**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc41469483)

[1 Исследовательский раздел 6](#_Toc41469484)

[1.1 Анализ предметной области 6](#_Toc41469485)

[1.2 Требования к разработке программного продукта 8](#_Toc41469486)

[1.3 Описание программных средств для реализации информационный системы 8](#_Toc41469487)

[1.4 CASE средства для проектирования информационной системы 11](#_Toc41469488)

[2 Специальный раздел 13](#_Toc41469489)

[2.1 Постановка задачи 13](#_Toc41469490)

[2.2 Проектирование программных средств 13](#_Toc41469491)

[2.3 Разработка структуры БД 14](#_Toc41469492)

[3. Технический раздел 18](#_Toc41469493)

[3.1 Описание интерфейса информационной системы 18](#_Toc41469494)

[3.2 Программная реализация заполнения данных из БД 20](#_Toc41469495)

[3.3 Программная реализация и описание интерфейса администратора 24](#_Toc41469496)

[ЗАКЛЮЧЕНИЕ 40](#_Toc41469497)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 41](#_Toc41469498)

# ВВЕДЕНИЕ

Современное население мира сейчас мало может себе представить свое существование без Интернета. Электронная торговля — это торговля через сеть при помощи компьютеров покупателя и продавца товара, предметом электронной торговли может быть любой товар, услуга, недвижимость, банковский продукт и т.п. На данный момент развитие сети Интернет способствует резкому росту этой технологии торговли среди всех торговых фирм и граждан. Теперь фирмы, малые предприятия, покупатели и др. участники рынка получили возможность работать в режиме онлайн, тем самым данный вид торговли облегчил сотрудничество людей

Централизованная доставка осуществляется силами и средствами поставщика и поэтому является наиболее эффективным методом товароснабжения. Так как, в его обязанности при этом входит погрузка транспортных средств, перевозка грузов собственным автотранспортом или принадлежащим автотранспортным организациям, обеспечение сохранности и своевременной доставки товаров в магазин.

Один из наиболее важных аспектов применения логистической информационной системы (ЛИС) в деятельности компании – работа с поступающими заказами клиентов.

Не менее важный аспект применения ЛИС в компании – ее использование в аналитической работе. При этом различают три уровня ее применения:

– макроуровень, внешняя среда логистики;

– микроуровень, внутрифирменная среда;

– межфирменный уровень, связи с другими компаниями.

В этом аспекте важны вопросы, находящиеся в компетенции менеджера по логистике компании и касающиеся использования электронно-вычислительной техники:

– интеграция планирования логистических операций с планированием на уровне компании в целом;

– интеграция операций по логистике с операциями, осуществляемыми компанией в целом;

– разработка стратегии для отдела логистики и разработка заданий для персонала этого отдела;

– интеграция информационных технологий, используемых в области логистики в рамках технологических сетей компании;

– интеграция информационных технологий с компаниями-партнерами;

В настоящее время существует достаточно большое количество логистических компаний. Актуальность темы курсовой работы обусловлена необходимостью автоматизации деятельности логистических компаний с целью повышения эффективности их деятельности.

Целью курсовой работы является разработка и проектирование информационной системы логистической компании.

Для достижения данной цели были поставлены следующие задачи:

1. Изучение технологий проектирования информационных систем.
2. Проектирование информационной системы предприятия.
3. Создание базы данных для приложения.
4. Разработка и описание интерфейса приложения.
5. Реализация подключения БД к среде разработки Visual Studio.
6. Реализация приложения в среде разработки Visual Studio на языке С#.

# Исследовательский раздел

## Анализ предметной области

Процесс заказа посылок осуществляется следующим образом:

1. Пользователь регистрируется в приложении.
2. Выбирает подходящий для него товар из предложенного списка.
3. Создаётся номер отслеживания и с этим номером добавляется товар в корзину.
4. В корзине создается заказ либо отменяется.

В соответствии с задачами необходимо разработать информационную систему «Доставка посылок». Проектирование базы данных начинается с анализа предметной области и выявления требований к ней. Необходимо разработать структуру и основные объекты базы данных «Доставка посылок», а также спроектировать главную кнопочную форму.

База данных состоит из пяти таблиц:

1. Товар (ID, название, категория, вес, цена, описание, количество в наличии, ID поставщика).
2. Заказ (ID, заказчик, товар, количество, номер отслеживания, доставлено).
3. Отслеживание (номер отслеживания, точка отправки, дата заказа, время на доставку).
4. Пользователь (ID, логин, пароль, ФИО, телефон, адрес).
5. Поставщик (ID, название поставщика, юридический адрес, адрес склада, дней на доставку).

Результирующей формой разрабатываемого приложения будет удобный пользовательский интерфейс для анализа информации, введенной пользователем или уже имеющейся в памяти компьютера, отвечающий предъявленным требованиям.

Входной информацией в приложении является данные, вводимые непосредственно оператором либо администратором базы данных. Технология ввода осуществляется стандартным путем, с помощью клавиатуры или с помощью мыши.

Выходная информация - информация, соответствующая предъявленным требованиям к разработанному приложению: таблицы, запросы, отчеты.

Процессы поставок в компаниях характеризуются нестабильным положением на локальных рынках, неоднозначными условиями среди потребителей. Поэтому наряду с усовершенствованием транспортно-распределительной логистической сети в компаниях, необходимо проводить маркетинговые исследования. Транспортно-распределительная логистика является главным показателем повышения конкурентоспособности любого предприятия. Анализ транспортно-распределительной логистической сети в компаниях обуславливает необходимость внедрения новых методов для компаний. Базируясь на мировом опыте, смело будет предположить, что необходимо внедрять новейшие в своём роде технологии, использовать передовые методы современной логистики, чтобы максимально совершенствовать систему грузоперевозок.

В условиях современных рыночных отношениях для повышения конкурентоспособности предприятия необходимо максимально снижать любые затраты, в частности это относится к затратам на производстве и логистическую деятельность.

К логистической деятельность организации относится:

1. Планирование и закупка материальных ресурсов.
2. Транспортировка и складирование материальных ресурсов.
3. Обработка заказов и контроль запасов.
4. Организация финансовой деятельности.
5. Создание систем связи и обработки данных.
6. Организация производства и распределения готовой продукции.

## Требования к разработке программного продукта

Разрабатывается информационная система для логистической компании.

Для приложения устанавливаются следующие требования:

1. Поиск информации по заказам;
2. Хранение информации о заказах;
3. Хранение информации о клиентах;
4. Хранение информации о обновлении базы данных;
5. Отслеживание заказов;
6. Отображение всех имеющихся в наличии товаров;
7. Обновление базы данных;

## Описание программных средств для реализации информационный системы

Проектирование информационной системы произведено в AllFusion Process Modeler.

AllFusion Process Modeler 7 (ранее BPwin) - инструмент для моделирования, анализа, документирования и оптимизации бизнес-процессов. AllFusion Process Modeler 7 можно использовать для графического представления бизнес-процессов. Графически представленная схема выполнения работ, обмена информацией, документооборота визуализирует модель бизнес-процесса. Графическое изложение этой информации позволяет перевести задачи управления организацией из области сложного ремесла в сферу инженерных технологий.

AllFusion Process Modeler 7 (BPwin) помогает четко документировать важные аспекты любых бизнес-процессов: действия, которые необходимо предпринять, способы их осуществления и контроля, требующиеся для этого ресурсы, а также визуализировать получаемые от этих действий результаты. AllFusion Process Modeler 7 повышает бизнес-эффективность ИТ-решений, позволяя аналитикам и проектировщикам моделей соотносить корпоративные инициативы и задачи с бизнес-требованиями и процессами информационной архитектуры и проектирования приложений. Таким образом, формируется целостная картина деятельности предприятия: от потоков работ в небольших подразделениях до сложных организационных функций.

AllFusion Process Modeler 7 (BPwin) эффективен в проектах, связанных с описанием действующих баз предприятий, реорганизацией бизнес-процессов, внедрением корпоративной информационной системы. Продукт позволяет оптимизировать деятельность предприятия и проверить ее на соответствие стандартам ISO 9000, спроектировать оргструктуру, снизить издержки, исключить ненужные операции и повысить эффективность. В основу продукта заложены общепризнанные методологии моделирования, например, методология IDEF0 рекомендована к использованию Госстандартом РФ[14].

Для разработки ПО «проектирование информационной системы логистической компании» была выбраны следующие программные средства:

1. Microsoft Access Database
2. Visual Studio

Microsoft Office Access – мощный редактор, который представляет собой систему управления базами данных, основанными на реляционной модели. Также необходимо отметить, что он использует динамическую модель обмена между сетевыми ресурсами и приложениями. При этом редактор использует продвинутые инструменты для обработки любого типа информации и представление их в виде четкой последовательной структуры.

В Access также внедрена поддержка динамической библиотеки ActiveX. Эта технология, которая помогает не только представлять информацию в виде текста, а также в виде объектов мультимедии. Реляционная модель позволяет устанавливать связи между базами и контролировать любое изменение, своевременно внося коррективы. Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

Рисунок 1 - Интерфейс Visual Studio

На рисунке показана среда Visual Studio с открытым проектом и несколькими окнами основных инструментов, которые вам, скорее всего, понадобятся:

1. Обозреватель решений (вверху справа) позволяет просматривать файлы кода, перемещаться по ним и управлять ими. Обозреватель решений позволяет упорядочить код путем объединения файлов в решения и проекты.
2. В окне редактора (центр), где вы, скорее всего, будете проводить большую часть времени, отображается содержимое файла. Здесь вы можете редактировать код или разрабатывать пользовательский интерфейс, например, окно с кнопками или текстовые поля.

Team Explorer (правый нижний угол) позволяет отслеживать рабочие элементы и использовать код совместно с другими пользователями с помощью технологий управления версиями, таких как Git и система управления версиями Team Foundation (TFVC) [5].

## CASE средства для проектирования информационной системы

В курсовой работе используется CASE-средства BPwin.

BPwin – мощный инструмент моделирования, который используется для анализа, документирования и реорганизации сложных бизнес-процессов.

BPwin позволяет определить точки конфликтов и достичь их согласования.

Основные возможности BPwin:

* 1. Моделирование функций (IDEF0) – систематический анализ бизнеса и рассмотрение регулярно решаемых задач-функций, ресурсов, результатов;
	2. Моделирование потоков данных (DFD), передающихся между различными операциями;
	3. Моделирование потоков работ (IDEF3) – анализ операций процесса, а также точек принятия решений, влияющих на ход процесса.

Система представляется как совокупность взаимодействующих работ или функций. Начало моделирования – это определение контекста. Его составляющие:

* 1. Субъект – это описание того, что входит в систему, а что за ее пределами, при этом учитывается широта и глубина – уровень детализации, области исследования, но после определения границ не рекомендуется вводить новые объекты, так как нарушаются связи (команда Model / Model Properties /Definition);
	2. Цель – это описание, для чего проводится моделирование (команда Model / Model Properties / Purpose / Purpose);
	3. Точка зрения – это определение, указывающее, чья точка зрения отражена в данной модели, но это должна быть точка зрения одного человека, ответственного за моделирование в целом (команда Model / Model Properties /Purpose / Viewpoint).

Для представления ИСП посылок из Китая воспользуемся программным обеспечением AllFusion Process Modeler 7 и создадим модель, описывающую все необходимые компоненты для работы предприятия.

AllFusion Process Modeler - мощный инструмент моделирования, позволяющий визуализировать, анализировать и совершенствовать бизнес-процессы. AllFusion Process Modeler повышает бизнес-эффективность ИТ-решений, предоставляя возможность аналитикам и проектировщикам моделей соотносить корпоративные инициативы и задачи с бизнес-требованиями и процессами, информационной архитектурой и процессом проектирования приложений.

BPwin - мощный инструмент моделирования, разработанный фирмой Computer Associates Technologies который используется для анализа, документирования и реорганизации сложных бизнес-процессов[14].

# Специальный раздел

## Постановка задачи

В ходе выполнения курсовой работы были поставлены следующие задачи:

1. Изучение технологий проектирования информационных систем.
2. Проектирование информационной системы предприятия.
3. Создание базы данных для приложения.
4. Разработка и описание интерфейса приложения.
5. Реализация подключения БД к среде разработки Visual Studio.
6. Реализация приложения в среде разработки Visual Studio на языке С#.

## Проектирование программных средств

На рисунке 2 представлена схема доставки посылок логистической компании, созданная в программе AllFusion Process Modeler.



Рисунок 2 - Модель доставки посылок

## Разработка структуры БД

Для создания БД потребуется Microsoft Access Database

Было создано 5 таблиц:

1. Пользователь
2. Поставщик
3. Товар
4. Заказ
5. Отслеживание

Первая таблица Пользователь. Здесь добавляем поля:

1. ID
2. Логин
3. Пароль
4. ФИО
5. Телефон
6. Адрес

После добавления всех полей, необходимо заполнить их необходимой информацией. Итог представлен на рисунке 3



Рисунок 3 - Таблица пользователь

Следующим действием будет создание таблицы Поставщик

В этой таблице будут следующие поля:

1. ID
2. Название поставщика
3. Юридический адрес
4. Адрес склада
5. Дней на доставку

После заполнения всех полей получим итоговый результат



Рисунок 4 - Таблица Поставщик

Таблица Товар. В ней будет находиться вся информация по посылках.

В данной таблице были добавлены следующие поля:

1. ID
2. Название
3. Категория
4. Вес
5. Цена
6. Описание
7. Количество в наличии
8. ID поставщика

Также заполним все поля.



Рисунок 5 - Таблица товар

Таблица Заказ. В ней будет находиться вся информация о заказанных товарах.

Были добавлены следующие поля:

1. ID
2. Заказчик
3. Товар
4. Количество
5. Номер отслеживания
6. Корзина
7. Добавлено

Была получена итоговая таблица.



Рисунок 6 - Таблица заказ

Таблица Отслеживание. В ней будет находиться вся информация о заказанных товарах.

Были добавлены следующие поля:

1. Номер отслеживания
2. Пункт отправки
3. Пункт прибытия
4. Дата заказа

Получим итоговую таблицу.



Рисунок 7 - Таблица отслеживание

Для того, чтобы можно было отслеживать кто заказал товар, у какого поставщика, адрес и т.д, была сделана связь между всеми таблицами.



Рисунок 8 - Связь между таблицами

В данной схеме устанавливаем связь таблиц Товар, Пользователь, Поставщик, Отслеживание, Заказ. В таблице Посылка установим связь поля Заказчик с кодом таблицы Заказчика. Аналогично делаем связь поля Поставщика с кодом таблицы Поставщик.

Это делается для того, чтобы понять, какая информация к чему относится.

## 3. Технический раздел

## 3.1 Описание интерфейса информационной системы

Для создания приложения была использовона среда разработки Visual Studio на языке C#. В этой среде разработаем программу, в которой можно будет увидеть всю информацию о посылках. Для этого окно авторизации, регистрацию.



Рисунок 9 - Форма авторизации



Рисунок 10 - Форма регистрации

После успешной регистрации, пользователь может войти в приложения.

Ему откроется окно, в котором он может увидеть всю необходимую информацию.

На рисунке 11 пользователь может проводить следующие действия:

Рисунок 11 - Форма пользователь

1. Видеть свои данные после регистрации и редактировать их. Можно сменить пароль в случае необходимости;

Рисунок 12 - Данные после регистрации

1. Видеть перечень всех имеющихся товаров;
2. Выбрать категорию товара. Включить фильтр, в котором можно выбрать сортировку товара. Добавить товар в корзину;

Рисунок 13 - Каталог

Рисунок 14 - Фильтры

1. Посмотреть свои заказы. Оформить заказ. Видеть отслеживание посылки;

Рисунок 15 - Заказы пользователя

## 3.2 Программная реализация заполнения данных из БД

Для примера рассмотрим авторизацию пользователей. При входе в приложение из БД используются данные отвечающие за каждую информацию о любом пользователе. В БД хранятся все учётные записи пользователей. Благодаря этому пользователь может войти в приложение под своими учётными данными.

В листинге 1 приведён фрагмент кода учетной записи пользователя.

Листинг 1 – Программная реализация учетной записи пользователя

using System;

using System.Data.OleDb;

using System.Windows.Forms;

namespace Main

{

 public partial class Authorize : Form

 {

 string db\_path = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Users\Dimon\Desktop\Parcels\_4-6-1\Parcels 4-6-1\Parcels\_app.mdb";

 OleDbConnection db;

 public Authorize()

 {

 InitializeComponent();

 db = new OleDbConnection(db\_path);

 db.Open();

 //DEBUG

 //login\_box.Text = "admin";

 //password\_box.Text = "admin";

 //login\_box.TabIndex = 999;

 //password\_box.TabIndex = 999;

 //END\_DEBUG

 }

 //проверка текста

 private void box\_KeyPress(object sender, KeyPressEventArgs e)

 {

 if ( (e.KeyChar==(char)Keys.Enter) && (sender as TextBox).Name == "password\_box")

 {

 enter\_button\_Click(enter\_button, e);

 }

 if (!char.IsDigit(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar))

 {

 e.Handled = true;

 }

 }

 //вход

 private void enter\_button\_Click(object sender, EventArgs e)

 {

 string login = login\_box.Text, password = password\_box.Text;

 //0 этап: проверяем не пустые ли поля

 if (string.IsNullOrEmpty(login) || string.IsNullOrEmpty(password))

 {

 MessageBox.Show("Поля Логин и Пароль должны быть заполнены.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

 return;

 }

 //1 этап: проверяем наличие логина в базе

 string query = $"SELECT Пользователь.Логин FROM Пользователь WHERE(((Пользователь.Логин) = \"{login}\"));";

 OleDbCommand cmd = new OleDbCommand(query, db);

 if (cmd.ExecuteScalar()==null)

 {

 MessageBox.Show("Пользователь с таким логином не зарегистрирован.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

 return;

 }

 //2 этап: проверяем для выбранного пользователя его пароль

 query = $"SELECT Пользователь.Пароль FROM Пользователь WHERE(((Пользователь.Логин) = \"{login.ToLower()}\"));";

 cmd = new OleDbCommand(query, db);

 string password\_db = cmd.ExecuteScalar().ToString();

 if (password\_db!=password)

 {

 MessageBox.Show("Неверный пароль. Повторите попытку входа", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

 return;

 }

 //3 этап: выполняем вход для выбранного пользователя

 query = $"SELECT Пользователь.ID FROM Пользователь WHERE(((Пользователь.Логин) = \"{login}\"));";

 cmd = new OleDbCommand(query, db);

 Hide();

 if (login != "admin")

 {

 UserForm user = new UserForm(db, Convert.ToInt32(cmd.ExecuteScalar()));

 user.Show();

 }

 else

 {

 AdminForm admin = new AdminForm(db);

 admin.Show();

 }

 }

 //открыть регистрацию

 private void register\_link\_Click(object sender, EventArgs e)

 {

 Register register = new Register(db);

 register.ShowDialog();

 }

 //закрытие формы

 private void Log\_IN\_FormClosing(object sender, FormClosingEventArgs e)

 {

 db.Close();

 Dispose();

 }

 }

}

## 3.3 Программная реализация и описание интерфейса администратора

Чтобы вести контроль всех учётных записей была создана учётная запись администратора. При входе в эту учётную запись предоставляется полный доступ ко всей информации. Администратор может редактировать данные, изменять, добавлять.

В листинге 2 приведен фрагмент кода формы «Администратор»:

Листинг 2 – Программный код формы «Администратор»

using System;

using System.Collections.Generic;

using System.Data.OleDb;

using System.Drawing;

using System.Windows.Forms;

namespace Main

{

 public partial class AdminForm : Form

 {

 OleDbConnection db;

 List<string[]> contractors\_id\_name = new List<string[]>();

 List<string[]> products\_array = new List<string[]>();

 List<string[]> contractors\_array = new List<string[]>();

 List<string[]> users\_array = new List<string[]>();

 public AdminForm(OleDbConnection db)

 {

 this.db = db;

 InitializeComponent();

 initialize\_data();

 }

 //инициализация начальная

 void initialize\_data()

 {

 get\_products\_data();

 get\_contractors\_data();

 get\_users\_data();

 get\_orders\_data();

 }

 /\*\*\* ДАННЫЕ ПОЛУЧЕНИЕ ИЗ БД\*\*\*/

 // --------- ТОВАР ----------//

 void get\_products\_data(int selected=0)

 {

 contractors\_id\_name.Clear();

 p\_comboBox.Items.Clear();

 products\_array.Clear();

 products\_box.Items.Clear();

 string query = "SELECT Поставщик.ID, Поставщик.[Название поставщика] FROM Поставщик ORDER BY Поставщик.ID;";

 OleDbDataReader reader = new OleDbCommand(query, db).ExecuteReader();

 while (reader.Read())

 {

 contractors\_id\_name.Add(new string[] { reader.GetValue(0).ToString(), reader.GetValue(1).ToString() });

 p\_comboBox.Items.Add(reader.GetValue(1).ToString());

 }

 query = "SELECT Товар.Категория FROM Товар GROUP BY Товар.Категория HAVING(((Товар.Категория)Is Not Null));";

 reader = new OleDbCommand(query, db).ExecuteReader();

 p\_textBox2.Items.Clear();

 List<string> categories = new List<string>();

 while (reader.Read())

 p\_textBox2.Items.Add(reader.GetString(0));

 query = "SELECT Товар.\* FROM Товар ORDER BY Товар.ID;";

 reader = new OleDbCommand(query, db).ExecuteReader();

 while (reader.Read())

 {

 string[] data = new string[8];

 for (int i = 0; i < data.Length; i++)

 {

 data[i] = reader.GetValue(i).ToString();

 }

 products\_array.Add(data);

 products\_box.Items.Add($"ID:{data[0]}.{data[1]}");

 }

 products\_box.SelectedIndex = selected;

 }

 // --------- ПОСТАВЩИКИ -----//

 void get\_contractors\_data(int selected = 0)

 {

 contractors\_array.Clear();

 contractor\_box.Items.Clear();

 string query = "SELECT Поставщик.\* FROM Поставщик ORDER BY Поставщик.ID;";

 OleDbDataReader reader = new OleDbCommand(query, db).ExecuteReader();

 while (reader.Read())

 {

 string[] data = new string[5];

 for (int i = 0; i < data.Length; i++)

 {

 data[i] = reader.GetValue(i).ToString();

 }

 contractors\_array.Add(data);

 contractor\_box.Items.Add($"ID:{data[0]}.{data[1]}");

 }

 contractor\_box.SelectedIndex = selected;

 }

 // -------- ПОЛЬЗОВАТЕЛИ ----//

 void get\_users\_data()

 {

 users\_array.Clear();

 users\_list.Items.Clear();

 string query = "SELECT Пользователь.ID, Пользователь.ФИО FROM Пользователь WHERE(((Пользователь.ID) <> 1));";

 OleDbDataReader reader = new OleDbCommand(query, db).ExecuteReader();

 while (reader.Read())

 {

 users\_list.Items.Add(reader.GetString(1));

 users\_array.Add(new string[] { reader.GetValue(0).ToString(), reader.GetString(1) });

 }

 }

 // -------- ЗАКАЗЫ ----------//

 void get\_orders\_data()

 {

 orders\_list.Items.Clear();

 string query = "SELECT Отслеживание.[Дата заказа], Пользователь.ФИО, Заказ.Количество, Товар.Цена, Товар.Название, Заказ.Доставлено " +

 "FROM Отслеживание INNER JOIN(Пользователь INNER JOIN (Товар INNER JOIN Заказ ON Товар.ID = Заказ.Товар) ON " +

 "Пользователь.ID = Заказ.Заказчик) ON Отслеживание.[Номер отслеживания] = Заказ.[Номер отслеживания] " +

 "WHERE(((Заказ.Корзина)= False));";

 OleDbDataReader reader = new OleDbCommand(query, db).ExecuteReader();

 while (reader.Read())

 {

 string[] data = new string[6];

 for (int i = 0; i < data.Length; i++)

 {

 if (i == 0)

 data[i] = reader.GetDateTime(i).ToShortDateString();

 else

 data[i] = reader.GetValue(i).ToString();

 }

 string order = $"{data[0]} , {data[1]} , Сумма покупки:{Convert.ToInt32(data[2])\* Convert.ToInt32(data[3])} " +

 $"Товар:{data[4]} , ";

 if (data[5] == "True")

 order += "Доставлен.";

 else

 order += "В пути.";

 orders\_list.Items.Add(order);

 }

 }

 /\*\*\* ДАННЫЕ ОБНОВЛЕНИЕ В БД \*\*\*/

 // --------- ТОВАР ----------//

 void update\_product\_data(int product\_id)

 {

 Control[] boxes = new Control[] {

 p\_textBox1,p\_textBox2,p\_textBox3,p\_textBox4,p\_textBox6,p\_textBox5

 };

 string[] tmp = contractors\_id\_name[p\_comboBox.SelectedIndex];

 string query = $"UPDATE Товар SET Товар.Название = \"{boxes[0].Text}\", " +

 $"Товар.Категория = \"{boxes[1].Text}\", " +

 $"Товар.[Вес (кг)] = \"{boxes[2].Text}\", " +

 $"Товар.Цена = {boxes[3].Text}, " +

 $"Товар.Описание = \"{boxes[4].Text}\", " +

 $"Товар.[Количество в наличии] = {boxes[5].Text}, " +

 $"Товар.[ID поставщика] = {tmp[0]} " +

 $"WHERE(((Товар.ID) = {product\_id}));";

 new OleDbCommand(query, db).ExecuteNonQuery();

 get\_products\_data(products\_box.SelectedIndex);

 }

 // --------- ПОСТАВЩИКИ -----//

 void update\_contractors\_data(int contractor\_id)

 {

 Control[] boxes = new Control[] {

 c\_textBox1,c\_textBox3,c\_textBox4,c\_textBox2

 };

 string query = $"UPDATE Поставщик SET " +

 $"Поставщик.[Название поставщика] = \"{boxes[0].Text}\", " +

 $"Поставщик.[Юридический адрес] = \"{boxes[1].Text}\", " +

 $"Поставщик.[Адрес склада] = \"{boxes[2].Text}\", " +

 $"Поставщик.[Дней на доставку] = {boxes[3].Text} " +

 $"WHERE(((Поставщик.ID) = {contractor\_id}));";

 new OleDbCommand(query, db).ExecuteNonQuery();

 get\_contractors\_data(contractor\_box.SelectedIndex);

 }

 /\*\*\* СОБЫТИЯ ТОВАР \*\*\*/

 //текст меняется product

 private void product\_TextChanged(object sender, EventArgs e)

 {

 string text = (sender as TextBox).Text;

 foreach (char item in text)

 {

 if ((sender as TextBox).Name == "p\_textBox3")

 {

 if (char.IsDigit(item) || item == ',')

 continue;

 else

 text = text.Remove(text.IndexOf(item), 1);

 }

 else

 {

 if (char.IsDigit(item))

 continue;

 else

 text = text.Remove(text.IndexOf(item), 1);

 }

 }

 (sender as TextBox).Text = text;

 (sender as TextBox).Select(text.Length, 0);

 }

 //выбор товара из списка

 private void products\_box\_SelectedIndexChanged(object sender, EventArgs e)

 {

 //влючаем выключаем кнопки управления//

 if (products\_box.SelectedIndex!=0)

 {

 prev\_product.Enabled = true; first\_product.Enabled = true;

 }

 else

 {

 prev\_product.Enabled = false; first\_product.Enabled = false;

 }

 if (products\_box.SelectedIndex != products\_box.Items.Count-1)

 {

 next\_product.Enabled = true; last\_product.Enabled = true;

 }

 else

 {

 next\_product.Enabled = false; last\_product.Enabled = false;

 }

 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

 //получаем выбранные данные

 string[] data = products\_array[products\_box.SelectedIndex];

 Control[] boxes = new Control[] {

 p\_textBox1,p\_textBox2,p\_textBox3,p\_textBox4,p\_textBox6,p\_textBox5

 };

 for (int i = 0; i < boxes.Length; i++)

 {

 boxes[i].Text = data[i + 1];

 }

 foreach (string[] item in contractors\_id\_name)

 {

 if (item[0] == data[7])

 p\_comboBox.SelectedIndex = contractors\_id\_name.IndexOf(item);

 }

 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

 }

 //обновить запись

 private void update\_product\_Click(object sender, EventArgs e)

 {

 string[] data = products\_array[products\_box.SelectedIndex];

 update\_product\_data(Convert.ToInt32(data[0]));

 }

 //удалить запись

 private void delete\_product\_Click(object sender, EventArgs e)

 {

 string[] data = products\_array[products\_box.SelectedIndex];

 string query = $"DELETE Товар.\*, Товар.ID FROM Товар WHERE(((Товар.ID) = {data[0]}));";

 new OleDbCommand(query, db).ExecuteNonQuery();

 get\_products\_data(products\_box.SelectedIndex-1);

 }

 //новая запись

 private void add\_product\_Click(object sender, EventArgs e)

 {

 string query = "INSERT INTO Товар ( [ID поставщика] ) VALUES (1);";

 new OleDbCommand(query, db).ExecuteNonQuery();

 get\_products\_data(products\_box.Items.Count);

 }

 /\*\*\* СОБЫТИЯ ПОСТАВЩИКИ \*\*\*/

 //текст меняется contractor

 private void delivery\_days\_TextChanged(object sender, EventArgs e)

 {

 string text = (sender as TextBox).Text;

 foreach (char item in text)

 {

 if (char.IsDigit(item))

 continue;

 else

 text = text.Remove(text.IndexOf(item), 1);

 }

 (sender as TextBox).Text = text;

 (sender as TextBox).Select(text.Length, 0);

 }

 private void contractor\_box\_SelectedIndexChanged(object sender, EventArgs e)

 {

 //влючаем выключаем кнопки управления//

 if (contractor\_box.SelectedIndex != 0)

 {

 prev\_contractor.Enabled = true; first\_contractor.Enabled = true;

 }

 else

 {

 prev\_contractor.Enabled = false; first\_contractor.Enabled = false;

 }

 if (contractor\_box.SelectedIndex != contractor\_box.Items.Count - 1)

 {

 next\_contractor.Enabled = true; last\_contractor.Enabled = true;

 }

 else

 {

 next\_contractor.Enabled = false; last\_contractor.Enabled = false;

 }

 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

 //получаем выбранные данные

 string[] data = contractors\_array[contractor\_box.SelectedIndex];

 Control[] boxes = new Control[] {

 c\_textBox1,c\_textBox3,c\_textBox4,c\_textBox2

 };

 for (int i = 0; i < boxes.Length; i++)

 {

 boxes[i].Text = data[i + 1];

 }

 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

 }

 //обновить запись

 private void update\_contractor\_Click(object sender, EventArgs e)

 {

 string[] data = contractors\_array[contractor\_box.SelectedIndex];

 update\_contractors\_data(Convert.ToInt32(data[0]));

 }

 //удалить запись

 private void delete\_contractor\_Click(object sender, EventArgs e)

 {

 string[] data = contractors\_array[contractor\_box.SelectedIndex];

 string query = $"DELETE Поставщик.\*, Поставщик.ID FROM Поставщик WHERE(((Поставщик.ID) = {data[0]}));";

 new OleDbCommand(query, db).ExecuteNonQuery();

 get\_contractors\_data(contractor\_box.SelectedIndex - 1);

 }

 //новая запись

 private void add\_contractor\_Click(object sender, EventArgs e)

 {

 string query = "INSERT INTO Поставщик ( [Название поставщика] ) VALUES(\"\");";

 new OleDbCommand(query, db).ExecuteNonQuery();

 get\_contractors\_data(contractor\_box.Items.Count);

 }

 /\*\*\* СОБЫТИЯ ПОЛЬЗОВАТЕЛИ \*\*\*/

 //изменить пароль выбранного пользователя

 private void change\_user\_password\_Click(object sender, EventArgs e)

 {

 string[] data = users\_array[users\_list.SelectedIndex];

 ChangePassword change = new ChangePassword(db, Convert.ToInt32(data[0]), true);

 change.ShowDialog();

 }

 //выбор пользователя

 private void users\_list\_SelectedIndexChanged(object sender, EventArgs e)

 {

 if (users\_list.SelectedIndex != -1)

 users\_list.ContextMenuStrip = users\_menu;

 else

 users\_list.ContextMenuStrip = null;

 }

 /\*\*\* СОБЫТИЯ ФОРМЫ \*\*\*/

 //кнопки управления списками

 private void manage\_buttons\_Click(object sender, EventArgs e)

 {

 switch ((sender as Button).Name)

 {

 case "first\_product":

 products\_box.SelectedIndex = 0;

 break;

 case "prev\_product":

 products\_box.SelectedIndex = products\_box.SelectedIndex - 1;

 break;

 case "next\_product":

 products\_box.SelectedIndex = products\_box.SelectedIndex + 1;

 break;

 case "last\_product":

 products\_box.SelectedIndex = products\_box.Items.Count - 1;

 break;

 case "first\_contractor":

 contractor\_box.SelectedIndex = 0;

 break;

 case "prev\_contractor":

 contractor\_box.SelectedIndex = contractor\_box.SelectedIndex - 1;

 break;

 case "next\_contractor":

 contractor\_box.SelectedIndex = contractor\_box.SelectedIndex + 1;

 break;

 case "last\_contractor":

 contractor\_box.SelectedIndex = contractor\_box.Items.Count - 1;

 break;

 }

 }

 //переключение вкладок формы

 private void admin\_tabs\_SelectedIndexChanged(object sender, EventArgs e)

 {

 switch (admin\_tabs.SelectedIndex)

 {

 case 0:

 {

 get\_products\_data();

 Size = new Size(837, 207);

 }

 break;

 case 1:

 {

 get\_contractors\_data();

 Size = new Size(837, 192);

 }

 break;

 case 2:

 {

 get\_users\_data();

 Size = new Size(490, 280);

 }

 break;

 case 3:

 {

 get\_orders\_data();

 Size = new Size(837, 207);

 }

 break;

 }

 }

 //закрытие формы

 private void AdminForm\_FormClosing(object sender, FormClosingEventArgs e)

 {

 db.Close();

 Environment.Exit(0);

 }

 }

}

На рисунках 16 – 19 представлены скриншоты приложения учетной записи «Администратор»:

Рисунок 17 - Окно поставщиков в учётной записи Администратор

Рисунок 16 - Окно товаров в учётной записи Администратор



Рисунок 18 - Окно поставщиков в учётной записи Администратор



Рисунок 19 - Окно поставщиков в учётной записи Администратор

# ЗАКЛЮЧЕНИЕ

За последние годы в мировой экономике все большую значимость приобретают такие изменения, как глобализация и интеграция. Эти изменения приводят к росту объемов мировой торговли, что в свою очередь ведет к увеличению спроса на логистические услуги (доставку товаров, обработку и другие операции). Проведенные исследования показали, что организация системы доставки товаров является сложной и многовариантной задачей. При решении данной задачи необходимо осуществить комплекс мероприятий, связанных с организацией системы доставки посылок, по анализу и выбору логистических посредников. При этом необходимо учесть, что основным критерием для потребителя в настоящие время является соблюдение сроков поставки (точно во время) и минимизация издержек.

Цель курсовой работы достигнута. Выполнены следующие задачи:

1. Изучены технологии проектирования информационных систем.
2. Проектирование информационной системы предприятия.
3. Созданы базы данных для приложения.
4. Разработан и описан интерфейс приложения.
5. Реализовано подключение БД к среде разработки Visual Studio.
6. Реализовано приложение в среде разработки Visual Studio на языке С#.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вичугова А.А. Инструментальные средства информационных систем, -М.: Вильямс 2015. – 136 с.
2. Мещеряков, С. В. Эффективные технологии создания информационных систем / С.В. Мещеряков, В.М. Иванов. - М.: Политехника, 2015. - 368 c.
3. MS Access и C#: работаем с базой данных из программы Windows Forms // VScode.ru программирование на языках C, C# и Java.
URL: <https://vscode.ru/prog-lessons/ms-access-i-c-sharp-rabotaem-s-bd.html> (дата обращения: 19.03.2020).
4. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс, 2015. - 1184 c.
5. Гвоздева, В. А. Введение в специальность программиста / В.А. Гвоздева. - М.: Форум, Инфра-М, 2015. - 208 c.
6. Абрамов, С.А. Математические построения и программирование / С.А. Абрамов. - М.: Наука, 2016. - 192 c.
7. Ван, Тассел Д. Стиль, разработка, эффективность, отладка и испытания программ / Ван Тассел Д. - М.: Мир, 2017. - 332 c.
8. Информационные технологии на автомобильном транспорте: Учебник / Под ред. Власова В.М. - М.: Academia, 2017. - 320 c.
9. Информационные технологии и вычислительные системы: Обработка информации и анализ данных. Программная инженерия. Математическое моделирование. Прикладные аспекты информатики / Под ред. С.В. Емельянова. - М.: Ленанд, 2015. - 104 c.
10. Гаврилов, М.В. Информатика и информационные технологии: Учебник / М.В. Гаврилов, В.А. Климов. - Люберцы: Юрайт, 2016. - 383 c.
11. Информационные технологии: учебное пособие / А.А. Вичугова, В.Н. Вичугов, Е.А. Дмитриева, Г.П. Цапко. - Томск: Изд-во ТПУ, 2012. - 105 с.
12. Аверченков В.И. Аудит информационной безопасности: учебное пособие для вузов. - М.: ФЛИНТА, 2011. - 269 с.
13. Оклей, П. И. Инструментальные средства и математические модели прогнозирования остаточного ресурса по фактически измеряемым параметрам оборудования электростанций / П.И. Оклей. - М.: Ленанд, 2015. - 160 c.
14. Александров, Д. В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы / Д.В. Александров. - М.: Финансы и статистика, 2016. - 224 c.
15. Оклей, П.И. Инструментальные средства и математические модели прогнозирования остаточного ресурса по ф / П.И. Оклей. - М.: Ленанд, 2015. - 483 c.
16. Горнаков, Станислав Инструментальные средства программирования и отладки шейдеров в DirectX и OpenGL: моногр. / Станислав Горнаков. - М.: БХВ-Петербург, 2016. - 263 c.