Титульный лист- стр. 1

Задание на КП – стр.2

Аннотация – стр. 3

СОДЕРЖАНИЕ

[Введение 5](#_Toc21697717)

[1 Техническое задание 6](#_Toc21697718)

[1.1 Введение 6](#_Toc21697719)

[1.2 Назначение разработки 6](#_Toc21697720)

[1.3 Требования к функциональным характеристикам 6](#_Toc21697721)

[2 Практическая часть 9](#_Toc21697722)

[2.1 Функциональная модель 9](#_Toc21697723)

[2.2 Диаграмма потоков данных 10](#_Toc21697724)

[2.3 Проектирование базы данных 11](#_Toc21697725)

[3 Программная реализация 13](#_Toc21697726)

[3.1 Инструментальные средства реализации программного приложения 13](#_Toc21697727)

[3.2 Описание модуля данных 13](#_Toc21697728)

[3.3 Структура программного приложения 14](#_Toc21697729)

[3.4 Справочная информация 14](#_Toc21697730)

[3.5 Основные функции обработки данных (переименовать) 15](#_Toc21697731)

[3.6 Формы отчётной информации 15](#_Toc21697732)

[Заключение 16](#_Toc21697733)

[Список использованных источников 17](#_Toc21697734)

[Приложение А 18](#_Toc21697735)

# Введение

Введение (2/3 - 1 страница) содержит краткий обзор вопросов, изложенных в курсовом проекте, краткое описание предметной области и поставленной задачи разработки программного приложения, используемую СУБД, программное средство реализации приложения.

Кратко охарактеризуйте главы курсового проекта.

# Техническое задание

## Введение

Данная автоматизированная система является системой учета всех клиентов предприятия и предназначена для обеспечения публичного доступа клиентов к открытой части информации.

Также автоматизированная система является системой учета предоставляемых услуг и расценок.

Областью применения программного продукта является сфера деятельности спортивного клуба.

## Основание разработки

## Назначение разработки

Автоматизированная информационная система «Спортивный клуб» предназначена для учета, обобщения, хранения, предоставления информации и услуг, предлагаемых спортивным клубом.

Пользователями программы выступают менеджеры, инструкторы, тренеры, посетители спортивного клуба.

Данные о посетителях отображаются в таблице клиентов.

Услуги реализуются за наличный и безналичный расчет. Информация о покупке услуг записывается в таблицу оплата.

Информация о предоставляемых услугах содержится в системе учета услуг: название занятия, цена за 1 занятие.

В систему учета посетителей клуба записывается: ФИО, паспортные данные, пол, телефонный номер.

В систему учета оплаты записывается: способ расчета, дата начала регистрации, видов занятий, количества посещений.

Спортивный клуб предоставляет следующие услуги: групповые занятия, частные занятия. Подробнее о предоставляемых услугах написано в **справке**.

## Требования к функциональным характеристикам

* + 1. Требования к функциональным характеристикам.

Программный продукт должен выполнять следующие функции:

* Хранение и обработка списка предоставляемых услуг и расценок.
* Регистрация новых клиентов и редактирование информации о клиентах. Хранение следующей информации о клиентах: ФИО, паспортные данные, телефонный номер.
* Возможность записи клиента с любого числа месяца на заданное количество занятий заданного вида.
* Оплата производится немедленно, выдается квитанция.
* Возможность пропуска занятия по уважительной причине путем продления абонемента.
* Оплата производится немедленно, выдается квитанция.
* Возможность пропуска занятия по уважительной причине путем продления абонемента.
* Поиск по разным полям в любой форме.
* Сортировка данных в табличном виде.
  + 1. Требования к надежности.

Программный продукт должен соответствовать современному уровню требований к надежности программного обеспечения:

* Предусматривать контроль вводимой информации и блокировку некорректных действий пользователя при работе с системой
* Обеспечивать корректное завершение вычислений с соответствующей диагностикой при превышении имеющихся вычислительных ресурсов
* Обеспечивать целостность информации, хранящейся в базе данных
  + 1. Требования к составу и параметрам технических средств.

Системные требования для работы программного продукта должны быть следующими: тактовая частота процессора – 1000 Гц; объем оперативной памяти 64 Мб; объем свободного дискового пространства 20 Мб; разрешение монитора 1024х768; наличие устройства чтения компакт-дисков; принтер.

* + 1. Требования к информационной и программной совместимости.

Программа должна работать в операционных системах Windows XP/7/8/10.

* 1. **Требования к программной документации**

Разрабатываемая система должна включать справочную информацию о работе системы и подсказки пользователю. В состав сопровождающей документации должны входить: расчетно-пояснительная записка, содержащая описание системы; руководство пользователя; руководство системного программиста.

* 1. **Этапы разработки**

В таблице 1.1 представлены этапы разработки информационной подсистемы «Учёта клиентов и оплаты занятий»

Таблица 1.1 – Этапы разработки информационной подсистемы «Учёта клиентов и оплаты занятий»

|  |  |  |  |
| --- | --- | --- | --- |
| Номер этапа | Название этапа | Срок | Отчетность |
| 1 | Начало разработки | 10.09.2019 | Получение темы курсовой работы |
| 2 | Изучение предметной области | 10.09.2019-15.09.2019 | Структурированный план разработки системы |
| 3 | Написание технического задания | 15.09.2019-  18.09.2019 | Техническое задание |
| 4 | Составление функциональной модели IDEF0 | 18.09.2019-  22.09.2019 | Диаграмма формата IDEF0 в программе Ramus |
| 5 | Составление модели потоков данных DFD | 22.09.2019-  27.09.2019 | Диаграмма формата DFD в программе Ramus |
| 6 | Составление ERD – диаграммы | 27.09.2019-  02.10.2019 | Диаграмма формата ERD в программе Visio |
| 7 | Создание базы данных и наполнение | 02.10.2019-  14.10.2019 | Схема база данных в MS Access |
| 8 | Программная реализация | 15.10.2019-  01.12.2019 | Программная система |
| 9 | Тестирование системы | 02.12.2019-  9.12.2019 | Результаты тестирования |
| 10 | Оформление пояснительной записки | 10.12.2019-  22.12.2019 | Пояснительная записка к курсовому проекту |
| 11 | Защита курсового проекта | 18.12.2019 | Оценка |

# Практическая часть

## Функциональная модель

Спортивный клуб действую на основе закона о защите прав потребителей, учитывает клиентов и оплату занятий. В спортивном клубе имеется прейскурант цен на оказываемые услуги. Менеджер может редактировать и вводить информацию. Входом, что инициирует осуществление деятельности спортивного клуба служит прейскурант, динамика расценок в хронологии, база данных клиентов клуба, деньги, выходом – каков результат этого действия – квитанция об оплате, отчетность. Стрелка управления называются «закон РФ о защите прав потребителей» Механизмами являются «менеджер» и «администратор».

На рисунке 2.1 представлена контекстная диаграмма IDEF0

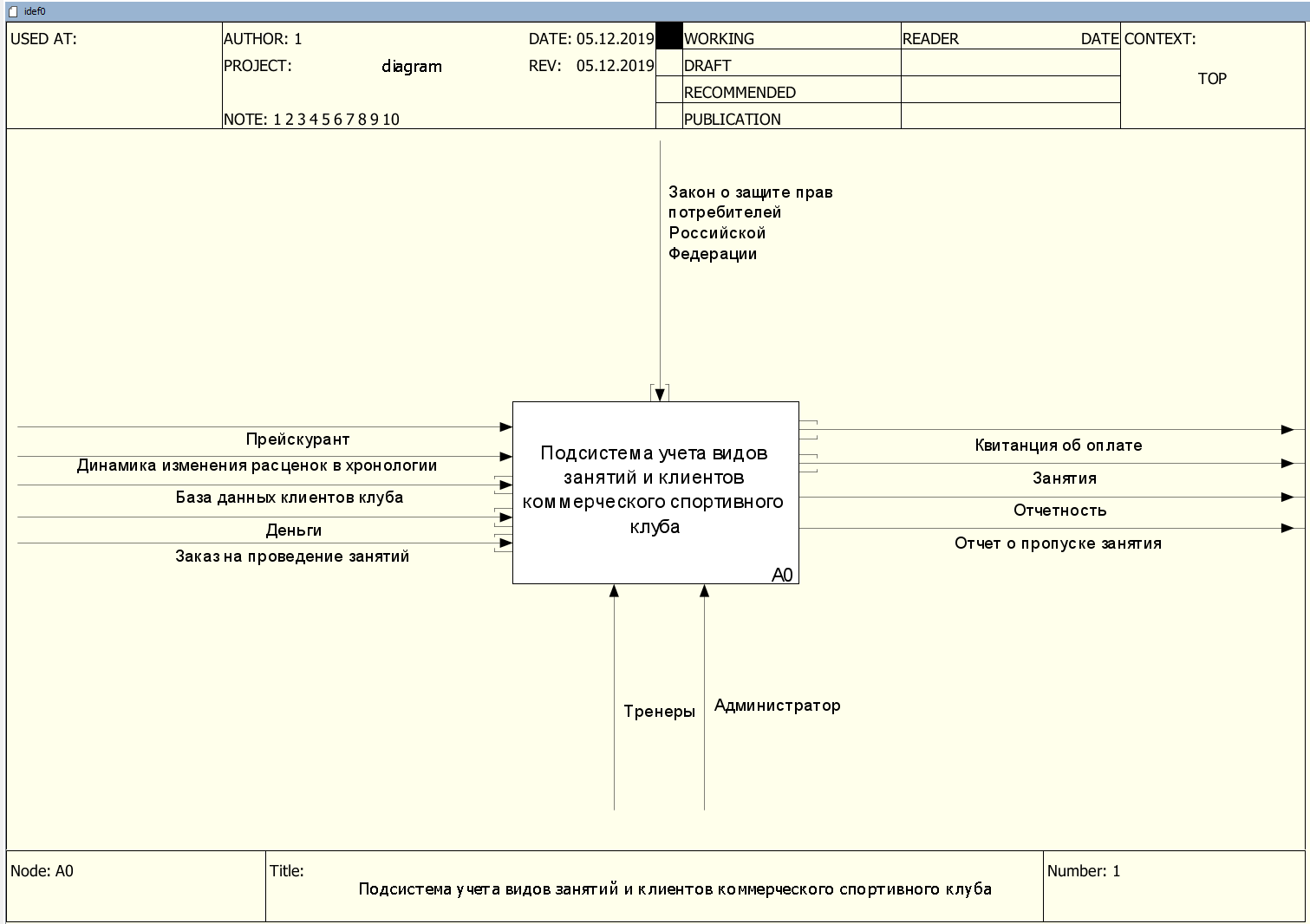


Рисунок 2.1 – Контекстная диаграмма формата IDEF0

Декомпозиция

Основные функции диаграммы IDEF0 1 уровня:

1. Предоставление услуг
2. Ведение системы учета видов занятий и расценок
3. Ведение системы учета клиентов
4. Формирование отчётов

На рисунке 2.2 представлена диаграмма первого уровня формата IFEF0

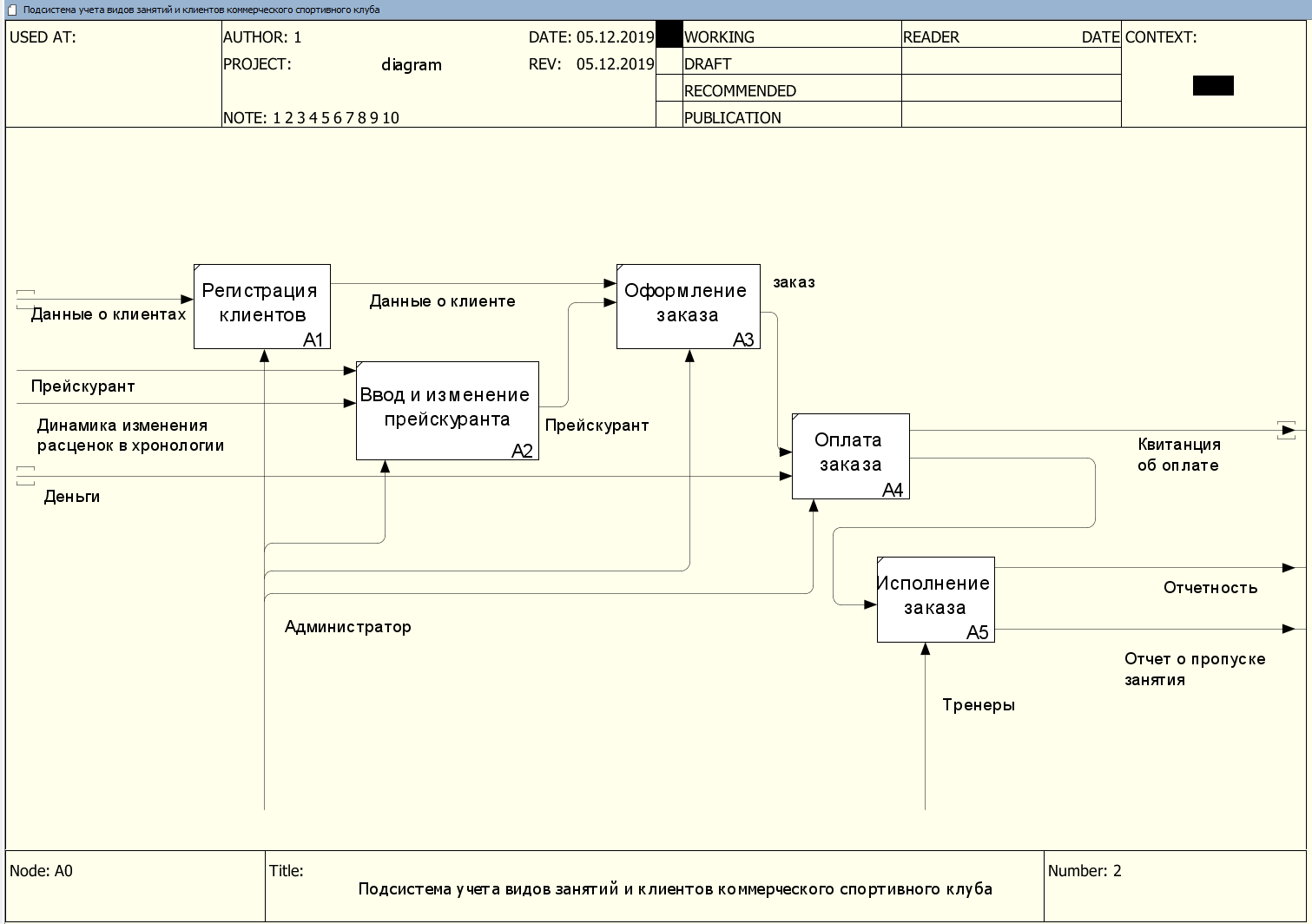


Рисунок 2.2 – Диаграмма первого уровня формата IDEF0

## Диаграмма потоков данных

При построении функциональной модели системы альтернативой методологии IDEF0 является методология диаграмм потоков данных (Data Flow Diagrams, DFD). В отличие от IDEF0, предназначенной для проектирования систем вообще, DFD предназначена для проектирования информационных систем. Ориентированность этой методологии на проектирование автоматизированных систем делает ее удобным и более выгодным инструментом при построении функциональной модели TO-BE.

Выявим внешние сущности подсистемы:

1. Правительство РФ
2. Подсистема учёта клиентов и оплаты занятий
3. Руководство

Контекстная диаграмма DFD для учёта клиентов и оплаты занятий показана на рисунке 2.3.

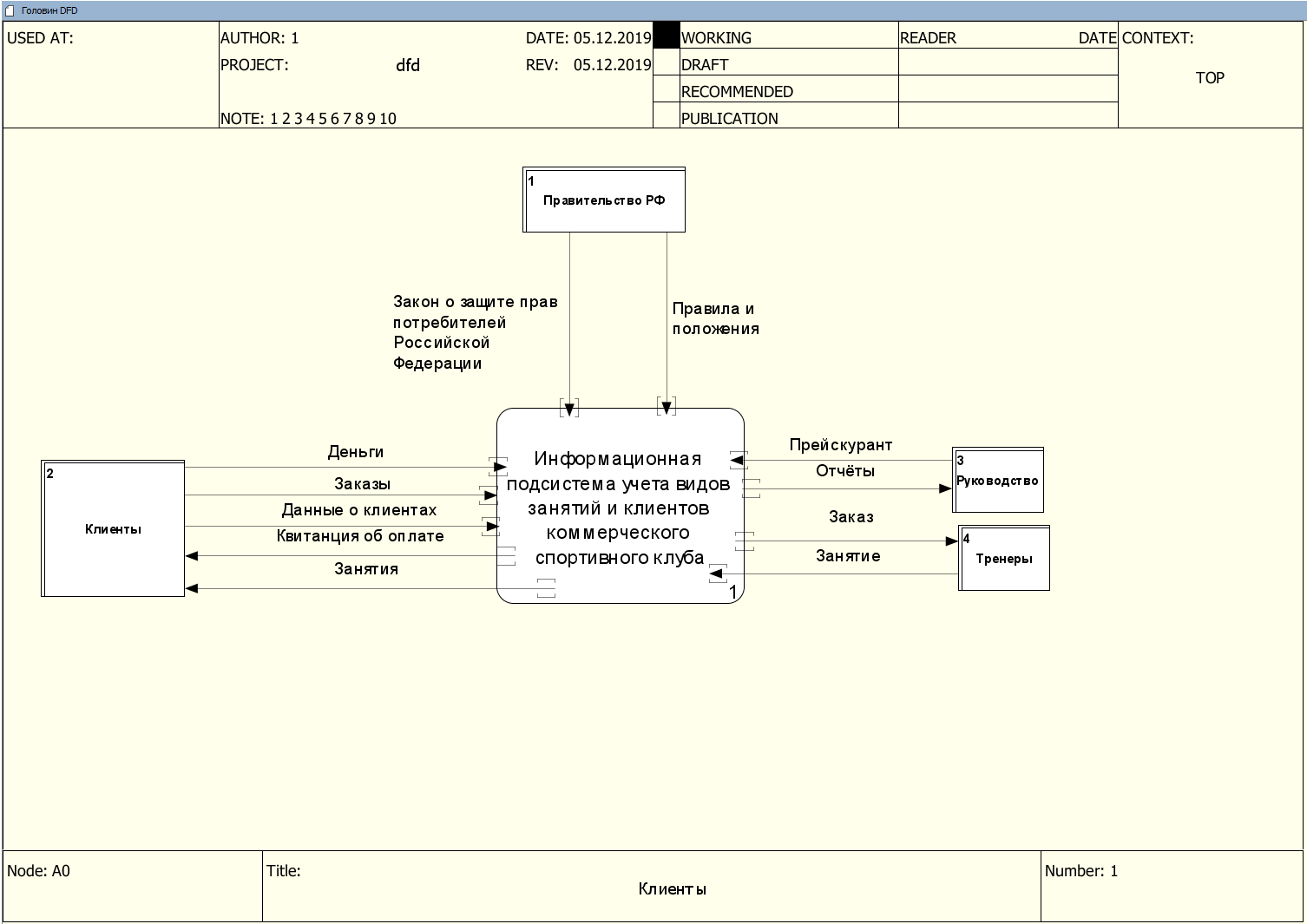
****

Рисунок 2.3 – Контекстная диаграмма DFD

Декомпозируем исходную систему на следующие бизнес-процессы:

1. Учёт переподготовок
2. Учёт проведенных занятий
3. Ввод и редактирование данных о квалификациях
4. Доступ к информации о тренерах
5. Формирование отчётов о необходимости переподготовки
6. Формирование отчётов о необходимости повышения квалификации

Диаграмма DFD первого уровня показана на рисунке 2.4.

Диаграмма DFD первого уровня показана на рисунке 2.4.

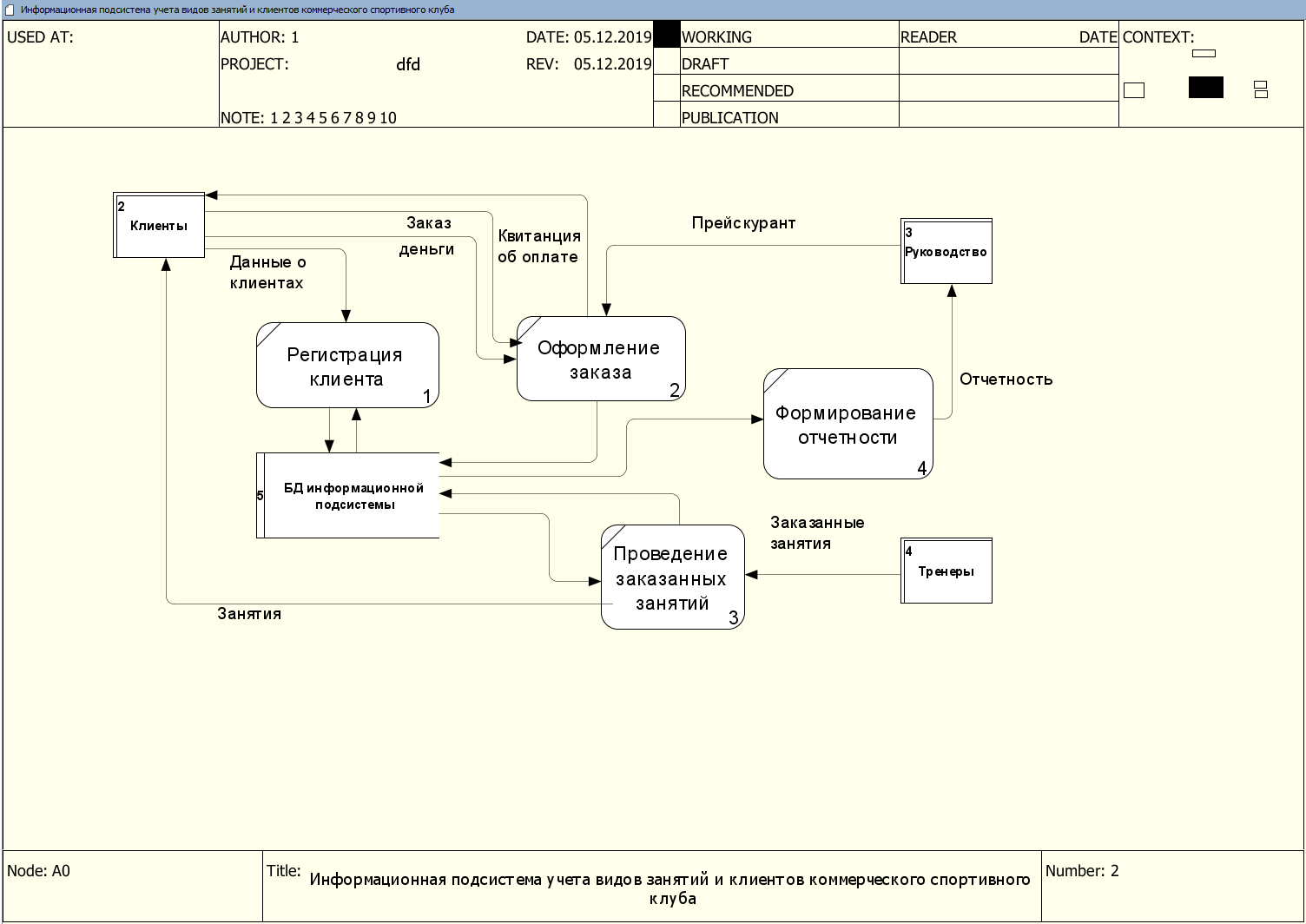


Рисунок 2.4 –Диаграмма DFD первого уровня

## Проектирование базы данных

Проектная часть работы содержит построение моделей: концептуальной, логической и физической.

Концептуальная модель данных представляется в виде диаграммы «сущность - связь» (ER- диаграмма), которая отражает сущности и связи между ними. Подробно описать сущности и связи между ними. Перечислить сначала независимые (стержневые) сущности, затем для каждой зависимой (дочерней) сущности пояснить смысл внешнего ключа связи, указать тип связи: один-к-одному, один- ко-многим.

Пример модели, построенной с помощью MS Visio, приведен на рисунке 2.5.

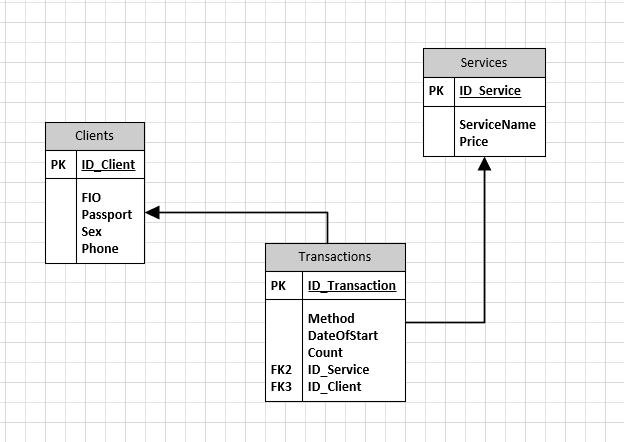
****

Рисунок 2.5 – ER- диаграмма базы данных

Логическая модель отражает реляционную схему базы данных в терминах СУБД MS Access. Выявить независимые и зависимые таблицы реляционной модели данных. Определить связи между ними, внешние ключи и их ограничения целостности.

Привести схему БД, построенной средствами СУБД MSAccess, описать таблицы и связи. Схема базы данных приведена на рисунке 2.6.

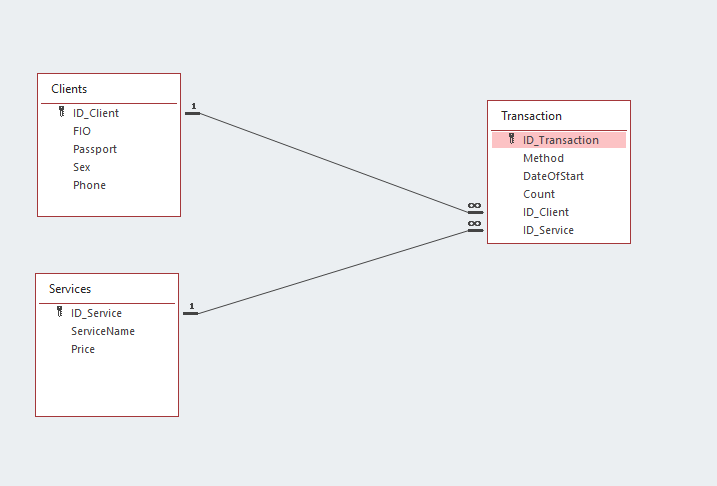


Рисунок 2.6 –Схема базы данных

# Программная реализация

## Инструментальные средства реализации программного приложения

Программные продукты, которые использовались для реализации поставленной задачи:

Microsoft Access – это реляционная система управления базами данных (Система управления базами данных), входящая в пакет MS Office.

Delphi – встроенная среда разработки ПО для Microsoft Windows, Mac OS, iOS и Android.

ADO – интерфейс программирования приложений для доступа к данным, разработанный компанией Microsoft (MS Access, MS SQL Server) и основанный на технологии компонентов ActiveX. Представляет данные из разнообразных источников (реляционных баз данных, текстовых файлов) в объектно-ориентированном виде.

Для хранения данных использовалась СУБД Access.

Среди средств, которые предлагает СУБД Access можно отметить возможность динамического обмена данными (DDE) между Access и другими приложениями, которые поддерживают эту технологию. Также имеется возможность применения технологии ActiveX, позволяющей использовать разработчику в своем программном продукте не только, те объекты, которые свойственны Access, но и объекты других приложений, например Excel или Word.

Весьма удобной возможностью является то, что пользователь при обработке данных может работать не только с БД обрабатываемого в Access формата, но и экспортировать данные других СУБД, имеющие совершенно другой формат представления, в частности, формат FoxPro, Paradox так далее. Кроме этого, при помощи Access пользователь может обрабатывать БД, поддерживающие открытый доступ к данным (стандарт ODBC), в частности, весьма популярных в последнее время серверов баз данных Oracle и SQL Server. При обработке данных в Access используется структурированный язык запросов SQL, который без преувеличения можно назвать стандартным языком БД. С его помощью можно выполнять самую разнообразную обработку имеющихся данных, в частности, создавать выборки требуемой структуры, вносить необходимые изменения в имеющиеся БД, преобразовывать или удалять таблицы, формировать данные для отчетов и многое другое.

Важным преимуществом СУБД Access является то, что с ее помощью можно разрабатывать системы, которые обрабатывают БД как на отдельном компьютере, так и в локальной сети предприятия или в Internet, используя режим обработки данных "клиент-сервер".

Необходимо также отметить, что Access предоставляет широкие возможности по созданию приложений, связанных с обработкой БД. При этом разработчику не обязательно быть программистом высокого класса, а вполне достаточно иметь представление о создании событийных приложений в среде Windows, а также владеть некоторыми навыками программирования на языке Visual Basic. В этом случае разработчик достаточно быстро сможет овладеть навыками по созданию приложений в Access, что позволит выполнять автоматизирование как простых, так и достаточно сложных задач, связанных с обработкой данных.

Программа создавалась в среде программирования Delphi

В Delphi используется довольно простой язык программирования Object – Pascal. Данный выбор был обусловлен следующими характеристиками:

* Cочетание возможностей быстрого прототипирования приложений с технологиями уровня предприятия обеспечивает плавное и предсказуемое развитие проектов любого масштаба.
* Delphi включает средства поддержки Web и интеграции с существующими Windows-приложениями. Возможность работы с разнообразными источниками информации и прозрачность подключения новых механизмов доступа к данным делает Delphi наиболее оптимальным выбором для профессионалов индустрии информационных технологий.
* Delphi это среда компилирующего типа, то есть в результате компиляции приложения создается выполнимый файл, который можно легко распространять на другие компьютеры, даже такие, где не установлена система Delphi.
* Delphi предоставляет готовые компоненты для разработки приложений, например, такие как кнопки, списки, поля редактирования и так далее.
* свойства (properties) и перегружаемость процедур и функций (overloading).

Модель программирования в Delphi - компонентная, что позволяет пользоваться компонентами, написанными другими разработчиками, даже не имея их исходного кода и уж подавно не изучая его. Применение компонентной модели приводит к тому, что довольно многое в поведении объектов программировать не нужно вообще

Система программирования Delphi рассчитана на программирование различных приложений и предоставляет большое количество компонентов для этого. К тому же работодателей интересует, прежде всего, скорость и качество создания программ, а эти характеристики может обеспечить только среда визуального проектирования, способная взять на себя значительные объемы рутинной работы по подготовке приложений, а также согласовать деятельность группы постановщиков, кодировщиков, тестеров и технических писателей. Возможности Delphi полностью отвечают подобным требованиям и подходят для создания систем любой сложности.

Для связи данных между программой и СУБД Access использовалась технология ADO

ADO обеспечивает единый доступ к различным источникам данных, благодаря этому появляется возможность создания приложения клиента по управлению распределенными и разноформатными базами данных (Access, FoxPro, InterBase, ORACLE, SQL Server и т. д.). Запрос по управлению данными формируется по одинаковым правилам и направляется к любому источнику данных. Серверы баз данных напрямую выполняют полученные запросы.

В этой технологии также предусмотрен доступ к файловым последовательностям, электронным таблицам, файлам электронной почты и т. д. через специализированные механизмы ADO и интерфейсы встраивания OLE DB. OLE DB представляет собой коллекцию специализированных объектов СОМ. Каждый такой объект состоит из: интерфейсов по обработке данных, специализированных интерфейсов по доступу к данным из конкретного источника и интерфейсов для взаимосвязи объектов между собой. Все специализированные объекты создаются и функционируют по соглашениям технологии СОМ.

Важную роль играют компоненты-провайдеры, которые координируют работу объектов ADO и обеспечивают их взаимодействие с набором данных в приложении клиента. В технологии ADO понятие «данные» включает в себя, кроме общепринятых наборов данных, и графические данные для геоинформационных систем, древовидные структуры реестров и т. д.