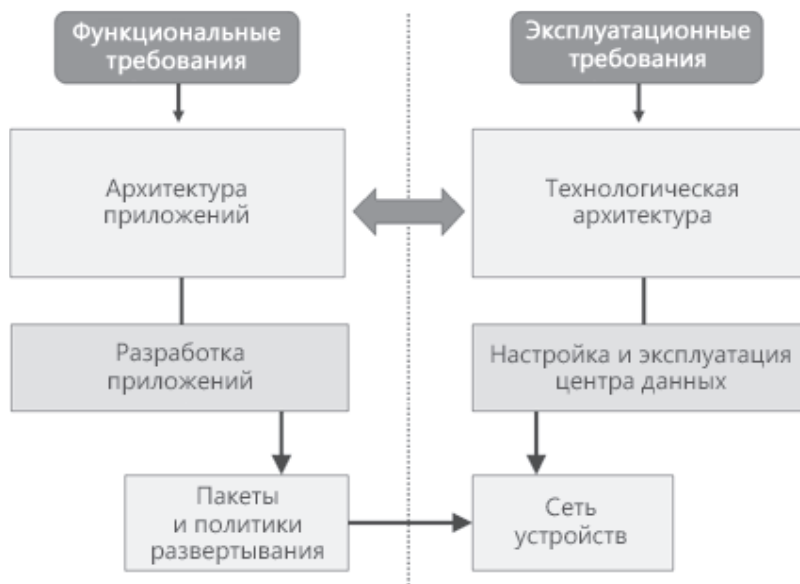


Рисунок 1- Связи между архитектурами

Для каждой архитектуры можно выделить несколько представлений: концептуальное, логическое и физическое.

Концептуальное представление обычно описывается в терминах, знакомых обычным пользователям (не ИТ-специалистам). Концептуальное представление используется для определения функциональных требований и представления приложения с точки зрения бизнес-пользователей в процессе создания бизнес-модели.

Логическое представление отражает основные функциональные компоненты системы и связи между ними независимо от технических подробностей реализации соответствующих функций. Архитекторы создают модели приложений, которые отражают логическое представление архитектуры приложения.

Физические представления наименее абстрактны и отражают компоненты конкретной реализации и связи между ними. Каждый из элементов в физическом представлении реализуется программно- или аппаратно-, как правило, в процессе проектирования или разработки.

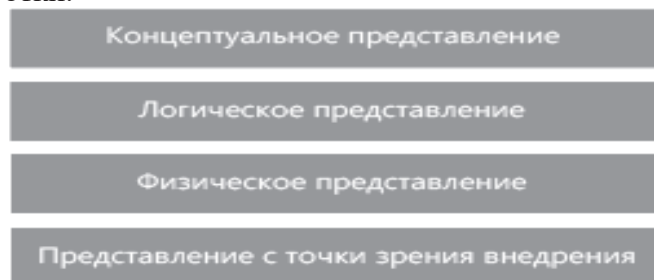


Рисунок 2- Архитектурные представления

При выборе технологической архитектуры ИС необходимо отразить тип многопользовательской архитектуры: файл-сервер или клиент-сервер, тип локальных сетей и сетевых операционных систем, а также типы ЭВМ для клиентской и серверной части архитектуры.

Пример:

Рассмотрим проектирование технологической архитектуры на примере: Компания ООО «5 Звезд» входит в состав торгового холдинга и занимается организацией демонстрации фильмов, продажа билетов, учетом поступлений от мероприятий. Среди партнеров компании ООО «5 Звезд» – крупные кино-концертные компании. Целями функционирования организации является осуществление реализации билетов, продвижение мероприятий, проведение маркетинговых исследований рынка, финансовая отчетность.

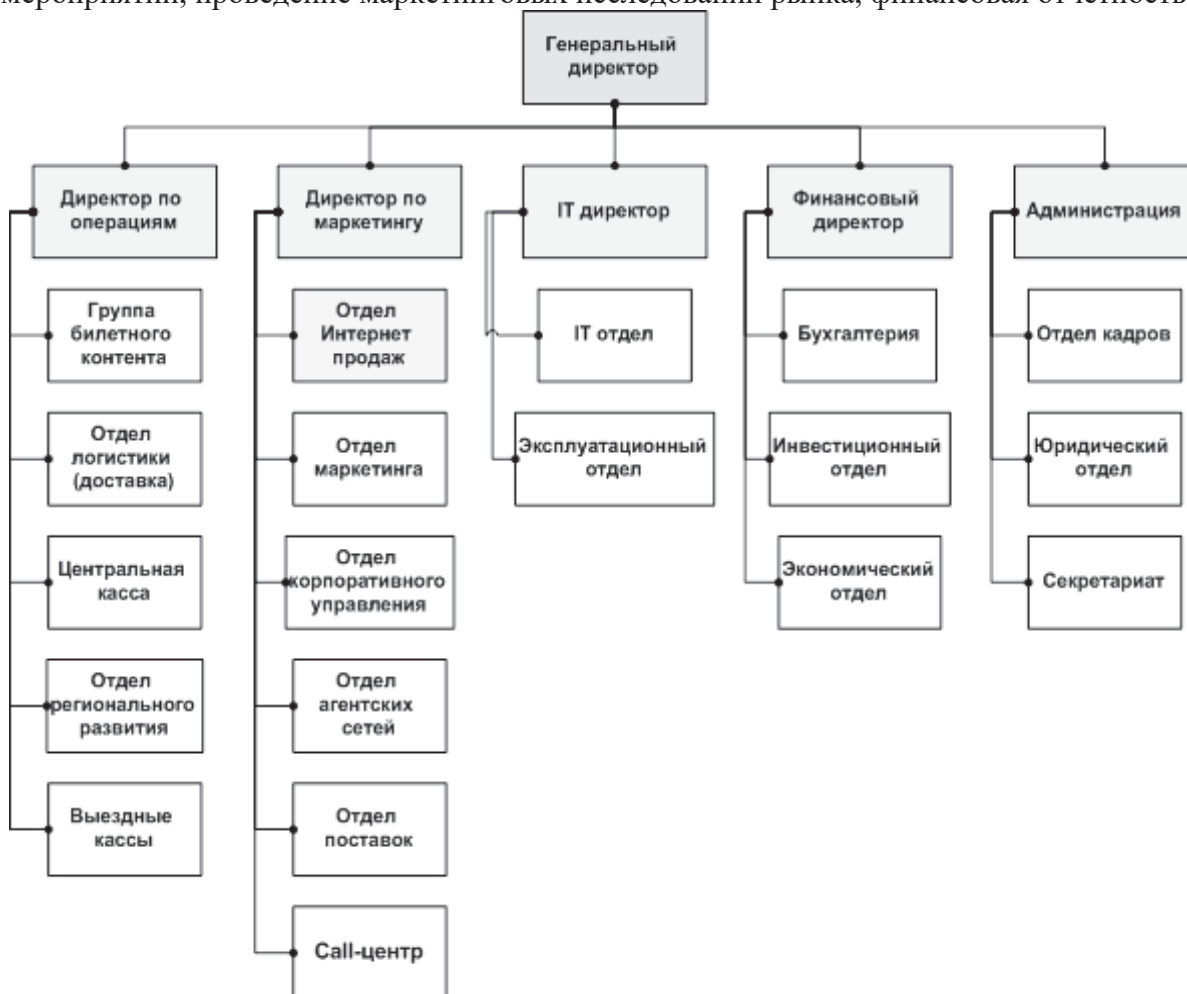


Рисунок 3-Схема общей организационной структуры управления предприятием

Организационная структура управления предприятием: компания ООО «5 Звезд» структурно состоит из пяти отделов, представленных на Рисунок – 3.

Компания ООО «5 Звезд» выполняет следующие виды деятельности (бизнес-функции):

Таблица 3- Виды деятельности компании

Функция управления	Код задачи	Наименование задачи	Назначение задачи	Исходная выходная информация	Входная информация
Заключение договоров	0803	Заключение и оформление договоров на снабжение	Документальное оформление договоров с поставщиками	Договор на снабжение	Справочник поставщиков, банков
Заключение договоров	0801 0804	Прием заказов и заключение и оформление договоров по продаже	Документальное оформление договоров с покупателями	Договор на продажу	Справочник покупателей, банков
	0802	Исследование рынка	Статистическая оценка деления рынка	Отчет об исследовании рынка	Прайслист конкурентов
	0805	Оформление заказов розничными покупателями через Internet	Оформление и запись в БД заказов через Internet	Реестр заказов покупателей, ВК, электронные поручения для клиентов, прайс-лист	Справочник покупателей, банков, электронных платёжных систем
Планирование	0806	Планирование продаж	Формирование плана купли и продажи	Планы купли и продажи	Договоры с покупателями и поставщиками, оценка емкости рынка, оценка деления рынка среди конкурентов, реестр заказов, окончательная ведомость
Учет продаж	0808	Учет продаж	Учет продаж	Реестр договоров с клиентами	Договоры с покупателями, реестр заказов
Анализ продаж	0809	Анализ продаж	Анализ продаж определенного билета определенным покупателем	Оценка спроса на определенный билет	Договоры с покупателями

Схема взаимосвязей функций компании может быть представлена следующим образом:

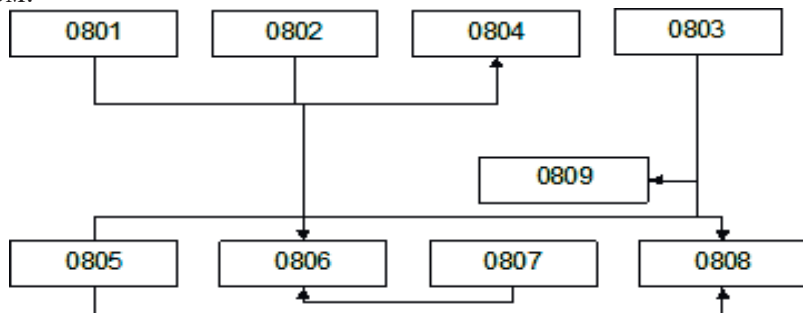


Рисунок 4 - Схема взаимосвязей функций компании

Программная и техническая архитектура ИС на предприятии состоит из двух подсистем, первая подсистема предназначена для автоматизации бизнес-процессов продаж, вторая - для автоматизации работы с финансовой составляющей процесса продаж:

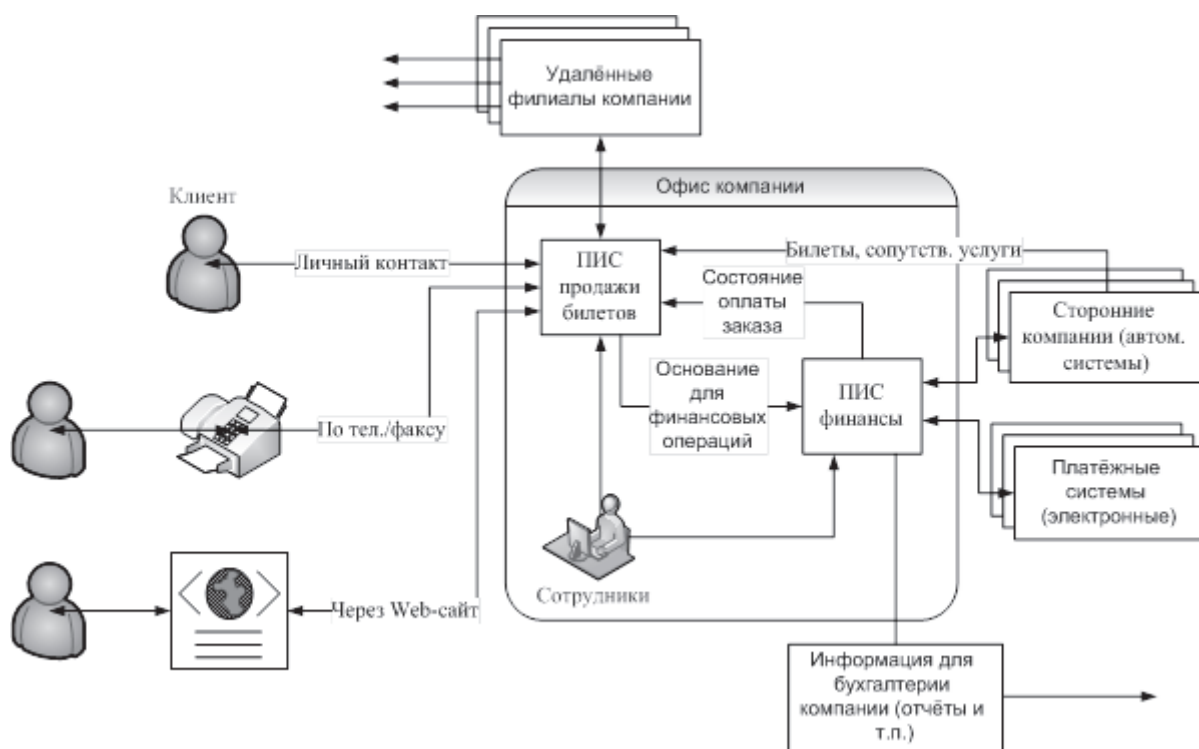


Рисунок 5 - Архитектура информационной системы компания ООО «5 Звезд» (концептуальное представления)

Работа удаленных офисов реализуется двумя способами: в режиме удаленного терминала; через web-интерфейс. Заказ через Интернет возможен в двух режимах: через собственный веб-сайт компании; путем интеграции данной системы с web-сайтами сторонних компаний.

Анализ рынка программных продуктов показывает, что данная автоматизированная система может быть реализована на платформе MS Windows, реализация Back-Office осуществляется в среде системы 1С-Предприятие.

Анализируя рынок программных продуктов, наше внимание привлекло ряд автоматизированных систем, которые подойдут нам, например:

1. Билетная система «Базис»

Вся информация, циркулирующая в билетном хозяйстве, структурируется, и заносится в общую базу данных. Программа, при необходимости, сама может отследить продажи билетов и предоставить отчет в электронной форме.

Проектное решение:

Решение задачи требует разработки сетевой инфраструктуры, потому что именно каналы сети могут соединять отдаленные узлы. Под топологией вычислительной сети понимают конфигурацию графа, вершинам которого отвечают компьютеры сети, а ребрам - физические связи между ними. Следует отметить, что конфигурация физических связей определяется электрическими соединениями компьютеров между собой и может отличаться от конфигурации логических связей между узлами сети. Логические связи являют собой маршруты передачи данных между узлами сети и создаются путем соответствующего налаживания коммуникационного оборудования.

Выбор топологии электрических связей существенно влияет на многие характеристики сети. Например, наличие резервных связей повышает надежность сети и дает возможность балансирования загрузки отдельных каналов. Простота присоединения новых узлов, что свойственно для некоторых топологий, делает сеть легко расширяемой. Экономические рассуждения часто приводят к выбору топологии, для которой характерным минимальная суммарная длина линий связи.

Сетевая инфраструктура организована с использованием архитектуры Fast Ethernet с физической организацией сети в виде звезды.

Основные преимущества:

Вся информация, циркулирующая в билетном хозяйстве, структурируется, и заносится в общую базу данных. Программа, при необходимости, сама может отследить продажи билетов и предоставить отчет в электронной форме.

Проектное решение:

Решение задачи требует разработки сетевой инфраструктуры, потому что именно каналы сети могут соединять отдаленные узлы. Под топологией вычислительной сети понимают конфигурацию графа, вершинам которого отвечают компьютеры сети, а ребрам - физические связи между ними. Следует отметить, что конфигурация физических связей определяется электрическими соединениями компьютеров между собой и может отличаться от конфигурации логических связей между узлами сети. Логические связи являются маршрутами передачи данных между узлами сети и создаются путем соответствующего налаживания коммуникационного оборудования.

Выбор топологии электрических связей существенно влияет на многие характеристики сети. Например, наличие резервных связей повышает надежность сети и дает возможность балансирования загрузки отдельных каналов. Простота присоединения новых узлов, что свойственно для некоторых топологий, делает сеть легко расширяемой. Экономические рассуждения часто приводят к выбору топологии, для которой является характерным минимальная суммарная длина линий связи.

Сетевая инфраструктура организована с использованием архитектуры Fast Ethernet с физической организацией сети в виде звезды.

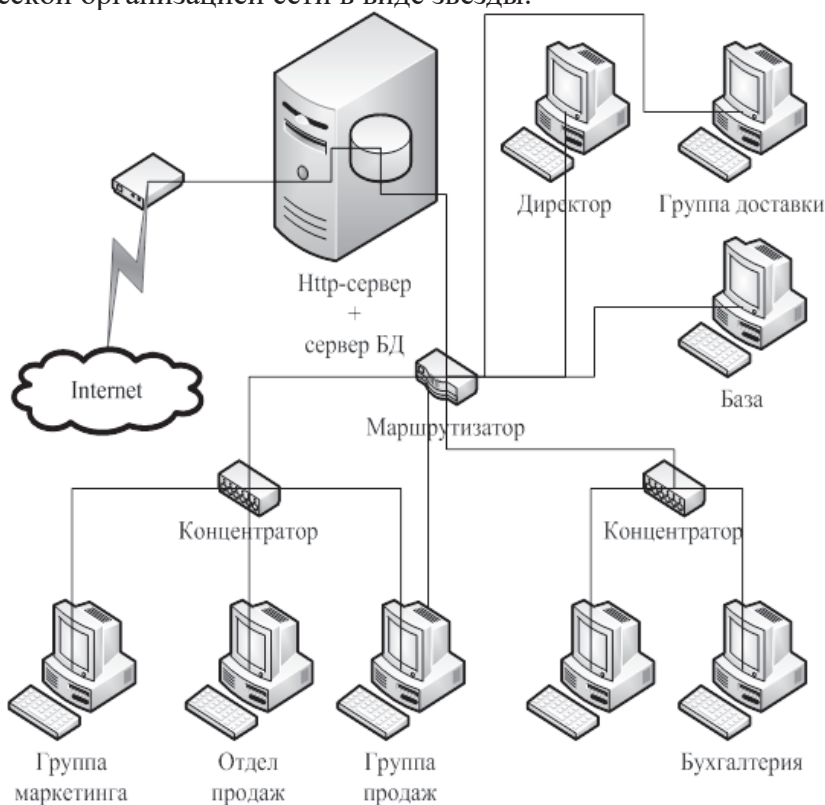


Рисунок 6 - Структура технического обеспечения ООО «5 Звезд»

На рисунке показано, что сеть имеет древовидную структуру. Ее центром является маршрутизатор, который соединяет сети всех подразделов организации в единственную вычислительную сеть. Также в ТО сети присутствующие концентраторы для соединения отдельных узлов сети. Они служат также для использования маршрутизатора, позволяющего локализовать трафик подразделов. Кроме того, такая организация сети повышает ее надежность, так как только при условии выхода из строя главного маршрутизатора возможна остановка функционирования сети.

Сети этого типа имеют физическую организацию в виде звезды, и логическую - общей шины. Как видно из рис 6, сердцем информационного обеспечения организации является распределенный сервер, который исполняет роль, как сервера web-дополнений, так и сервера распределенной БД.

Для хранения и выборки данных используется СУБД Interbase компании Borland Software Corporation. Она зарекомендовала себя как легкая СУБД с достаточно высокими скоростными показателями и малой потребностью системных ресурсов. Кроме того, по сравнению со стандартной для решения задач данного типа СУБД MySQL, СУБД Interbase имеет достаточные функциональные возможности для последующей интеграции в подсистемы торговой организации. Это, прежде всего, объясняется поддержкой триггеров, процедур, которые сохраняются на сервере, и представлений. Также используется бесплатный HTTP-сервер Apache, который зарекомендовал себя как безопасный, надежный, быстрый сервер с возможностью подключения модулей расширения.

Для разметки Web-страниц использовался язык гипертекстовой разметки HTML (HyperText Markup Language). Сам язык реализован в виде дескрипторов маркеров, которые описывают размещения элементов страницы, а также дополнительные характеристики каждого элемента.

1. ПОРЯДОК ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ

Лабораторная работа №4 выполняется командой, разрабатывается единое приложение, члены команды самостоятельно распределяют работы, результаты выполнения лабораторной работы представляются преподавателю командой, каждый член команды должен быть готов ответить на вопросы и сообщить о индивидуальном вкладе в командный проект.

ЧАСТЬ 1.

Для выполнения Части 1 лабораторной работы №4 необходимо разработать:

- Функциональную архитектуру приложения.
- Структуру диалога приложения.
- Дерево программных модулей.
- Алгоритм функционирования системы.

Разработку программной архитектуры информационной системы необходимо выполнить в строгой синхронизации с моделями IDEF0, DFD, ERA, разработанными на 1 и 2 лабораторных работах по дисциплине.

ЧАСТЬ 2.

Для выполнения Части 2 лабораторной работы №4 необходимо:

- Выполнить анализ системных компонентов, требующихся для поддержки архитектуры приложений и информационной архитектуры информационной системы: компьютеры и серверы; операционные системы; сетевые компоненты; принтеры; модемы и т.д. и т.п.
- Разработать структуру технического обеспечения информационной системы.

Разработку технологической архитектуры информационной системы необходимо выполнить в строгой синхронизации с моделями IDEF0, DFD, ERA, разработанными на 1 и 2 лабораторных работах по дисциплине.

ЧАСТЬ 3.

- Сформировать отчет в формате Microsoft Word, состоящий из скриншотов моделей по разделам с титульной страницей и названием файла в формате «Отчет_Фамилия_Л4.docx»:
- Ответом на задание прикрепить в Электронную образовательную среду Станкина файл: Отчет_Фамилия_Л4.doc