Частное учреждение образования   
«Колледж бизнеса и права»

ПРОГРАММА ДЛЯ АВТОМАТИЗАЦИИ УЧЁТА КЛИЕНТАМИ ВЫПОЛНЕНЫХ ЗАКАЗОВ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

«Базы данных и системы управления базами данных»

КП Т.593001.401

Учащийся (Н.С. Веринский)

Руководитель проекта (С.В. Банцевич )

2018

Содержание

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

КП Т.593001.401 ПЗ

Разраб.

Веринский Н.С.

Провер.

Банцевич С.В.

Реценз.

Н. контр.

Утверд.

*Программа для автоматизации учёта оплаты клиентами выполненных заказов*

Лит.

Листов

КБП

0

у

# Введение

Учет заказов на логистических центрах является важнейшей задачей, решение которой позволяет снизить затраты на производстве и оптимизировать затрат организации.

Дабы избежать мошенничества со стороны исполнителя, и клиента, применяются системы учета клиентами выполненных заказов. Благодаря подобным системам можно составлять договор между исполнителем и нанимателем. Наличие платежного календаря в подобных системах позволит снизить риск просрочки платежа, следовательно, и понизить издержки, такие как – пеня, что позволит нанимателю экономить свои деньги, а исполнителю получать оплату вовремя.

Целью курсового проекта автоматизации учёта оплаты клиентами выполненных заказов является разработка программного средства «TransOrder», которое будет осуществлять ведение базы данных, содержать информацию о заказчиках, исполнителях, транспорте, используемом при транспортировке, ведение календаря оплат и автоматическое составление договора.

Пояснительная записка состоит из шести разделов, содержащих необходимую информацию по организации эксплуатации программного средства.

В первом разделе «Объектно-ориентированный анализ и проектирование системы» раскрывается организационная сущность задачи, описывается предметная область и круг задач, которые должны быть автоматизированы. Описывается задача, перечисляются основные функции программы. Строится информационная модель, отражающая сущности задачи, их свойства и взаимосвязи.

Во втором разделе «Вычислительная система» перечисляются требования к аппаратному и программному обеспечению компьютера, проводится характеристика операционной системы, обоснование выбранной среды для разработки программы.

В третьем разделе «Проектирование задачи» проводится объектно-ориентированный анализ задачи, приводится требования к приложению, концептуальный прототип, организация данных, функции и элементы управления, проектирование справочной системы приложения.

В четвертом разделе «Описание программного средства» представлены общие сведения о программном средстве и его функциональном назначении, входные и выходные данные.

В пятом разделе «Методика испытаний» описываются требования к аппаратному и программному обеспечению компьютера для проведения испытаний, требования к характеристикам программы применительно к условиям эксплуатации, требования к информационной и программной совместимости. Представляются результаты функционального тестирования.

Шестой раздел «Применение» предназначен для описания сведений о назначении программного средства и области его применения. В этом разделе приводится структура справочной системы, а также методика ее использования.

В заключении анализируется созданное программное средство, определяется степень соответствия поставленной задачи и выполненной работы.

Приложение А содержит текст программных модулей.

Приложение Б содержит выходные данные.

В графической части представлены диаграммы вариантов использования, классов, деятельности, последовательности и компонентов.

# Объектно-ориентированный анализ и проектирование системы

## Сущность задачи

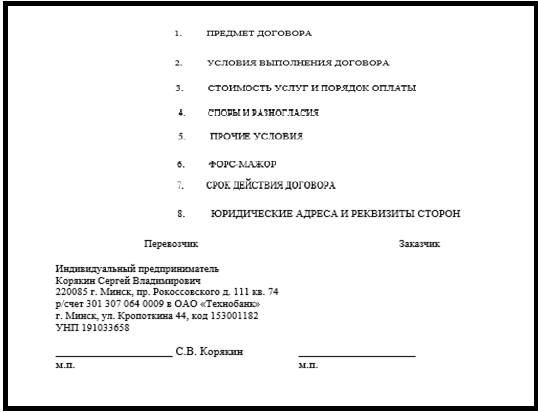
Предметной областью решаемой задачи является торгово-транспортная организация, предоставляющая услуги транспортировки и доставки грузов и заказов, а также хранящая информацию о тягачах, сцепках, полуприцепах и их видах, водителях, оплатах, заказах, пунктах разгрузки/погрузки, договорах и фирмах-заказчиках, а также ведет журнал договоров и карточек договоров.

Расчет оплаты и пени в случае не своевременной выплаты рассчитывается автоматически.

Основой любой операции является документ, выступающий: сводом условий выполнения заказа, гарантом выполнения обязанностей со стороны организации и клиента, а также сводом инструкций в случае нарушения договорных условий.

Для того чтобы заказ мог быть выполнен и оформлен необходимо наличие договора на оказание автотранспортных услуг.

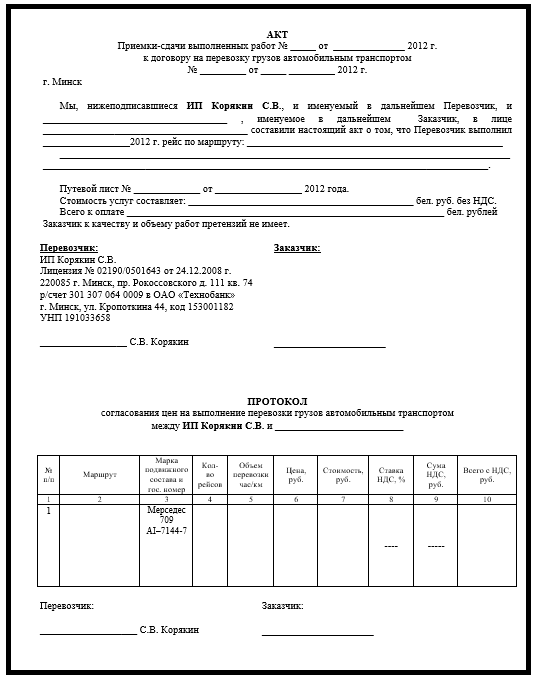
Шаблон договора на оказания автотранспортных услуг представлен рисунке 1.1

****

**Рисунок 1.1**

**Для снятия всех форм ответственности с исполнителя и заказчика необходимо создать акт выполненных работ.**

**Шаблон акта выполненных работ приведен на рисунке 1.2**

****

**Рисунок 1.2**

Исходя из анализа предметной области можно выделить следующие задачи, подлежащие автоматизации:

* ведение базы данных (БД) содержащей информацию о тягачах, сцепках, полуприцепах и их видах, оплатах, заказах, пунктах разгрузки/погрузки, договорах и фирмах-заказчиках;
* автоматизировать составление плана-графика платежей по договору;
* реализовать функцию контроля как всей оплаты по договору, так и поэтапной оплаты в соответствии с планом-графиком оплаты;
* предусмотреть возможность просмотра и поиска необходимой пользователю информации.
* обеспечить возможность просмотра и поиска необходимой пользователю информации; предоставить возможность сортировки данных по различным критериям;
* создать отчёты по интересующей пользователя информации (договор), в том числе и статистической с возможностью выбора аналитики и оцениваемых параметров;
* организовать возможность экспорта в файлы Microsoft Office интересующей пользователя информации.

Существует несколько аналогов разрабатываемой программы, которые автоматизируют процесс учета расчета клиентами выполненных заказов. Например, программа компании «БелДорТяжТранс»: «CargoCRM». Разрабатываемая программа является упрощённой версией этой программы.

## Проектирование модели

Цель моделирования данных состоит в обеспечении разработчика информационной системы концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Наиболее распространенным средством моделирования данных являются диаграммы «Сущность-связь» (ERD). С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). ERD непосредственно используются для проектирования реляционных баз данных. Нотация ERD была впервые введена П. Ченном и получила дальнейшее развитие в работах Баркера.

Диаграмма «Сущность-связь» представлена на рисунке 1.3.

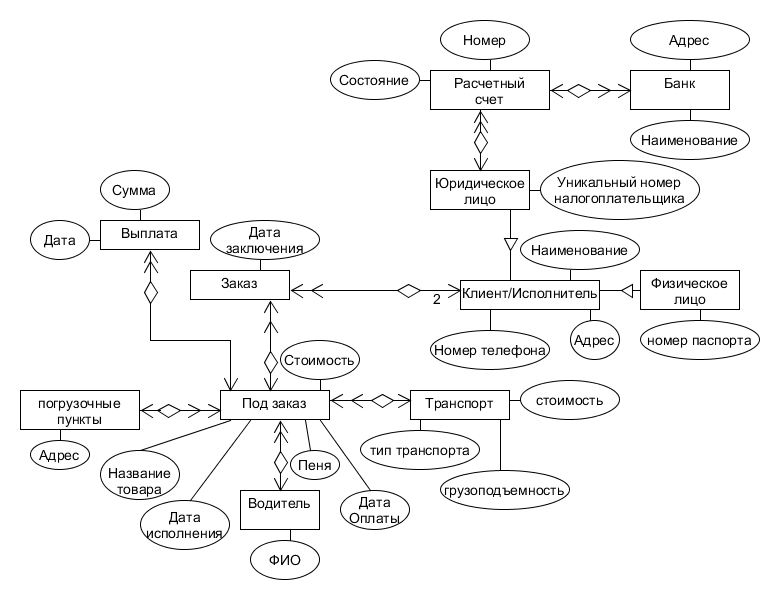


Рисунок 1.3

Исходя из исследования предметной области, можно выделить следующие сущности разработки: «Заказ», «Выплата», «Транспорт», «Клиент/Исполнитель», «Транспорт», «Погрузочные пункты», «Водитель», «Юридическое лицо», «Расчетный счет», «Банк».

Для сущности «Выплата» можно выделить следующие атрибуты:

– дата;

– сумма.

Для сущности «Погрузочные пункты» атрибутом будет являться «Адрес».

Для сущности «Заказ» атрибутом является:

– дата заключения.

Для сущности «Под заказ» выделены следующие атрибуты:

– стоимость;

– пеня;

– дата исполнения;

– название товара.

Для под сущности «Клиент/исполнителя» - «Юридическое лицо» атрибутом является:

– уникальный номер налогоплательщика.

Для сущности «Расчетный счет» атрибутами являются:

– состояние;

– номер расчетного счета.

Для сущности «Банк» атрибутами является:

– адрес;

– наименование.

Для под сущности «Клиент/исполнителя» - «Физическое лицо» атрибутами является:

– номер паспорта;

Для сущности «Погрузочные пункты» атрибутом является:

– адрес.

Для сущности «Транспорт» атрибутами является:

– грузоподъемность;

– стоимость;

– тип транспорта.

Для сущности «Клиент/Исполнитель» атрибутом является:

– номер телефона;

– адрес.

Для сущности «Водитель» атрибутом является:

– фамилия имя отчество (ФИО).

Перечень этих диаграмм представляет собой неотъемлемую часть графической нотации языка Unified Modeling Language (UML), сам процесс объектно-ориентированного программирования (ООП) неразрывно связан с процессом построения этих диаграмм. Совокупность построенных таким образом диаграмм содержит всю информацию, необходимую для реализации проекта сложной системы.

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций – диаграмм. В терминах языка UML определены следующие виды диаграмм: диаграмма вариантов использования, диаграмма классов, диаграмма деятельности, диаграмма последовательности, диаграмма компонентов.

Суть диаграммы вариантов использования состоит в том, что проектируемая система представляется в виде множества сущностей или актёров, взаимодействующих с системой с помощью, так называемых, вариантов использования.

Суть диаграммы вариантов использования состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых вариантов использования.

Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы.

Пользователь представляет собой внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует её функциональные возможности для достижения определённых целей или решения частных задач. При этом актёры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой. Каждый актёр может рассматриваться как некоторая отдельная роль относительно конкретного варианта использования. Суть диаграммы вариантов использования состоит в том, что проектируемая система представляется в виде множества сущностей или актёров, взаимодействующих с системой с помощью, так называемых, вариантов использования.

Данная программа имеет следующие основные функции:

– осуществление ведение базы данных;

– осуществление работы с базой данных;

– контроль календаря выплат;

– осуществление расчёта статистики;

– создание отчётов.

К вспомогательным функциям, расширяющим возможности системы относятся следующие функции:

– предоставление информации о водителях;

– предоставление информации о пунктах разгрузки и погрузки;

– предоставление информации о заказах;

– предоставление информации о заказчиках;

– предоставление информации о исполнителях;

– предоставление информации о выплатах;

– предоставление справочной информации.

Диаграмма вариантов использования представлена Диаграмма вариантов использования представлена в графической части на листе 1.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы. Диаграмма классов для проектируемой системы представлена в графической части на листе 2.

Полное описание некоторых основных классов проекта, которое состоит из трех разделов, представлено в таблице 1.1.

Таблица 1.1 – Состав некоторых основных классов проекта

|  |  |  |
| --- | --- | --- |
| Имя класса | Имя поля | Методы |
|  |  |  |

При моделировании поведения проектируемой или анализируемой системы возникает необходимость детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, переход в следующее состояние срабатывает только при завершении этой операции. Графически диаграмма деятельности представляется в форме графа, вершинами которого являются состояния действия, а дугами - переходы от одного состояния действия к другому. Основная цель использования диаграмм деятельности - визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения. Диаграмма деятельности для функции поиска информации представлена в графической части на листе 5.

Для моделирования взаимодействия объектов в UML используются соответствующие диаграммы взаимодействия. Если рассматривать взаимодействия объектов во времени, тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.

Временной аспект поведения имеет существенное значение при моделировании синхронных процессов, описывающих взаимодействия объектов. Именно для этой цели и используются диаграммы последовательности, в которых ключевым моментом является динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения: одно - слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии; второе - вертикальная временная ось, направленная сверху вниз, на которой начальному моменту времени соответствует самая верхняя часть диаграммы. Диаграмма последовательности для проектируемой системы представлена в графической части на листе 4.

Рассмотренные ранее диаграммы отражали концептуальные аспекты построения модели системы и относились к логическому уровню представления. Особенность логического представления заключается в том, что оно оперирует понятиями, которые не имеют самостоятельного материального воплощения. Другими словами, различные элементы логического представления, такие как классы, ассоциации, состояния, сообщения, не существуют материально или физически. Они лишь отражают наше понимание структуры физической системы или аспекты ее поведения.

Основное назначение логического представления состоит в анализе структурных и функциональных отношений между элементами модели системы. Однако для создания конкретной физической системы необходимо некоторым образом реализовать все элементы логического представления в конкретные материальные сущности. Для описания таких реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление модели.

Диаграмма компонентов описывает объекты реального мира - компоненты программного обеспечения. Эта диаграмма позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами. Вид диаграммы компонентов для данной проектируемой системы представлен в графической части на листе 2. Она содержит следующие компоненты:

– программные компоненты, созданные в среде Visual Studio: «.сs», «.xsc», «.resx»;

– файл базы данных «database.mdf»;

– файл справочной системы «about\_info.chm».

# Вычислительная система

## 2.1 Требования к аппаратным и операционным ресурсам

Конфигурация компьютера, на котором будет разрабатываться программное приложение:

– процессор AMD A6 2,60 Ггц;

– оперативная память DDR3 4Гбайт;

– жёсткий диск HDD Baracuda 1Тбайт;

– видеокарта AMD Radeon R5 450 4Гбайт;

– материнская плата Asus;

– монитор Viewsonic 24 дюйма.

Для удобной работы с программой необходимо наличие мыши и клавиатуры.

## Инструменты разработки

Инструментами разработки программного средства будут являться:

– операционная система Windows 10;

– среда разработки Microsoft Visual Studio 2017;

– язык программирования Visual С#;

– система управления базами данных Microsoft SQL Server 2017;

– программа для создания справки Help & Manual;

– программа для создания диаграмм UMLet;

– офисный пакет MS Office;

– программа для создания инсталлятора Smart Install Maker.

Windows 10 – многозадачная мультимедийная система семейства dos.

Microsoft Visual Studio 2017 – средство разработки [программного обеспечения](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), разрабатываемое корпорацией [Microsoft](http://ru.wikipedia.org/wiki/Microsoft) и включающее [язык программирования](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) и [среду разработки](http://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8). В то же время Visual Studio 2017 [объектно-ориентированны](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)й язык программирования. Среда разработки Visual Studio 2017 включает инструменты для визуального конструирования пользовательского интерфейса [11].

Среда разработки Visual Studio (VS) включает инструменты для визуального конструирования пользовательского интерфейс. Visual Studio считается хорошим [средством быстрой разработки](http://ru.wikipedia.org/wiki/RAD_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29) прототипов программы, для разработки приложений баз данных и вообще для компонентного способа создания программ, работающих под управлением операционных систем семейства [Microsoft Windows](http://ru.wikipedia.org/wiki/Microsoft_Windows) [12].

Microsoft SQL Server 2017 – система управления реляционными базами данных (СУРБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими Система управления базами данных (СУБД) в этом сегменте рынка [10].

Help & Manual – это приложение для быстрого создания файлов справки (help-файлов), справочных систем, on-line руководств пользователя, пособий и технической документации к программному обеспечению и техническим системам. Help & Manual 7 поддерживает вывод документации в такие форматы, как hypertext markup language (HTML) (on-line руководства), chemical tanker (CHM) (файлы справки для операционной системы (ОС) MS Windows®), rich text format (RTF), и portable document format (PDF). И все это на основе единого проекта-источника [13].

UMLet – это [инструмент Unified Modeling Language (UML](https://en.wikipedia.org/wiki/UML_tool)) с [открытым исходным кодом на](https://en.wikipedia.org/wiki/Open-source) основе [Java](https://en.wikipedia.org/wiki/Java_(programming_language)" \o "Java (язык программирования))[,](https://en.wikipedia.org/wiki/UML_tool)предназначенный для обучения [унифицированному языку моделирования](https://en.wikipedia.org/wiki/Unified_Modeling_Language)и для быстрого создания диаграмм UML. Это инструмент рисования, а не инструмент моделирования, поскольку нет базового словаря или каталога объектов многократного использования. UMLet распространяется по [лицензии General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) (GNU) . UMLet имеет простой пользовательский интерфейс, который использует коды форматирования текста для изменения основных фигур с помощью украшений и аннотаций, поэтому нет леса икон или диалогов списка параметров по-своему. Это требует от пользователя изучения еще одного языка разметки текста, но усилия небольшие, а разметка очевидна для опытного дизайнера [14].

[Microsoft Office](https://ru.wikipedia.org/wiki/Microsoft_Word)  –  [офисный пакет](https://ru.wikipedia.org/wiki/%D0%9E%D1%84%D0%B8%D1%81%D0%BD%D1%8B%D0%B9_%D0%BF%D0%B0%D0%BA%D0%B5%D1%82) [приложений](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), созданных корпорацией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft" \o "Microsoft) для операционных [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows" \o "Microsoft Windows), [Windows Phone](https://ru.wikipedia.org/wiki/Windows_Phone" \o "Windows Phone), [Android](https://ru.wikipedia.org/wiki/Android" \o "Android), [OS X](https://ru.wikipedia.org/wiki/OS_X), [iOS](https://ru.wikipedia.org/wiki/IOS" \o "IOS). В состав этого пакета входит программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных и др. Microsoft Office является сервером [OLE](https://ru.wikipedia.org/wiki/Object_Linking_and_Embedding)-объектов и его функции могут использоваться другими приложениями, а также самими приложениями Microsoft Office. Поддерживает скрипты и макросы, написанные на [VBA](https://ru.wikipedia.org/wiki/Visual_Basic_for_Applications). [9].

Smart Install Maker – утилита предоставляет алгоритм Cabinet для сжатия и создания компактных инсталляторов, поддерживает создание многоязычного инсталлятора (до 25 языков), а также предоставляет средства для управления внешним видом программы-установки [15].

# 3 Проектирование задачи

## 3.1 Требования к приложению

Данное приложение не требует никаких специальных средств защиты либо ограничений прав доступа к данным.

Разрабатываемое приложение должно иметь понятный и удобный в использовании интерфейс, чтобы взаимодействовать между программой и пользователям было максимально упрощённо. Добавление, удаление и изменение будет происходить с помощью ленточной формы. Для более удобного перемещение между таблицами будут использованы вкладки. Программное приложение должно обеспечить оперативный поиск необходимой информации по различным критериям, для этого будут разработаны процедуры обработки информации.

Все входные данные должны проверяться на ошибки. При совершении пользователям ошибки, ему будет предоставлено диалоговое окно с разъяснением ошибки.

Кроме этого, формы в приложении должны быть выполнены в едином стиле, сдержанной цветовой гамме, иметь стандартные элементы управления данными, не должны быть перегружены информацией, и в свое время понятны простому пользователю.

При конструировании форм в необходимых случаях нужно предусмотреть возможность защиты данных от изменений, например, при просмотре информации, установить ограничения на корректировку записей, а также отсутствие данных при выполнении отбора записей в базе данных.

В создаваемом программном средстве должна быть справочная система, в которой будут предусмотрены все возможные способы работы системы. Справочная система должна содержать ответы на вопросы, которые могут возникнуть у пользователя.

## 3.2 Концептуальный прототип

Концептуальный прототип состоит из описания внешнего пользовательского интерфейса, а именно, элементов управления.

При создании данного приложения важную роль играют формы, так как они являются основным диалоговым средством работы пользователя. Разрабатываемое приложение будет содержать главную форму и две дополнительных. Такая структура интерфейса позволит классифицировать основные функции программы по группам.

При проектировании концептуального прототипа предполагается, что при запуске программы первой будет загружаться форма «Главное меню», макет которой представлен на рисунке 3.1.



Рисунок 3.1

На главной форме располагаются кнопки «Актуализация данных», «Просмотр данных», которые осуществляют переход на соответствующие формы. Кнопки «Справка», «Выход» предназначены для открытия справочной системы и выхода из программы соответственно. Переходы на другие формы осуществляется с помощью компонента «Button».

На форме «Актуализация данных» располагаются текстовые поля для ввода данных, таблицы для просмотра информации, кнопки «Добавить», «Удалить», «Изменить», «Сохранить» для работы с данными и кнопка «Назад» для возврата на главную форму. Макет формы «Актуализация данных» представлен на рисунке 3.2.

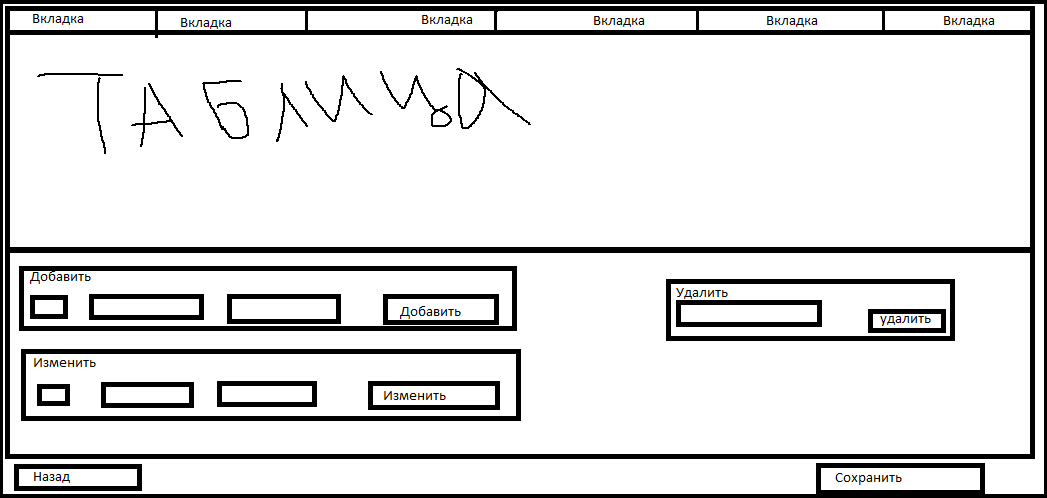


Рисунок 3.2

На форме «Актуализация данных» присутствует система вкладок для работы с другими таблицами соответственно: «Страна», «Награды», «Отчего прививка», «Прививки», «Место размещения электронного чипа», «Породы», «Владельцы», «Паспорт животного», «Получение награды», «Ветеринарное свидетельство». Макеты форм описанных вкладок аналогичные, представленному на рисунке 3.2. Для перехода по вкладкам используется компонент «Tab Control», для отображения данных из базы данных, используется компонент «Data Grid View», для ввода данных используется компонент Text Box, для добавления, удаления и изменения используются компоненты «Button».

На форме «Просмотр данных» располагаются текстовые поля для сортировки и поиска данных, таблицы для просмотра информации, кнопки «Сортировать / поиск», «Создать отчёт» и кнопка «Назад» для возврата на главную форму. Макет формы «Просмотр данных» представлен на рисунке 3.3.

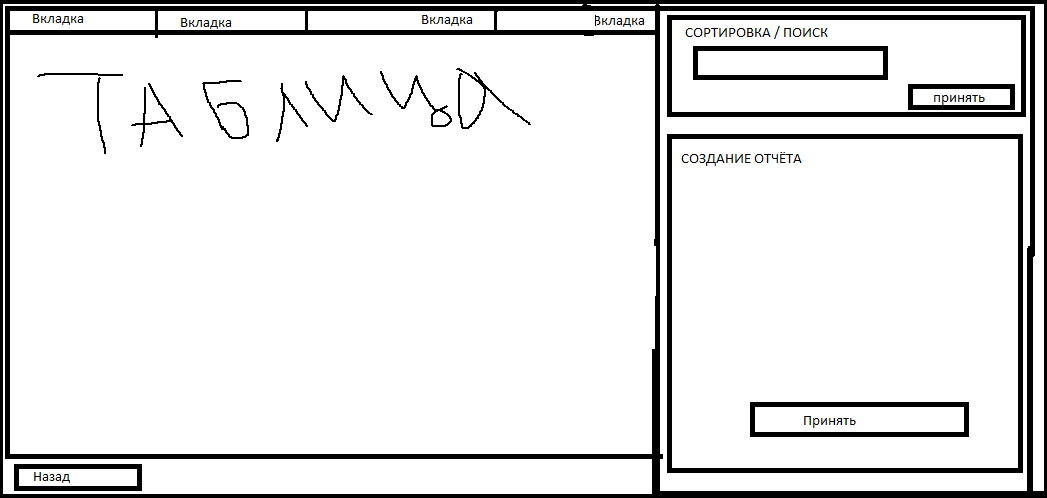


Рисунок 3.3

На форме «Просмотр данных» присутствует система вкладок для работы с другими таблицами соответственно: «Владельцы», «Паспорт животного», «Ветеринарное свидетельство», «Награды». Макеты форм соответствующих вкладок аналогичные, представленному на рисунке 3.3. Для перехода по вкладкам используется компонент «Tab Control», для отображение данных из таблицы базы данных используется компонент «Data Grid View», для ввода данных используется компонент «Text Box», для сортировки, поиска, создания отчётов используются компоненты «Button».

## 3.3 Организация данных

Организация данных подразумевает создание модели данных, главными элементами которой являются сущности и их связи.

Реляционная модель основана на математическом понятии отношения, представлением которого является таблица. В реляционной модели отношения используются для хранения информации об объектах, представленных в базе данных. Отношение имеет вид двухмерной таблицы, в которой строки соответствуют записям, а столбцы - атрибутам. Каждая запись должна однозначно характеризоваться в таблице. Для этого используют первичные и вторичные ключи. Достоинством реляционной модели является простота и удобство физической реализации.

Реляционная модель базы данных подразумевает нормализацию всех таблиц данных. Нормализация – это формальный метод анализа отношений на основе их первичного ключа и функциональных зависимостей, существующих между их атрибутами.

Структура базы данных разрабатываемого программного средства включает 9 таблиц.

Структура данных таблиц, и их краткое описание приводится в таблицах 1-9.

Таблица «PAY» хранит информацию о выплатах. Структура приведена в таблице 3.1.

Таблица 3.1 - Структура таблицы «PAY»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| ID | int | 4 | Номер выплаты |
| PAYMENT\_DATE | date | 3 | Дата совершения выплаты |
| PRICE | money | 8 | Сумма выплаты |

Таблица «BANK» хранит информацию о банках. Структура приведена в таблице 3.2.

Таблица 3.2 - Структура таблицы «BANK»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| ID | int | 4 | Идентификатор банка |
| ADDRES | varchar | 30 | Адрес |
| TITLE | varchar | 30 | Название банка |

Таблица «BANK\_ACCOUNT» хранит информацию о расчетных счетах. Структура приведена в таблице 3.3.

Таблица 3.3 - Структура таблицы «BANK\_ACCOUNT»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| ID | int | 4 | Идентификатор банка |
| BANK\_ID | varchar | 30 | Адрес |
| MONEY\_COUNT | varchar | 30 | Название банка |

Таблица «CONTRAGENT» хранит информацию о клиентах и исполнителях. Структура приведена в таблице 3.4.

Таблица 3.4 - Структура таблицы «CONTRAGENT»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| ID | int | 4 | Идентификатор банка |
| ADDRES | varchar | 30 | Адрес |
| PASSPORT\_NUMBER | varchar | 7 | Название банка |
| PASSPORT\_REG | varchar | 2 |  |
| PHONE\_NUMBER | varchar | 9 |  |
| TITLE | varchar | 60 |  |
| TYP | bit | 1 |  |
| BANK\_ACCOUNT\_ID | int | 4 |  |

Таблица «FULL\_ORDER» хранит информацию о заказах. Структура приведена в таблице 3.5.

Таблица 3.5 - Структура таблицы «FULL\_ORDER»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| ID | int | 4 | Идентификатор заказа |
| SAMMI\_ORDER\_ID | money | 8 | Цена заказа |
| DATE\_OF\_CONCLUSION | date | 3 | Дата заключения договора |
| CUSTOMER\_ID | date | 3 | Дата финальной выплаты |
| ORDERER | bit | 1 | Состояние :(оплачено, не оплачено) |

Таблица «SAMMI\_ORDER» хранит информацию о банках. Структура приведена в таблице 3.6.

Таблица 3.6 - Структура таблицы «SAMMI\_ORDER»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| ID | int | 4 | Идентификатор банка |
| LOADING\_POINT\_ID | int | 4 | Адрес |
| TRANSPORT\_ID | int | 4 | Имя банка |
| DRIVER\_ID | int | 4 |  |
| PAY\_ID | int | 4 |  |
| PENALTY | money | 8 |  |
| DATE\_OF\_EXECUTION | date |  |  |
| PAYMENT\_DATE | date |  |  |

Таблица «DRIVER» хранит информацию о водителях. Структура приведена в таблице 3.7.

Таблица 3.7 - Структура таблицы «DRIVER»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| ID | int | 4 | Идентификатор типа |
| FIO | varchar | 60 | Фамилия, имя ,отчество водителя |

Таблица «TRANSPORT» хранит информацию о транспортных средствах. Структура приведена в таблице 3.8.

Таблица 3.8 - Структура таблицы «TRANSPORT»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| ID | int | 4 | Идентификатор транспортного средства |
| PRICE | money | 8 | Тип транспорта |
| ID\_TRANSPORT\_TYPE | int | 4 | Идентификатор типа транспортного средства |
| CAPACITY | int | 4 | Грузоподъёмность |

Таблица «TRANSPORT\_TYPE» хранит информацию о типах транспортных средств. Структура приведена в таблице 3.9.

Таблица 3.9 - Структура таблицы «TRANSPORT\_TYPE»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| ID | int | 4 | Идентификатор типа |
| TITLE | varchar | 20 | Название типа |

Таблица «LOADING\_POINT» хранит информацию о пунктах погрузки и разгрузки. Структура приведена в таблице 3.10.

Таблица 3.10 - Структура таблицы «LOADING\_POINT»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| ID | int | 4 | Номер пункта |
| ADDRES | varchar | 30 | Адрес |
| PRICE |  |  |  |

Физическая структура базы данных представлена на рисунке 3.4

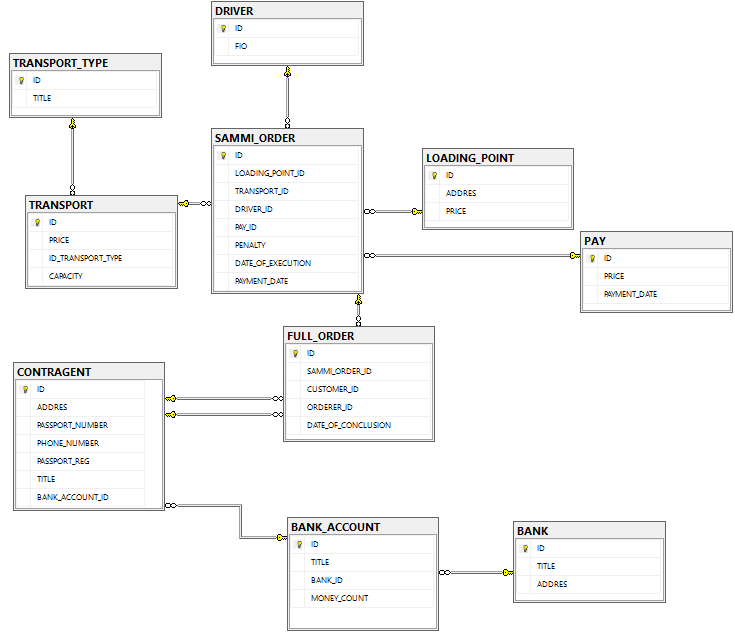


Рисунок 3.4

## 3.4 Функции и элементы управления

Основными функциями данного проекта являются подсчёт статистической информации.

## 3.5 Проектирование справочной системы

Для корректной работы с приложением требуется обеспечить пользователя справочной системой, в которой будут приведены приемы работы с приложением, включающие данные о том, что произойдёт после нажатия на определенную кнопку или при выборе вкладки.

Справочная система необходима для ознакомления с программой. В ней должна присутствовать информация, которая поможет в решении проблемы с приложением, а также может напомнить, как пользоваться программным средством.

Система справки данного приложения будет содержать следующие разделы:

– «Введение»;

– «Инструкция»;

– «Об авторе».

Справочная система будет создана с помощью программного средства Help & Manual.

Help & Manual – это приложение для быстрого создания файлов справки (help-файлов), справочных систем, on-line руководств пользователя, пособий и технической документации к программному обеспечению и техническим системам. Help & Manual 7 поддерживает вывод документации в такие форматы, как hypertext markup language (HTML) (on-line руководства), chemical tanker CHM (файлы справки для операционной системы (ОС) MS Windows), rich text format (RTF), и portable document format (PDF). И все это на основе единого проекта-источника [13].

Справка будет вызываться клавишей f1 на всех формах, а также при нажатии на кнопку «Справка», которая находиться на главном меню.

# 4 Описание программного средства

## 4.1 Общие сведения

Приложение «TransOrder.exe» предназначено для учета клиентов их заказов, и сроков выплат. Файл базы данных называется «Database.mdb» вес, которого составляет 10 Мбайт. Само приложение весит 120 Мбайт. Для успешного использования программы, на компьютере должно быть свободного места на диске не меньше 200 Мбайт, оперативная память должна быть от 4000 Мбайт. Приложение использует стандартные библиотеки такие как:

– System.Configuration;

– System.Data;

– System.Data.SqlClient;

– Excel = Microsoft.Office.Interop.Excel;

– Microsoft.Office;

– Microsoft.Office.Interop;

– System.IO;

– System.Reflection;

– Microsoft.Reporting;

– Microsoft.ReportingServices;

Для запуска приложения всего лишь запустить ярлык «TransOrder.exe».

Компьютер должен работать под управлением операционной системы, начиная с Windows 10 и выше.

Инсталляция производится с установочного диска.

## 4.2 Функциональное назначение

Назначение программного средства «TransOrder» является учет оплаты клиентами выполненных заказов для организаций занимающихся доставкой товаров.

В данной программе классом решаемых задач являются:

Решаемые задачи курсового проекта:

– возможность добавлять, изменять и удалять данные в таблицы;

– возможность сортировать, фильтровать и искать данные в различных таблицах;

– возможность создавать отчёты по различным критериям.

Сетевая поддержка не предусмотрена.

## 4.3 Входные и выходные данные

# 6 Применение

## 6.1 Назначение программы

Программа предназначена для администрирования данных о выполненных заказах «», предназначена для своевременной и полной оплаты заказов и мониторинга состояния заказов. В этой программе возможно составление договора и отчетов по различным критериям, так же и статические отчеты о реализованных выплатах и заказах.

## 6.2 Условия применения

Для применения данного программного средства необходимы следующие технически требования:

* процессор Intel Core 2 Duo или выше;
* минимальный объем оперативной памяти — 2000 Мбайт;
* операционная система Windows 10 и выше;
* Framework v4.6.1 и выше
* устройство для чтения дисков.

Программа адаптирована под все компьютеры. Для переноса программы на другой компьютер достаточно будет перенести папку с проектом и до установить необходимый фрэймворк.

## 6.3 Справочная система

Справочная система программного средства представляет собой отдельный файл «TransOrder \_Info.chm» с полным описанием основных функций программы в формате \*.chm. В справочной системе даны ответы на типичные вопросы, возникающие при работе с приложением, что, несомненно, должно помочь при освоении программного средства.

Справка имеет следующие разделы:

– «Введение»;

­– «Инструкция»;

– «Об авторе».

Структура справочной системы представлена на рисунке 6.1.

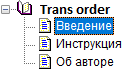


Рисунок 6.1

В разделе «Введение» рассказывается, что из себя представляет программное средство. Содержимое раздела представлено на рисунке 6.2.

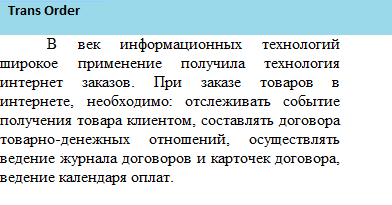


Рисунок 6.2

В разделе «Инструкция» рассказывается подробно как пользоваться программой. Содержимое раздела представлено на рисунке 6.3.

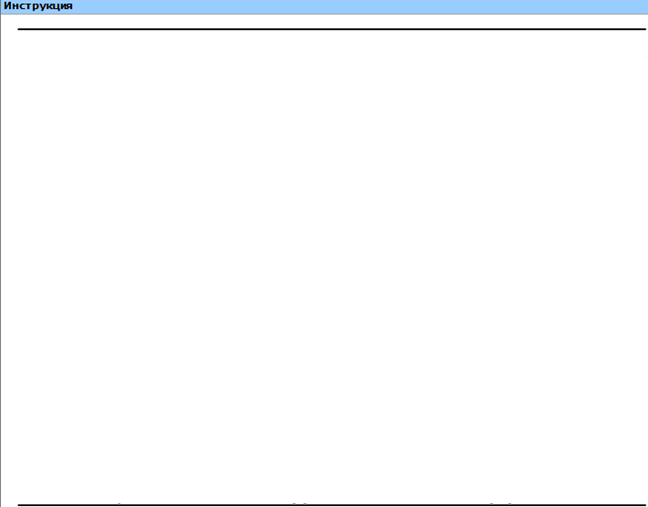


Рисунок 6.3

В разделе «Об авторе» рассказывается об человеке, который создал это программу. Содержание раздела представлено на рисунке 6.4.



Рисунок 6.4

Заключение

В рамках курсового проектирования по дисциплине «Базы данных и системы управления базами данных», было разработано программное средство «TransOrder.exe» автоматизирующее учет клиентами выполненных заказов.

Для достижения цели курсового проекта были решены следующие задачи:

* определена вычислительная система необходимая создания программного средства;
* разработаны логическая и физическая модели данных;
* по модели выполнено проектирование задачи;
* разработано программное средство;
* описано создано программное средство;
* выбрана методика испытаний;
* описан процесс функционального тестирования;
* приведены примеры в области применения.

Разработка имеет интуитивно понятный графический интерфейс, позволяющий даже с минимальным знанием компьютера, использовать данное программное средство.

В процессе разработки данного программного средства были применены и закреплены знания по уже изученному материалу.

Были отработаны навыки владения методами надежного программирования и эффективной разработки программного обеспечения в Microsoft vs 2017 с использованием язык программирования C#, разработана база данных средствами системы управления базами данных Microsoft SQL server 2017.

Программа реализована в полном объеме и в соответствии с техническим заданием полностью отлажена и протестирована. Поставленные задачи выполненные.

(Программа реализована не в полном объеме и не полностью соответствует техническому заданию, однако отлажена и протестирована, планируется доработать в следующих версиях программы. Программное средство готово к практическому использованию «»)