Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет   
имени первого Президента России Б. Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ

**Отчёт по Лабораторной работе №1**

по дисциплине «Средства управления информационными ресурсами АС»

**Тема:** Создание Windows-приложения мастерами Visual Studio

Научный руководитель Парфенов Ю. П.

Курс, группа РИЗ-300016у

Студент Кулаков М. И.

Екатеринбург, 2022

# СОДЕРЖАНИЕ

1 Постановка задачи и вариант задания 3

2 Скрипт создания базы данных 4

3 Образы экранных форм в конструкторе и исполняемой программе с описанием используемых компонент и мастеров. 14

4 Описание навигационной панели и её использования для управления данными 19

# 1 Постановка задачи и вариант задания

1. Постройте или восстановите из предыдущего семестра базу, соответствующую варианту вашего индивидуального задания. Варианты индивидуальных заданий приведены в приложении.

2. Наполните таблицы 3–4 строками данными, соответствующими заданным ограничениям.

3. Создайте новый проект приложения Windows в Visual Studio.

4. Предусмотрите в форме пять вкладок по числу выполняемых лабораторных работ.

5. На первой вкладке «Лабораторная работа №1» разместите компоненты, необходимые для просмотра и управления данными из основной таблицы индивидуального задания.

6. Проверьте работу приложения добавлением, удалением и изменением данных в основной таблице.

7. Изучите классы, созданные конструктором экранной формы на основе структуры используемой таблицы БД.

Вариант №10:

Типовые объекты предметной области (сущности): Оборудование - единица оборудования, находящаяся на материальном учёте (монитор, системный блок, принтер, КИП и т. д.). Ответственный пользователь – сотрудник, несущий ответственность за сохранность оборудования. Требуется:

* связь «Оборудование» ↔ «Ответственный пользователь»: в разные периоды оборудование предаётся разным пользователям. Атрибуты:
  + дата получения;
  + дата возврата;
  + цель использования;
  + причина возврата. Справочник: поломка оборудования, увольнение сотрудника и т. д.
* сущность «Оборудование». Атрибуты:
  + инвентарный номер;
  + серийный номер;
  + название;
  + дата постановки на учёт;
  + стоимость;
  + адреса гарантийного сервиса;
  + поставщик (хранится как отдельная сущность), справочник;
* сущность «Ответственный пользователь». Атрибуты:
  + табельный номер;
  + ФИО;
  + место работы:
  + цех;
  + номер участка;
  + отдел;
  + номер комнаты.

# 2 Скрипт создания базы данных

*------------------------------------------------------------  
------------------------- Контрольная работа по Базам данных  
------------------------- Студента группы РИЗ-300016у  
------------------------- Кулакова Максима Ивановича  
------------------------------------------------------------  
-- Вариант №10 «Учёт оборудования»  
------------------------------------------------------------  
  
  
-- Создание базы данных***USE master**;  
**GO  
  
DROP DATABASE IF EXISTS** lessonsUrFU;  
**GO  
  
CREATE DATABASE** lessonsUrFU  
 **ON PRIMARY** (  
 **NAME** = lessonUrFU\_main\_db,  
 **FILENAME** = **N'/var/opt/mssql/data/lessonUrFU\_main.mdf'**,  
 **SIZE** = 10 **MB**,  
 **MAXSIZE** = 15 **MB**,  
 **FILEGROWTH** = 10%),  
 **FILEGROUP SECONDARY** (  
 **NAME** = lessonUrFU\_secondary\_db,  
 **FILENAME** = **N'/var/opt/mssql/data/lessonUrFU\_secondary.ndf'**,  
 **SIZE** = 5 **MB**,  
 **MAXSIZE** = 10 **MB**,  
 **FILEGROWTH** = 10%)  
 **LOG ON** (  
 **NAME** = lessonUrFU\_db\_log,  
 **FILENAME** = **N'/var/opt/mssql/data/lessonUrFU.ldf'**,  
 **SIZE** = 1 **MB**,  
 **MAXSIZE** = 5 **MB**,  
 **FILEGROWTH** = 10%  
 );  
**GO***------------------------------------------------------------  
-- Создание таблиц:  
-- 1. Таблица-справочник «Поставщики оборудования»  
-- 2. Таблица-справочник «Причина возврата»  
-- 3. Таблица «Ответственне»  
-- 4. Таблица «Оборудование»  
-- 5. Таблица для связей «Возврат оборудования»  
------------------------------------------------------------***USE** lessonsUrFU;  
**GO***-- Таблица-справочник «Поставщики оборудования»***CREATE TABLE** Suppliers  
(  
 **Id INT IDENTITY**, *-- Идентификатор поставщика* **SupplierName NVARCHAR**(50) **NOT NULL**, *-- Наименование поставщика* **CONSTRAINT** PK\_Suppliers **PRIMARY KEY** (**Id**),  
 **CONSTRAINT** UQ\_Suppliers\_SuppliersName **UNIQUE** (**SupplierName**)  
) **ON** [SECONDARY];  
  
*-- Таблица-справочник «Причина возврата»***CREATE TABLE** ReasonsForReturn  
(  
 **Id INT IDENTITY**, *-- Идентификатор причины возврата* **ReasonName NVARCHAR**(50) **NOT NULL**, *-- Наименование причины возврата* **CONSTRAINT** PK\_ReasonsForReturn **PRIMARY KEY** (**Id**),  
 **CONSTRAINT** UQ\_ReasonsForReturn\_ReasonName **UNIQUE** (**ReasonName**)  
) **ON** [SECONDARY];  
**GO***-- Таблица «Оборудование»***CREATE TABLE** Equipment  
(  
 **InventoryID INT IDENTITY**, *-- Инвентарный номер оборудования* **SerialNumber INT NOT NULL**, *-- Серийный номер* **EquipmentName NVARCHAR**(60) **NOT NULL**, *-- Наименование* **RegistrationDate DATE NOT NULL** *-- Дата регистрации* **CONSTRAINT** DF\_Equipment\_RegistrationDate **DEFAULT** *GETDATE*(),  
 **Price MONEY NULL**, *-- Стоимость* **WarrantyServiceAddress NVARCHAR**(100) **NULL**, *-- Адреса гарантийного обслуживания* **Supplier INT NOT NULL**, *-- Поставщик* **CONSTRAINT** PK\_Equipment **PRIMARY KEY** (**InventoryID**),  
 **CONSTRAINT** UQ\_Equipment\_SerialNumber **UNIQUE** (**SerialNumber**),  
 **CONSTRAINT** FK\_Equipment\_Suppliers **FOREIGN KEY** (**Supplier**)  
 **REFERENCES** Suppliers (**Id**)  
 **ON UPDATE CASCADE**) **ON** [PRIMARY];  
**GO***-- Таблица «Ответственные»***CREATE TABLE** Responsible  
(  
 **PersonnelNumber INT IDENTITY**, *-- Табельный номер сотрудника* **Surname NVARCHAR**(30) **NOT NULL**, *-- Фамилия* **Name NVARCHAR**(30) **NOT NULL**, *-- Имя* **MiddleName NVARCHAR**(50) **NULL**, *-- Отчество* **Workshop NVARCHAR**(35) **NOT NULL**, *-- Цех* **LotNumber INT NOT NULL**, *-- Номер участка* **Department NVARCHAR**(35) **NOT NULL**, *-- Отдел* **RoomNumber INT NOT NULL**, *-- Номер комнаты* **Telephone NVARCHAR**(16) **NOT NULL**, *-- Телефон* **CONSTRAINT** PK\_Responsible **PRIMARY KEY** (**PersonnelNumber**),  
 **CONSTRAINT** CH\_Responsible\_LotNumber **CHECK** (**LotNumber** >= 0),  
 **CONSTRAINT** CH\_Responsible\_RoomNumber **CHECK** (**RoomNumber** > 99),  
 **CONSTRAINT** CH\_Responsible\_Telephone **CHECK** (**Telephone LIKE  
 '+[0-9] [0-9][0-9][0-9] [0-9][0-9][0-9] [0-9][0-9]-[0-9][0-9]'**)  
) **ON** [PRIMARY];  
**GO***-- Таблица для связей «Ответственность за оборудование»***CREATE TABLE** Responsibility  
(  
 **Id INT IDENTITY**, *-- Уникальный номер записи* **Responsible INT NOT NULL**, *-- Табельный номер ответственного* **Equipment INT NOT NULL**, *-- Инвентарный ID оборудования* **DateOfReceiving DATE NOT NULL** *-- Дата получения* **CONSTRAINT** DF\_Responsibility\_DateOfReceiving **DEFAULT** *GETDATE*(),  
 **ReturnDate DATE NULL** *-- Дата возврата* **CONSTRAINT** DF\_Responsibility\_ReturnDate **DEFAULT NULL**,  
 **PurposeOfUse NVARCHAR**(50) **NULL**, *-- Цель использования* **ReasonForReturn INT NULL**, *-- Причина возврата* **CONSTRAINT** PK\_Responsibility **PRIMARY KEY** (**Id**),  
 **CONSTRAINT** FK\_Responsibility\_Responsible **FOREIGN KEY** (**Responsible**)  
 **REFERENCES** Responsible (**PersonnelNumber**)  
 **ON DELETE CASCADE  
 ON UPDATE CASCADE**,  
 **CONSTRAINT** FK\_Responsibility\_Equipment **FOREIGN KEY** (**Equipment**)  
 **REFERENCES** Equipment (**InventoryId**)  
 **ON DELETE CASCADE  
 ON UPDATE CASCADE**,  
 **CONSTRAINT** CH\_Responsibility\_ReturnDate **CHECK** (**ReturnDate** >= **DateOfReceiving**),  
 **CONSTRAINT** FK\_Responsibility\_ReasonForReturn **FOREIGN KEY** (**ReasonForReturn**)  
 **REFERENCES** ReasonsForReturn (**Id**)  
 **ON DELETE SET NULL  
 ON UPDATE CASCADE**,  
 **CONSTRAINT** CH\_Responsibility\_ReasonForReturn **CHECK** ((**ReasonForReturn IS NOT NULL AND ReturnDate IS NOT NULL**) **OR  
 ReasonForReturn IS NULL**)  
) **ON** [SECONDARY];  
**GO***------------------------------------------------------------  
-- Создание представлений  
------------------------------------------------------------***CREATE VIEW** vw\_Responsibility **AS  
SELECT** Rs.**Id**,  
 **R**.**Surname**,  
 **R**.**Name**,  
 **R**.**MiddleName**,  
 E.**SerialNumber**,  
 E.**EquipmentName**,  
 Rs.**DateOfReceiving**,  
 Rs.**ReturnDate**,  
 Rs.**PurposeOfUse**,  
 Rs.**ReasonForReturn**,  
 RF.**ReasonName  
FROM** Responsibility **AS** Rs  
 **LEFT OUTER JOIN** Responsible **AS R ON R**.**PersonnelNumber** = Rs.**Responsible  
 LEFT OUTER JOIN** Equipment **AS** E **ON** E.**InventoryID** = Rs.**Equipment  
 LEFT OUTER JOIN** ReasonsForReturn **AS** RF **ON** RF.**Id** = Rs.**ReasonForReturn**;  
**GO  
  
CREATE VIEW** vw\_Equipment **AS  
SELECT** E.**InventoryID**,  
 E.**SerialNumber**,  
 E.**EquipmentName**,  
 E.**RegistrationDate**,  
 E.**Price**,  
 E.**WarrantyServiceAddress**,  
 **S**.**SupplierName  
FROM** Equipment **AS** E  
 **INNER JOIN** Suppliers **AS S ON S**.**Id** = E.**Supplier**;  
**GO***------------------------------------------------------------  
-- Задание №1 Поиск записей об оборудовании  
-- находящегося у пользователя  
------------------------------------------------------------***CREATE PROCEDURE** *sp\_DisplayEquipmentByUser* @surname **NVARCHAR**(30),  
 @name **NVARCHAR**(30),  
 @middleName **NVARCHAR**(50) = **NULL**,  
 @date **DATETIME** = **NULL AS  
BEGIN  
 DECLARE** @isSurname **BIT**, @isName **BIT**, @isMiddleName **BIT**, @isReturnDate **BIT**, @fullNames **NVARCHAR**(110) = **N''**;  
  
 **IF** (@surname **IS NULL**)  
 *-- Имя не указано, выполняем пропуск проверки* **SET** @isSurname = 1;  
 **ELSE  
 BEGIN  
 SELECT** @fullNames = **Surname FROM** Responsible **WHERE Surname** = @surname;  
  
 **IF** (**SELECT** *COUNT*(*\**) **FROM** Responsible **WHERE Surname** = @surname) = 0  
 *-- Имя указано, но не найдено в таблице Responsible* **SET** @isSurname = 0;  
 **ELSE** *-- Имя указано и найдено в таблице Responsible* **SET** @isSurname = 1;  
 **END  
  
 IF** (@name **IS NULL**)  
 *-- Фамилия не указана, выполняем пропуск проверки* **SET** @isName = 1;  
 **ELSE  
 BEGIN  
 SELECT** @fullNames = @fullNames + **' '** + **Name FROM** Responsible **WHERE Name** = @name;  
  
 **IF** (**SELECT** *COUNT*(*\**) **FROM** Responsible **WHERE Name** = @name) = 0  
 *-- Фамилия указана, но не найдена в таблице Responsible* **SET** @isName = 0;  
 **ELSE** *-- Фамилия указана и найдена в таблице Responsible* **SET** @isName = 1;  
 **END  
  
 IF** (@middleName **IS NULL**)  
 *-- Отчество не указано, выполняем пропуск проверки* **SET** @isMiddleName = 1;  
 **ELSE  
 BEGIN  
 SELECT** @fullNames = @fullNames + **' '** + **MiddleName FROM** Responsible **WHERE MiddleName** = @middleName;  
  
 **IF** (**SELECT** *COUNT*(*\**) **FROM** Responsible **WHERE MiddleName** = @middleName) = 0  
 *-- Отчество указано, но не найдено в таблице Responsible* **SET** @isMiddleName = 0;  
 **ELSE** *-- Отчество указано и найдено в таблице Responsible* **SET** @isMiddleName = 1;  
 **END  
  
 IF** (@date **IS NULL OR** (**SELECT** *COUNT*(*\**) **FROM** Responsibility **WHERE** @date **BETWEEN DateOfReceiving AND ReturnDate**) > 0)  
 *-- Дата не указана или найдена в таблице Responsibility* **SET** @isReturnDate = 0;  
 **ELSE** *-- Дата найдена в таблице Responsibility* **SET** @isReturnDate = 1;  
  
 **DECLARE** @count **INT**;  
 **SELECT** @count = *COUNT*(*\**)  
 **FROM** Responsibility **AS** Rs  
 **INNER JOIN** Equipment **AS** E **ON** E.**InventoryID** = Rs.**Equipment  
 INNER JOIN** Responsible **AS R ON R**.**PersonnelNumber** = Rs.**Responsible  
 WHERE** (**R**.**Surname** = @surname **OR** @isSurname = 1)  
 **AND** (**R**.**Name** = @name **OR** @isName = 1)  
 **AND** (**R**.**MiddleName** = @middleName **OR** @isMiddleName = 1)  
 **AND** ((@date **BETWEEN** Rs.**DateOfReceiving AND** Rs.**ReturnDate**) **OR** @isReturnDate = 1);  
  
 **IF** @count = 1  
 **SELECT R**.**Surname** + **' '** + **R**.**Name AS N'ФИО ответственного'**,  
 E.**EquipmentName AS N'Наименование оборудования'**,  
 E.**InventoryID AS N'Инвентарный ID'**,  
 E.**SerialNumber AS N'Серийный номер'**,  
 E.**Price AS N'Цена'**,  
 E.**RegistrationDate AS N'Дата постановки на учет'**,  
 Rs.**DateOfReceiving AS N'Дата получения'**,  
 Rs.**ReturnDate AS N'Дата возврата'**,  
 Rs.**PurposeOfUse AS N'Цель использования'**,  
 Rs.**ReasonForReturn AS N'Причина возврата'  
 FROM** Responsibility **AS** Rs  
 **INNER JOIN** Equipment **AS** E **ON** E.**InventoryID** = Rs.**Equipment  
 INNER JOIN** Responsible **AS R ON R**.**PersonnelNumber** = Rs.**Responsible  
 WHERE** (**R**.**Surname** = @surname **OR** @isSurname = 1)  
 **AND** (**R**.**Name** = @name **OR** @isName = 1)  
 **AND** (**R**.**MiddleName** = @middleName **OR** @isMiddleName = 1)  
 **AND** ((@date **BETWEEN** Rs.**DateOfReceiving AND** Rs.**ReturnDate**) **OR** @isReturnDate = 1);  
 **ELSE  
 PRINT N'Количество пользователей с '** + @fullNames + **N' равно '** + *CONVERT*(**NVARCHAR**, @count) + **N'!'**;  
**END**;  
**GO***-- Вывод сведений об оборудовании у Кудрявцева Артура Романовича на момент 25.05.2020***EXEC** *sp\_DisplayEquipmentByUser* @surname = **N'Кудрявцев'**, @name = **N'Артур'**, @middleName = **N'Романович'**,  
 @date = **'2020-05-25'**;  
**GO***------------------------------------------------------------  
-- Задание №2 Отображение пользователей задерживающие  
-- оборудование дольше среднего  
------------------------------------------------------------***CREATE PROCEDURE** *sp\_DisplayUsersDelayingEquipment* **AS  
BEGIN  
 DECLARE** @avgDays **INT** = (**SELECT** *AVG*(*DATEDIFF*(**DAY**, **DateOfReceiving**, *COALESCE*(**ReturnDate**, *GETDATE*())))  
 **FROM** Responsibility);  
  
 **SELECT R**.**Surname AS N'Фамилия'**,  
 **R**.**Name AS N'Имя'**,  
 **R**.**MiddleName AS N'Отчество'**,  
 E.**EquipmentName AS N'Наименование оборудования'**,  
 *DATEDIFF*(**DAY**, Rs.**DateOfReceiving**, *COALESCE*(Rs.**ReturnDate**, *GETDATE*())) **AS N'Количество дней'  
 FROM** Responsibility **AS** Rs  
 **INNER JOIN** Responsible **AS R ON R**.**PersonnelNumber** = Rs.**Responsible  
 INNER JOIN** Equipment **AS** E **ON** Rs.**Equipment** = E.**InventoryID  
 WHERE** *DATEDIFF*(**DAY**, **DateOfReceiving**, *COALESCE*(**ReturnDate**, *GETDATE*())) > @avgDays  
**END**;  
**GO***-- Вывод пользователей задерживающие оборудование дольше среднего***EXEC** *sp\_DisplayUsersDelayingEquipment*;  
**GO***------------------------------------------------------------  
-- Задание №3 Изменение первичного ключа  
-- таблицы Equipment  
------------------------------------------------------------***CREATE PROCEDURE** *sp\_ChangePrimaryKeyEquipment* @oldInventoryID **INT**,  
 @newInventoryID **INT AS  
BEGIN** *-- Временное снятие ограничений* **SET IDENTITY\_INSERT** Equipment **ON**;  
 **ALTER TABLE** Equipment  
 **DROP CONSTRAINT** UQ\_Equipment\_SerialNumber;  
  
 *-- Создание новой записи* **INSERT INTO** Equipment (**InventoryID**, **SerialNumber**, **EquipmentName**, **RegistrationDate**, **Price**, **WarrantyServiceAddress**,  
 **Supplier**)  
 **SELECT** @newInventoryID, **SerialNumber**, **EquipmentName**, **RegistrationDate**, **Price**, **WarrantyServiceAddress**, **Supplier  
 FROM** Equipment  
 **WHERE InventoryID** = @oldInventoryID;  
  
 *-- Изменение связки у зависимостей* **UPDATE** Responsibility  
 **SET Equipment** = @newInventoryID  
 **WHERE Equipment** = @oldInventoryID;  
  
 *-- Удаление старой записи* **DELETE  
 FROM** Equipment  
 **WHERE InventoryID** = @oldInventoryID;  
  
 *-- Восстановление ограничений* **ALTER TABLE** Equipment  
 **ADD CONSTRAINT** UQ\_Equipment\_SerialNumber **UNIQUE** (**SerialNumber**);  
 **SET IDENTITY\_INSERT** Equipment **OFF**;  
**END**;  
**GO***-- Изменение первичного ключа***EXEC** *sp\_ChangePrimaryKeyEquipment* @oldInventoryID = 1, @newInventoryID = 10000;  
  
*-- Вывод сведений об оборудовании с изменёным первичным ключём***SELECT** *\****FROM** Equipment;  
**GO***------------------------------------------------------------  
-- Задание №4 Изменение первичного ключа таблицы Equipment  
-- с удалением связанных данных  
------------------------------------------------------------***CREATE PROCEDURE** *sp\_ChangePrimaryKeyEquipmentWithDel* @oldInventoryID **INT**,  
 @newInventoryID **INT AS  
BEGIN** *-- Временное снятие ограничений* **SET IDENTITY\_INSERT** Equipment **ON**;  
 **ALTER TABLE** Equipment  
 **DROP CONSTRAINT** UQ\_Equipment\_SerialNumber;  
  
 *-- Создание новой записи* **INSERT INTO** Equipment (**InventoryID**, **SerialNumber**, **EquipmentName**, **RegistrationDate**, **Price**, **WarrantyServiceAddress**,  
 **Supplier**)  
 **SELECT** @newInventoryID, **SerialNumber**, **EquipmentName**, **RegistrationDate**, **Price**, **WarrantyServiceAddress**, **Supplier  
 FROM** Equipment  
 **WHERE InventoryID** = @oldInventoryID;  
  
  
 *-- Удаление старой записи* **DELETE  
 FROM** Equipment  
 **WHERE InventoryID** = @oldInventoryID;  
  
 *-- Восстановление ограничений* **ALTER TABLE** Equipment  
 **ADD CONSTRAINT** UQ\_Equipment\_SerialNumber **UNIQUE** (**SerialNumber**);  
 **SET IDENTITY\_INSERT** Equipment **OFF**;  
**END**;  
**GO***-- Переменая для получения старого первичного ключа***DECLARE** @oldInventID **INT**;  
  
*-- Получение старого первичного ключа***SELECT** @oldInventID = **InventoryID  
FROM** Equipment  
**WHERE SerialNumber** = 55953018;  
  
*-- Изменение первичного ключа, с удалением связанных данных***EXECUTE** *sp\_ChangePrimaryKeyEquipmentWithDel* @oldInventoryID = @oldInventID, @newInventoryID = 2582;  
**GO***-- Вывод сведений об оборудовании без удалённых связанных данных***SELECT** *\****FROM** vw\_Responsibility;  
**GO***------------------------------------------------------------  
-- Задание №5 Получение среднего времени задержки  
-- оборудования и список часто задерживаемых устройств  
------------------------------------------------------------***CREATE PROCEDURE** *sp\_DisplayEquipmentDelaying* @inventoryID **INT** = **NULL**, @equipmentName **NVARCHAR**(60) = **NULL AS  
BEGIN  
 IF** @inventoryID **IS NOT NULL OR** @equipmentName **IS NOT NULL  
 SELECT** Equipment.**InventoryID AS N'Инвентарный номер'**,  
 *LOWER*(Equipment.**EquipmentName**) **AS N'Наименование оборудования'**,  
 *AVG*(*DATEDIFF*(**DAY**, **DateOfReceiving**, *COALESCE*(**ReturnDate**, *GETDATE*()))) **AS N'Среднее время задержки'  
 FROM** Responsibility  
 **INNER JOIN** Equipment **ON** Equipment.**InventoryID** = Responsibility.**Equipment  
 WHERE** (@inventoryID **IS NULL OR** Equipment.**InventoryID** = @inventoryID)  
 **AND** (@equipmentName **IS NULL OR** *LOWER*(Equipment.**EquipmentName**) = *LOWER*(@equipmentName))  
 **GROUP BY** Equipment.**InventoryID**, *LOWER*(Equipment.**EquipmentName**);  
 **ELSE  
 BEGIN  
 PRINT N'Не указаны параметры для поиска!'**;  
 **SELECT NULL AS N'Инвентарный номер'**,  
 **NULL AS N'Наименование оборудования'**,  
 **NULL AS N'Среднее время задержки'**;  
 **END** *-- Четыре самых часто задерживаемых устройства* **SELECT TOP** 4 Equipment.**SerialNumber AS N'Серийный номер'**,  
 Equipment.**EquipmentName AS N'Наименование оборудования'**,  
 *AVG*(*DATEDIFF*(**DAY**, **DateOfReceiving**, *COALESCE*(**ReturnDate**, *GETDATE*()))) **AS N'Среднее время задержки'**,  
 *COUNT*(*\**) **AS N'Количество пользователей'  
 FROM** Responsibility  
 **INNER JOIN** Equipment **ON** Equipment.**InventoryID** = Responsibility.**Equipment  
 GROUP BY** Equipment.**SerialNumber**, Equipment.**EquipmentName  
 ORDER BY** *COUNT*(*\**) **DESC**;  
**END**;  
**GO***-- Вывод среднего времени задержки по оборудованию***EXEC** *sp\_DisplayEquipmentDelaying* @inventoryID = 7;  
  
*-- Экспорт базы данных***BACKUP DATABASE** lessonsUrFU **TO DISK** = **N'/var/opt/mssql/data/lessonsUrFU.bak'  
 WITH DESCRIPTION** = **N'Контрольная работа выполненая в рамках дисплины "Базы данных" в УрФУ.'**,  
 **NOFORMAT**, **INIT**, **NAME** = **N'Control Work'**, **SKIP**, **NOREWIND**, **NOUNLOAD**, **STATS** = 10, **CHECKSUM  
GO**

# 3 Образы экранных форм в конструкторе и исполняемой программе с описанием используемых компонент и мастеров.

В процессе работы был использован «Мастер настройки источника данных». С его помощью в проект была добавлена база данных в качестве источника данных, а также строка подключения к базе данных. Основные этапы представлены в следующих снимках экрана:

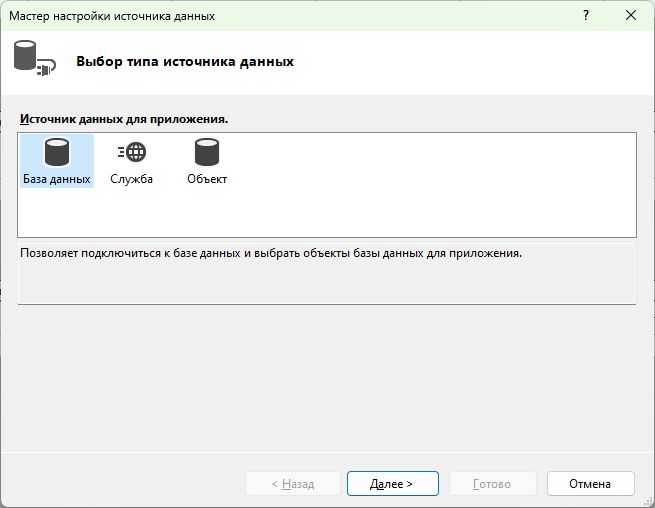


Рисунок №1 – Выбор типа источника данных

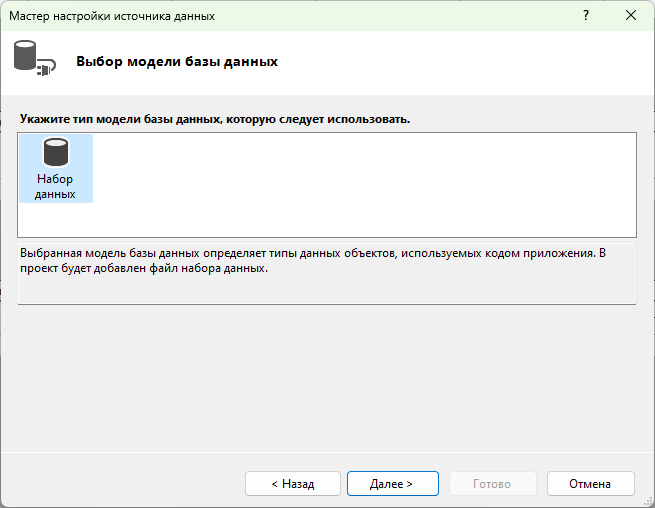


Рисунок №2 – Выбор модели базы данных

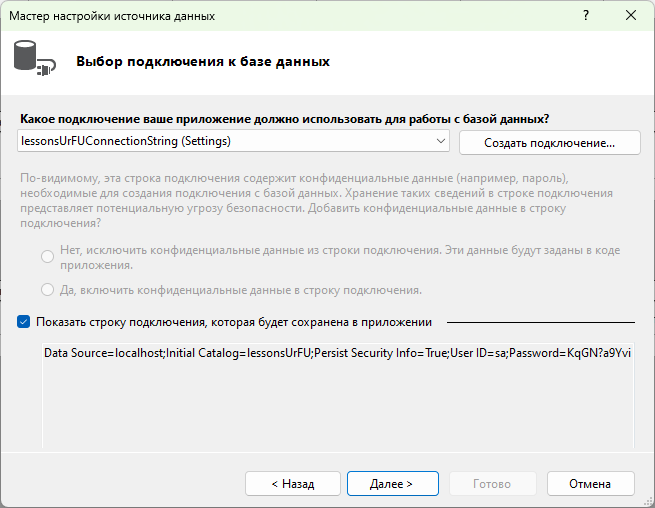


Рисунок №3 – Выбор (и создание) строки подключения к базе данных

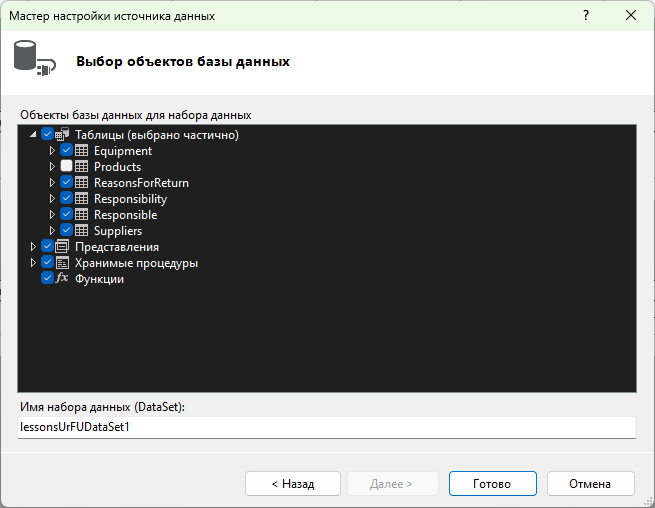


Рисунок №4 – Выбор требуемых объектов базы данных

Описание основных объектов, используемых в созданной форме:

* **tablesBindingNavigator:** объект класса ***BindingNavigator***, представляющий собой навигационную панель, при работе с таблицами баз данных.
* **tabControl:** объект класса ***TabControl***, выполняющий объединение ряда страниц приложения с помощью панели с вкладками;
* **label{\*}:** текстовые метки (где в названии **{\*}** является названием таблицы), используемая в качестве заголовка перед таблицей. Представляют собой объекты класса ***Label***.
* **dataGridView{\*}:** контейнеры (где в названии **{\*}** является названием таблицы), выполняющие отображение строк и столбцов таблицы. Позволяют, создавать, изменять и удалять записи. Являются объектами класса ***DataGridView***.
* **dataSetLessonUrfu:** представляет собой кэш данных из используемой базы данных lessonsUrFU. Кэш используется в дальнейшем другими объектами ADO.NET. Является объектом сгенерированного класса, наследуемого от класса ***System.Data.DataSet***. Не отображаются в форме.
* **bindingSource{\*}:** инкапсулируют источники данных и обеспечивают возможности навигации, фильтрации, сортировки и обновления данных. Являются объектами класса ***BindingSource***. В указанном названии **{\*}** является названием таблицы. Не отображаются в форме, используются рядом других объектов. Является экземплярами класса ***BindingSource***.
* **tableAdapter{\*}:** выполняют подключение к базам данным для выполнения ряда запросов. Тем самым он также выполняет синхронизацию данных ***DataGridView*** с базой данных. Являются объектами сгенерированных классов. В указанном названии **{\*}** является названием таблицы. Не отображаются в форме.

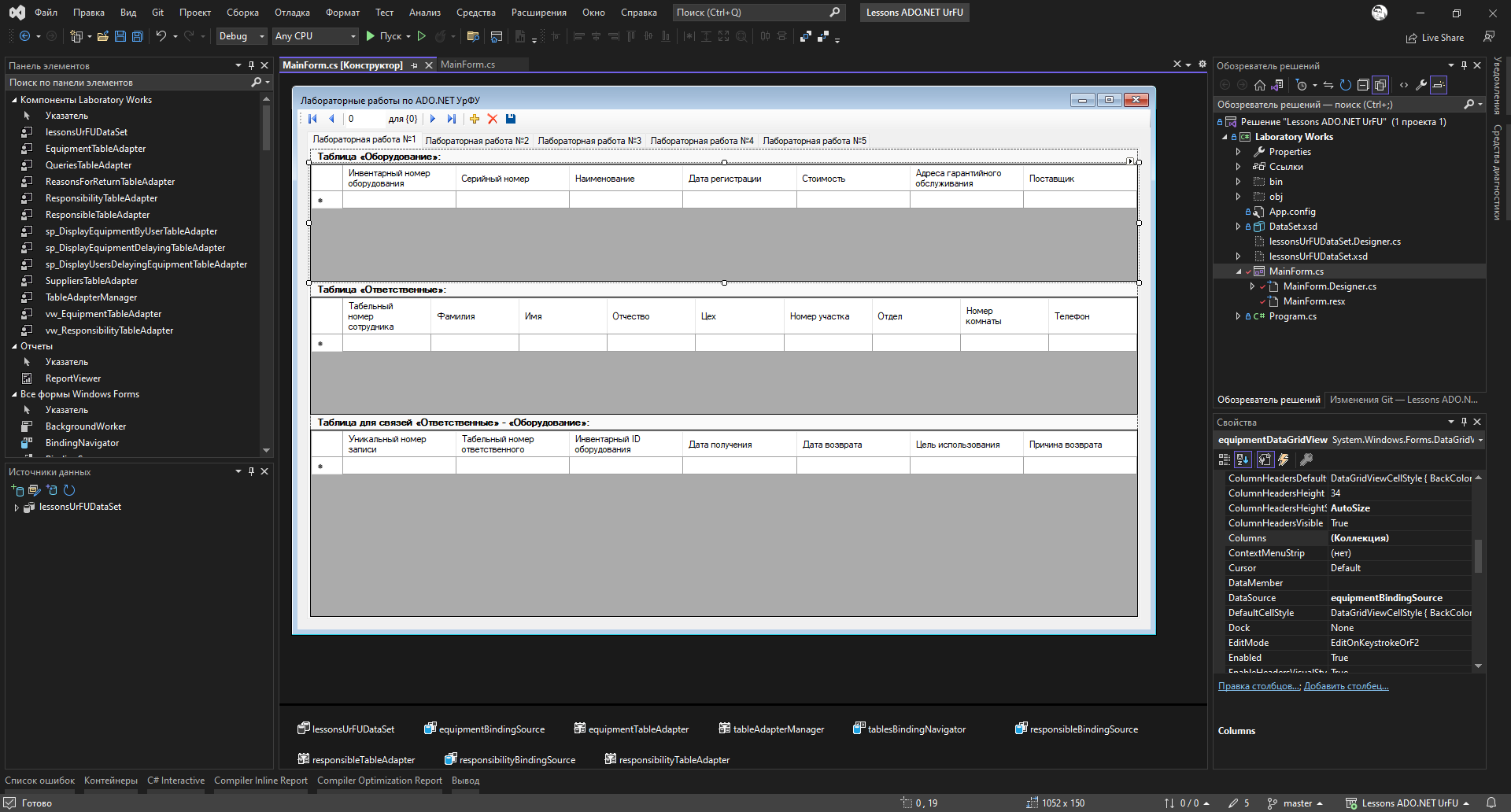


Рисунок №5 – Экранная форма в конструкторе

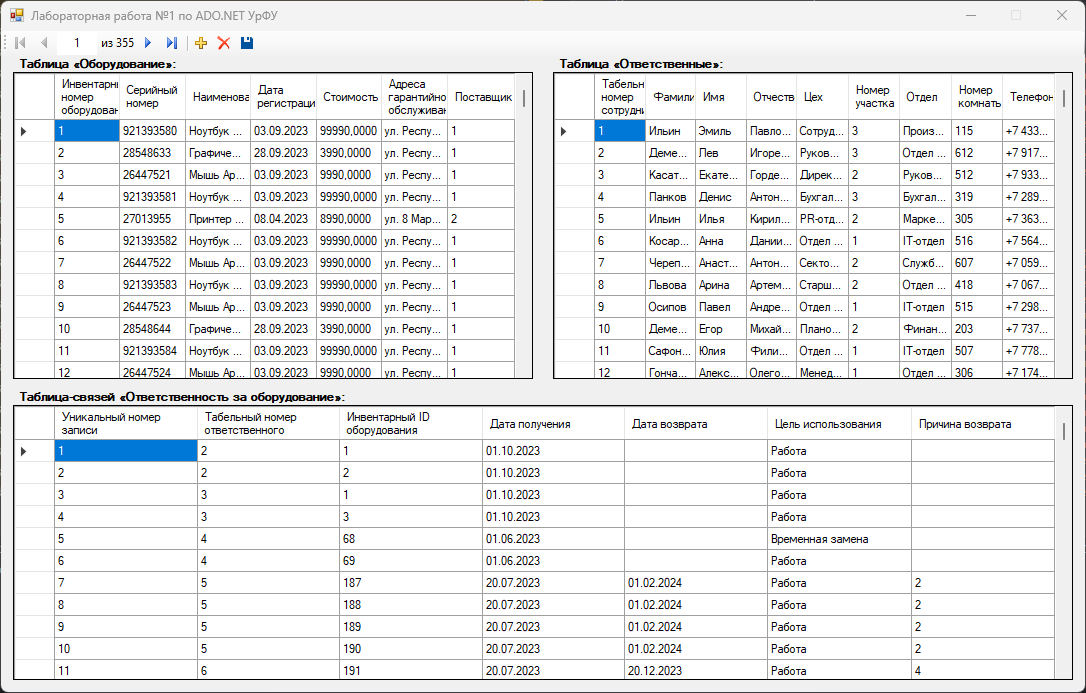


Рисунок №6 – Исполняемая программа

# 4 Описание навигационной панели и её использования для управления данными



Рисунок №7 – Навигационная панель

Навигационная панель представляет собой объект класса ***System.Windows.Forms.BindingNavigator***. Он включает в себя ряд элементов класса ***ToolStripButton***:

* **navigatorMoveFirstItem:** выполняет переход к первой строке.
* **navigatorMovePreviousItem:** выполняет переход к предыдущей строке.
* **navigatorPositionItem:** отображает текущую позицию в ***DataGridView*** и позволяет изменить её, путём ввода нового значения.
* **navigatorCountItem:** общее количество строк в текущем ***DataGridView***.
* **navigatorMoveNextItem:** выполняет переход к следующей строке.
* **navigatorMoveLastItem:** выполняет переход к последней строке.
* **navigatorAddNewItem:** инициирует добавление новой строки в ***DataGridView***.
* **navigatorDeleteItem:** выполняет удаление строки из ***DataGridView***.
* **navigatorSaveItem:** выполняет отправку изменений сделанных в ***DataGridView***, в исходную базу данных.