МИНОБРНАУКИ РФ

ФГБОУ ВПО Тверской государственный технический университет

Кафедра “Программное обеспечение”

Курсовая работа

дисциплина “Базы данных”

Тема: “ Тема: “ Разработка системы автоматизации с использованием СУБД: магазин ”

Выполнил:

студент группы

ПИН 17.05

Иванов Роман Вячеславович

Проверил:

Артемов И.Ю

Тверь 2019

Содержание.

Оглавление

[Постановка задачи 3](#_Toc8138216)

[Введение. 3](#_Toc8138217)

[Проектная часть. 6](#_Toc8138218)

[Работа с MySql Open server 6](#_Toc8138219)

[Функционал. 7](#_Toc8138220)

[Схема данных. 10](#_Toc8138221)

[Диаграмма классов. 11](#_Toc8138222)

[Тестирование 11](#_Toc8138223)

[Вывод. 15](#_Toc8138224)

[Литература. 15](#_Toc8138225)

# Постановка задачи

Разработка системы автоматизации с использованием СУБД (MySQL/Firebird/PosgreSQL/MSSQL): магазин.

В результате выполнения курсовой работы должно быть разработано программное обеспечение  (в виде исходных кодов) и оформлен отчет о проделанной работе (в печатном и электронном виде).

Исходный код программного обеспечения должен:

* Разрабатывать на языке SQL;
* Содержать полную (корректную) обработку ошибок;
* Выполнять все поставленные задачи

# Введение.

Сегодня большое значение имеет работа с данными. Для хранения данных используются различные системы управления базами данных: MS SQL Server, Oracle, MySQL и так далее. И большинство крупных приложений так или иначе используют для хранения данных эти системы управления базами данных. Однако чтобы осуществлять связь между базой данных и приложением на C# необходим посредник. И именно таким посредником является технология ADO.NET.

ADO.NET предоставляет собой технологию работы с данными, которая основана на платформе .NET Framework. Эта технология представляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Причем важно отметить, что систем управления баз данных может быть множество. В своей сущности они могут различаться. MS SQL Server, например, для создания запросов использует язык T-SQL, а MySQL и Oracle применяют язык PL-SQL. Разные системы баз данных могут иметь разные типы данных. Также могут различаться какие-то другие моменты. Однако функционал ADO.NET построен таким образом, чтобы предоставить разработчикам унифицированный интерфейс для работы с самыми различными СУБД.

Основу интерфейса взаимодействия с базами данных в ADO.NET представляет ограниченный круг объектов: Connection, Command, DataReader, DataSet и DataAdapter. С помощью объекта **Connection** происходит установка подключения к источнику данных. Объект **Command** позволяет выполнять операции с данными из БД. Объект **DataReader** считывает полученные в результате запроса данные. Объект **DataSet** предназначен для хранения данных из БД и позволяет работать с ними независимо от БД. И объект **DataAdapter** является посредником между DataSet и источником данных. Главным образом, через эти объекты и будет идти работа с базой данных.

Однако чтобы использовать один и тот же набор объектов для разных источников данных, необходим соответствующий **провайдер данных**. Собственно через провайдер данных в ADO.NET и осуществляется взаимодействие с базой данных. Причем для каждого источника данных в ADO.NET может быть свой провайдер, который собственно и определяет конкретную реализацию вышеуказанных классов.

По умолчанию в ADO.NET имеются следующие встроенные провайдеры:

* Провайдер для MS SQL Server
* Провайдер для OLE DB (Предоставляет доступ к некоторым старым версиям MS SQL Server, а также к БД Access, DB2, MySQL и Oracle)
* Провайдер для ODBC (Провайдер для тех источников данных, для которых нет своих провайдеров)
* Провайдер для Oracle
* Провайдер EntityClient. Провайдер данных для технологии ORM Entity Framework
* Провайдер для сервера SQL Server Compact 4.0

Кроме этих провайдеров, которые являются встроенными, существует также множество других, предназначенных для различных баз данных, например, для MySQL.

Основные пространства имен, которые используются в ADO.NET:

* System.Data: определяет классы, интерфейсы, делегаты, которые реализуют архитектуру ADO.NET
* System.Data.Common: содержит классы, общие для всех провайдеров ADO.NET
* System.Data.Design: определяет классы, которые используются для создания своих собственных наборов данных
* System.Data.Odbc: определяет функциональность провайдера данных для ODBC
* System.Data.OleDb: определяет функциональность провайдера данных для OLE DB
* System.Data.Sql: хранит классы, которые поддерживают специфичную для SQL Server функциональность
* System.Data.OracleClient: определяет функциональность провайдера для баз данных Oracle
* System.Data.SqlClient: определяет функциональность провайдера для баз данных MS SQL Server
* System.Data.SqlServerCe: определяет функциональность провайдера для SQL Server Compact 4.0
* System.Data.SqlTypes: содержит классы для типов данных MS SQL Servera
* Microsoft.SqlServer.Server: хранит компоненты для взаимодействия SQL Server и среды CLR

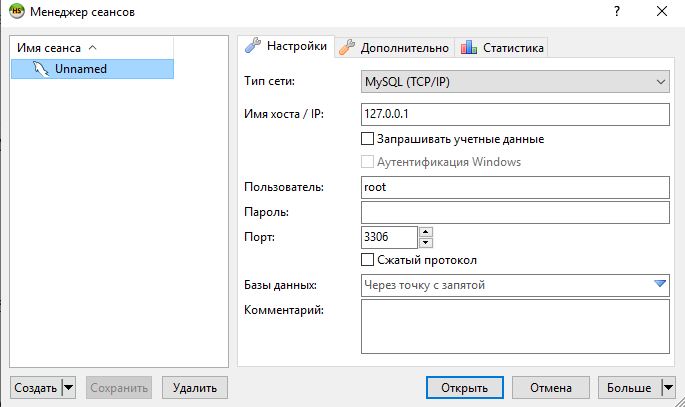
Схематично архитектуру ADO.NET можно представить следующим образом:

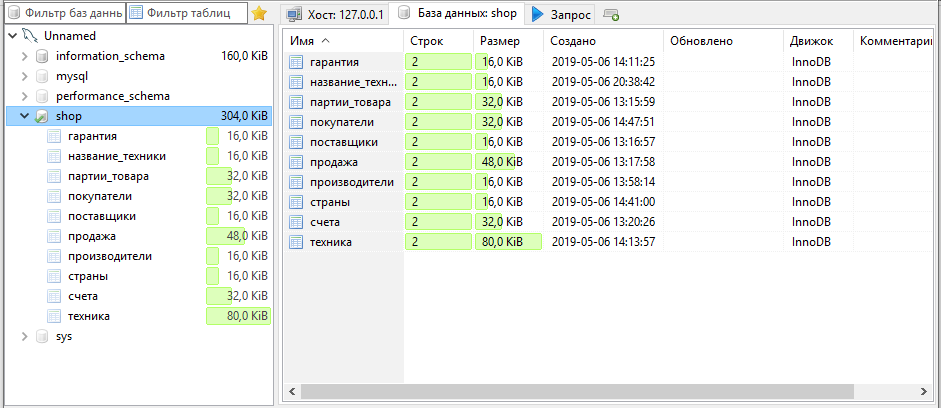
Функционально классы ADO.NET можно разбить на два уровня: подключенный и отключенный. Каждый провайдер данных .NET реализует свои версии объектов Connection, Command, DataReader, DataAdapter и ряда других, который составляют подключенный уровень. То есть с помощью них устанавливается подключение к БД и выполняется с ней взаимодействие. Как правило, реализации этих объектов для каждого конкретного провайдера в своем названии имеют префикс, который указывает на провайдер:

Другие классы, такие как DataSet, DataTable, DataRow, DataColumn и ряд других составляют отключенный уровень, так как после извлечения данных в DataSet мы можем работать с этими данными независимо от того, установлено ли подключение или нет. То есть после получения данных из БД приложение может быть отключено от источника данных.

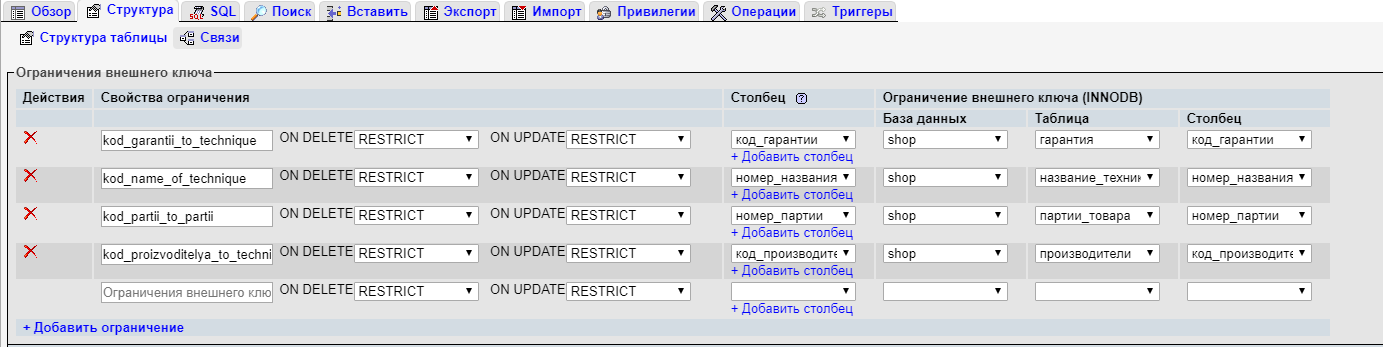
# Проектная часть.

## Работа с MySql Open server

На локальном сервере была создана база данных «Магазин» состоящая из 10 таблиц. 



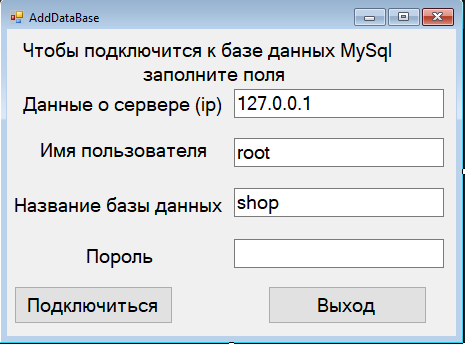
Все таблицы связаны между собой.



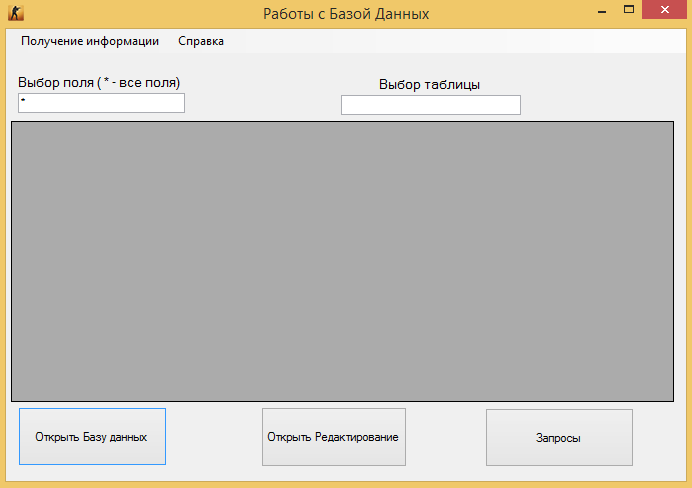
## Функционал приложения.

Вход в базу данных

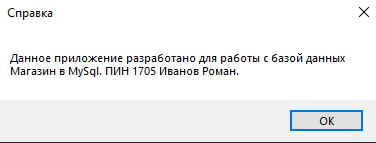
Пользователю предлагается ввести данные доступа для получения соединения с базой данных.



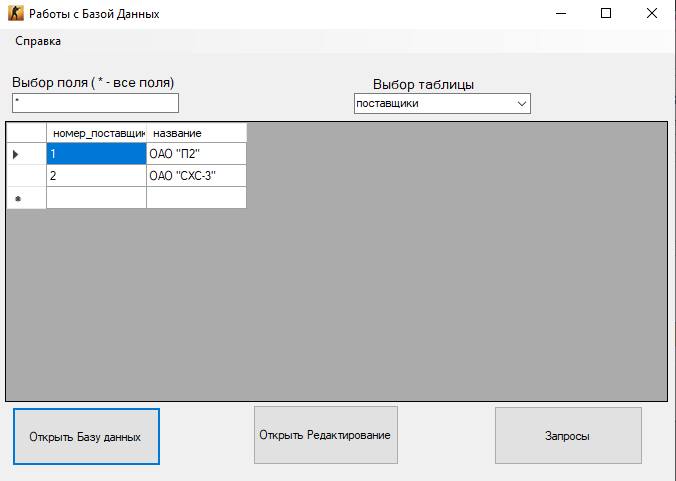
На данной картинке представлено основное окно для работы с базой данных.

Для того, что бы узнать какие поля и таблицы находятся в БД, нажимаем на «Получение информации». Открывается данное окно, где нам предлагается выбрать из «Списка таблиц» таблицу и получить данные по ней

Мы можем получить информацию о БД, нажав на кнопку «Справка»

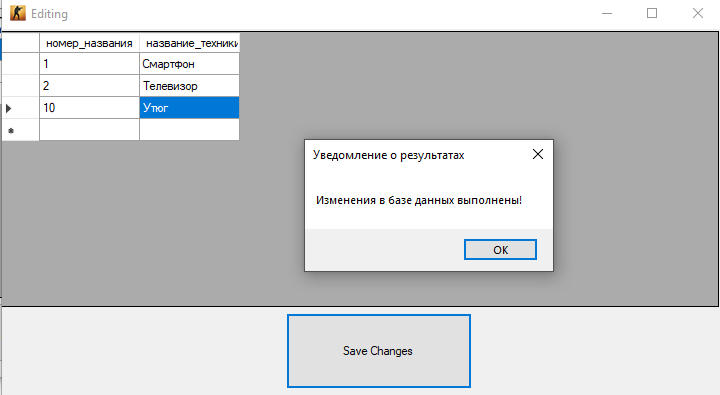


Чтобы открыть БД нужно в поле «Выбор таблицы» ввести желаемую таблицу и нажать на кнопку «Открыть БД» (Названия таблиц находятся в меню>получение информации).



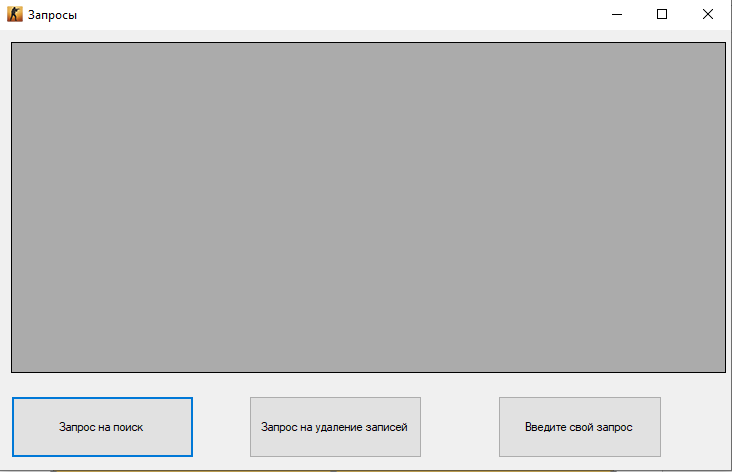
Что бы редактировать БД нажимаем на кнопку «Открыть Редактирование»

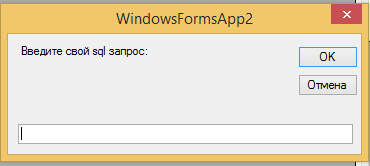
В открывшемся окне мы можем удалять, добавлять и менять запись.



Запросы, при нажатии на кнопку «Запросы», открывается данное окно.

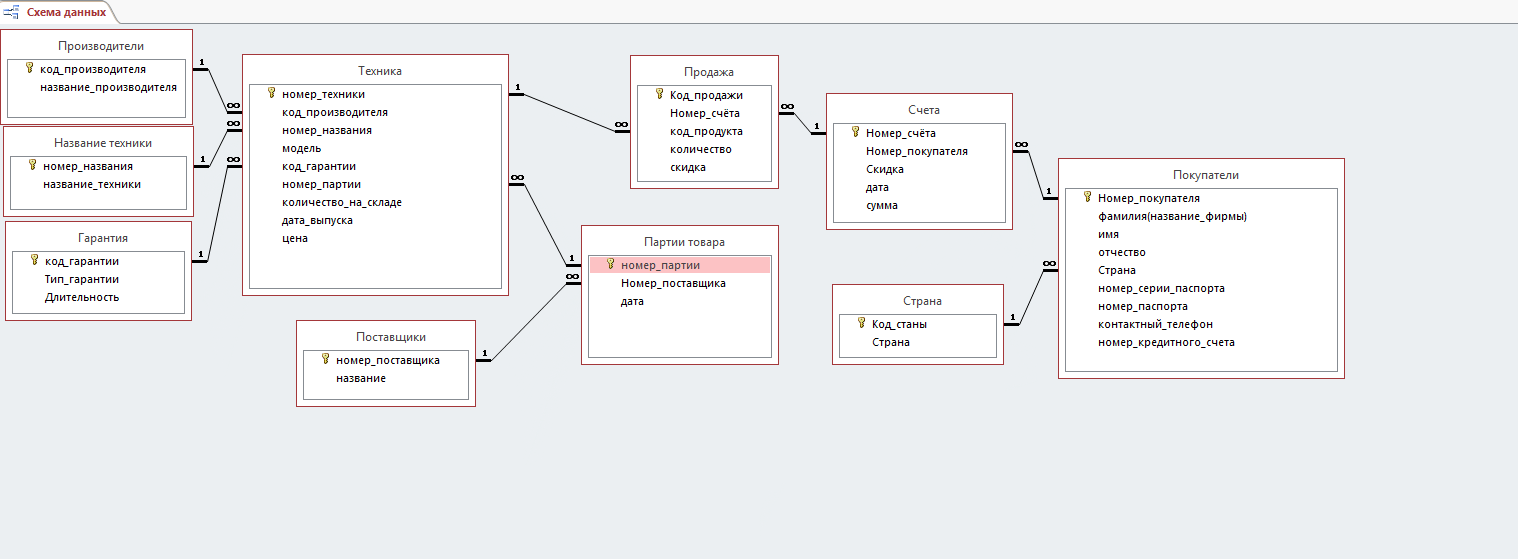
Вы можете воспользоваться предложенными запросами или ввести свой запрос.





## Схема данных.

На данной картинке представлена Схема данных нашей БД

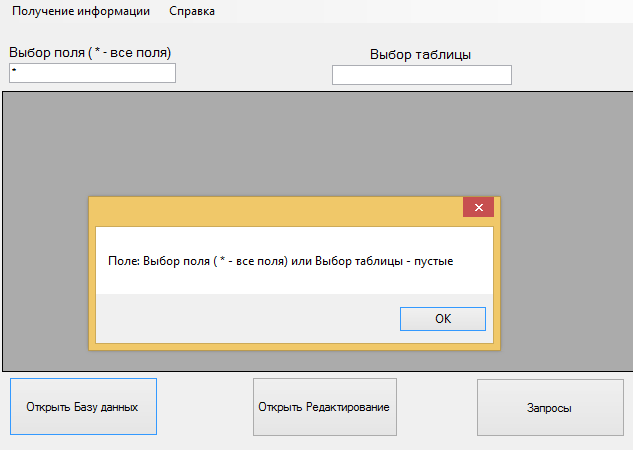


## Тестирование

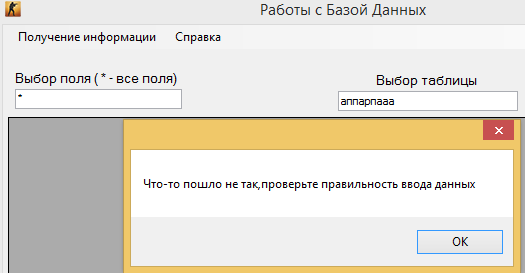
Приведу пример на 2х формах, так как остальные формы имею такую же проверку на ошибки, как и все оставшиеся.

Главная Форма

Если нажати на кнопку «Открыть БД» Поле: Выбор поля ( \* - все поля) или Выбор таблицы - пустые то программа выдает на такое окно



Если в написании названия таблицы или поля была допущена ошибка, то программа выдает нам это.

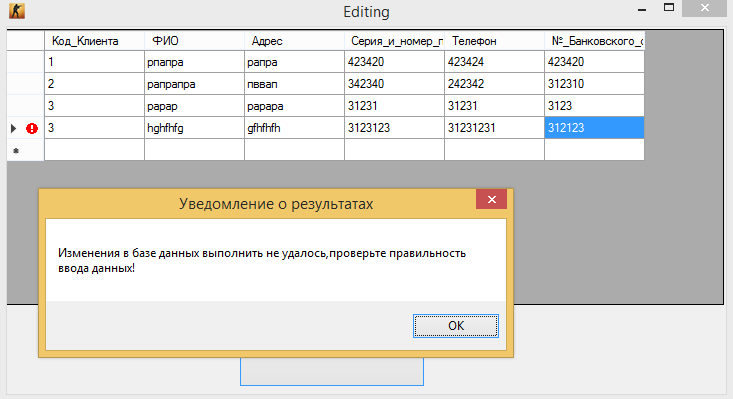


Код:

|  |
| --- |
| private void OpenBd() |
|  | { |
|  |  |
|  | if (filds.Text != "" && tables.Text != "") |
|  | { |
|  | try |
|  | { |
|  | OpenConnection(); |
|  | string filds1 = filds.Text; |
|  | string tables1 = tables.Text; |
|  | string query = String.Format("SELECT {0} From {1}", filds1, tables1);//SQL Запрос |
|  | MySqlCommand command = new MySqlCommand(query, myConnection); |
|  |  |
|  | //Вывод данных в DataGridView |
|  | MySqlDataAdapter MyDA = new MySqlDataAdapter(); |
|  | MyDA.SelectCommand = command; |
|  | DataTable dataSet = new DataTable(); |
|  | MyDA.SelectCommand = command; |
|  | MyDA.Fill(dataSet); |
|  | dataGridView1.DataSource = dataSet; |
|  | MyDA.Update(dataSet); |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  | CloseConnection(); |
|  | } |
|  |  |
|  | catch (Exception) |
|  | { |
|  | MessageBox.Show("Что-то пошло не так,проверьте правильность ввода данных"); |
|  | CloseConnection(); |
|  | } |
|  | } |
|  | else |
|  | { |
|  | MessageBox.Show("Поле: Выбор поля ( \* - все поля) или Выбор таблицы - пустые "); |
|  | } |
|  | } |

Форма «Редактирование»

При вводе в форму неправильных данных, например ключевого поля с таким с похожим обозначением или ввода таблицу, в которой доступны только цифры, букв программа выдает нам такое предупреждение



Код:

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e) |
|  | { | |
|  | try |
|  | { |
|  | adap.Update(ds.Tables[0]); | |
|  | MessageBox.Show("Изменения в базе данных выполнены!", | |
|  | "Уведомление о результатах", MessageBoxButtons.OK); | |
|  | } | |
|  | catch (Exception) | |
|  | { | |
|  | MessageBox.Show("Изменения в базе данных выполнить не удалось,проверьте правильность ввода данных! ", | |
|  | "Уведомление о результатах", MessageBoxButtons.OK); | |
|  | } | |
|  | } | |
|  | MySqlDataAdapter adap = null; | |
|  | DataSet ds = new DataSet(); | |
| private void Editing\_Load(object sender, EventArgs e) |  | |
|  | { | |
|  | try | |
|  | { | |
|  | myConnection.Open(); | |
|  | string filds1 = filds.Text; | |
|  | string tables1 = tables.Text; | |
|  | string query = String.Format("SELECT {0} From {1}", filds1, tables1);//SQL Запрос | |
|  |  | |
|  | //Вывод данных в DataGridView | |
|  | adap = new MySqlDataAdapter(query, myConnection); | |
|  | MySqlCommandBuilder bulder = new MySqlCommandBuilder(adap); | |
|  | adap.UpdateCommand = bulder.GetUpdateCommand(); | |
|  | adap.InsertCommand = bulder.GetInsertCommand(); | |
|  | adap.DeleteCommand = bulder.GetDeleteCommand(); | |
|  | adap.Fill(ds); | |
|  | dataGridView1.DataSource = ds.Tables[0]; | |
|  | } | |
|  | catch (Exception) | |
|  | { | |
|  | MessageBox.Show("Что-то пошло не так,проверьте правильность ввода данных"); | |
|  | this.Close(); | |
|  | } | |
|  | } | |
|  | } | |

## Вывод.

Таким образом, в ходе практической части выполнены следующие этапы: описание предметной области, проектирование программного комплекса, программная реализация и анализ качества программного обеспечения.

# Литература.

РАБОТА С БАЗАМИ ДАННЫХ НА ЯЗЫКЕ C# ТЕХНОЛОГИЯ ADO .NET -Составители О. Н. Евсеева А. Б. Шамшев.