

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ

Н.В. Морозова,
З. Б. Батчаева

БАЗЫ ДАННЫХ

Практикум для обучающихся 3 курса
по направлению подготовки 09.03.04 Программная инженерия

Черкесск
2024

УДК 004.6
ББК 32.973
М 80

Рассмотрено на заседании кафедры Прикладной информатики
Протокол № 2 от 18.09 2023 г.
Рекомендовано к изданию редакционно-издательским советом СКГА.
Протокол № 26 от 29. 09. 2023 г.

Рецензенты: Кочкарлова П. А. – к.ф.– м. н., доцент кафедры ПИ

М80 Морозова, Н.В. Базы данных: практикум для обучающихся 3 курса по направлению подготовки 09.03.04 Программная инженерия / Н. В. Морозова, З. Б. Батчаева.– Черкесск: БИЦ СКГА, 2024. – 44 с.

Практикум для выполнения лабораторных работ по дисциплине «Базы данных» в СУБД MS Access и программной среде Visual Studio 2022. В нем раскрыты последовательно вопросы создания таблиц, подключения файла с базой данных в программной среде Visual Studio, создания форм и поиска данных по выбранным параметрам, по результатам поиска описан процесс создания отчетов.

Данное пособие разработано для обучающихся вузов по направлению 09.03.04 Программная инженерия.

УДК 004.6
ББК 32.973

© Морозова Н.В., Батчаева З. Б., 2024
© ФГБОУ ВО СКГА, 2024

СОДЕРЖАНИЕ

1. Введение	4
2. Цель и задачи выполнения лабораторных работ	5
3. Лабораторная работа № 1.Создание таблиц базы данных в СУБД MS Access	6
4. Лабораторная работа № 2.Подключение базы данных в Visual Studio	12
5. Лабораторная работа № 3 Разработка пользовательского интерфейса в Visual Studio: создание форм	21
6. Лабораторная работа № 4 Поиск данных в БД	31
7. Лабораторная работа № 5 Создание отчетов по выбранным параметрам	35
Список использованных источников	41

Введение

Практикум содержит рекомендации по выполнению лабораторных работ по дисциплине «Базы данных» в СУБД MS Access и программной среде Visual Studio 2022. В СУБД MS Access 2022 обучающиеся проектируют и создают структуру базы данных, заполняют таблицы, устанавливают связи между ними.

Для разработки приложения используется программная среда Visual Studio 2022.

Visual Studio 2022 - интегрированная среда разработки, позволяющая создавать разные приложения, используя богатый выбор инструментов среды.

В практикуме раскрыты последовательно вопросы создания таблиц, подключения файла с базой данных в программной среде Visual Studio, создания форм и поиска данных по выбранным параметрам, по результатам поиска описан процесс создания отчетов. Рассматривается пример готового приложения.

Данное пособие разработано для обучающихся вузов по направлению 09.03.04 Программная инженерия.

2.Цель и задачи выполнения лабораторных работ

Целью выполнения лабораторных работ является получение практических навыков работы в СУБД MS Access 2022 и программной среде Visual Studio 2022:

1. применение языка SQL для формирования простых и сложных запросов к реляционной БД, которые позволяют создать, изменить, найти, удалить, добавлять записи в БД;

2. формирование умений организовать поиск, хранение, обработку и анализ информации, ввод информации в базу данных и её представление в требуемом формате;

3. формирование умений корректно проектировать структуру БД с учетом требований нормализации отношений и ограничений предметной области, функциональных зависимостей.

Задачи пособия: научить обучающихся проектировать и создавать таблицы базы данных, устанавливать связи между ними; создавать формы, организовывать поиск данных по выбранным параметрам и формировать отчеты по результатам поиска данных.

В результате изучения данного пособия обучающийся усвоит следующие компетенции:

1. ОПК-8.1. Корректно проектирует базы данных с учетом функциональных зависимостей.

2. ОПК-8.2. Грамотно осуществляет поиск, хранение, обработку и анализ информации из различных источников и баз данных.

3. ОПК-8.3. Интерпретирует основные понятия информационных систем и баз данных, определяет состав и основные функции систем управления базами данных.

ЛАБОРАТОРНАЯ РАБОТА № 1

Тема: Создание таблиц баз данных в СУБД Access

Цель выполнения работы. Создание базы данных: таблиц и связей между ними в СУБД Access.

Контрольное задание

Создать базу данных для выбранной предметной области: определить структуру таблиц и заполнить таблицы базы данных, установить связи между ними.

Решение задачи состоит из нескольких этапов:

1. Проведем анализ предметной области - «Питомник». У каждого обучающегося определена своя предметная область.

Открыв окно СУБД Access, необходимо выбрать раздел «Новая база данных» (рис.1).

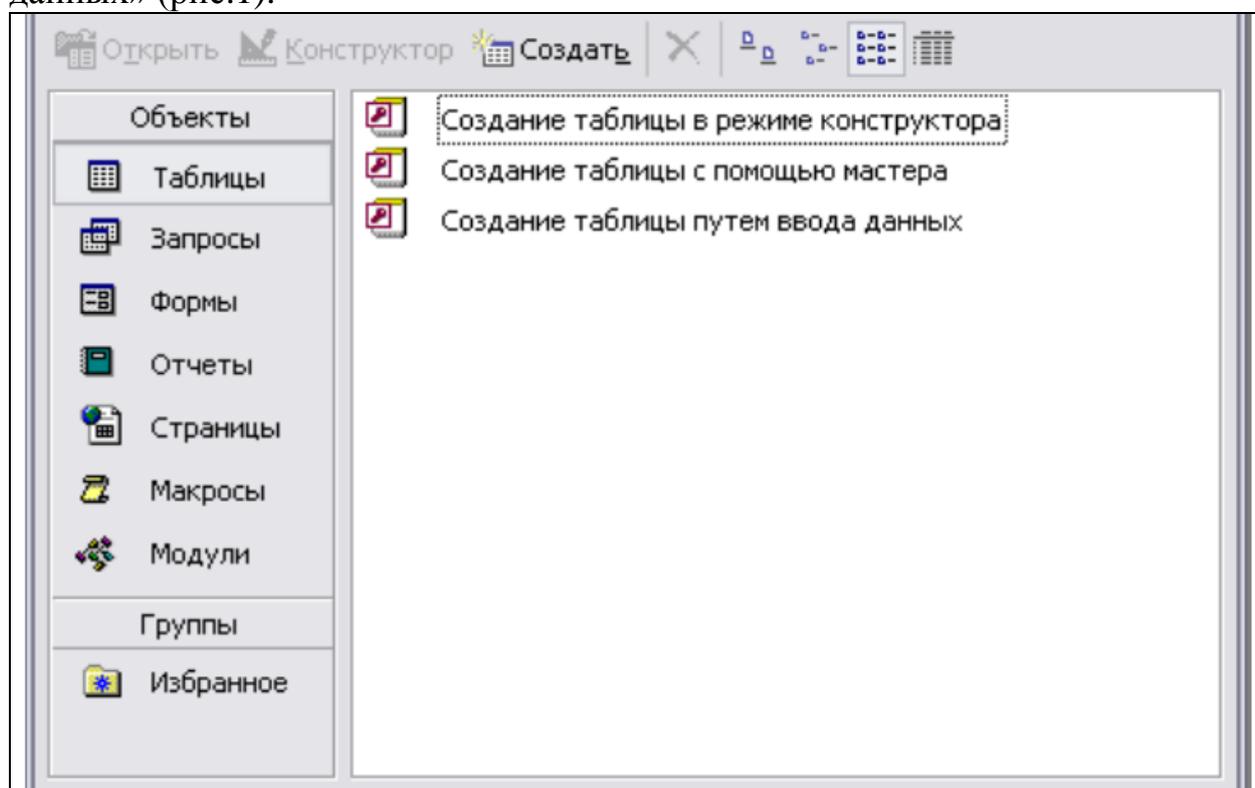


Рисунок 1– Окно создания базы в СУБД Access

Опишем сущности данной предметной области: Животные, Заявки, Сотрудники, Порода и Тип животного, Клиенты. Определим состав 6 таблиц нашей базы данных «Питомник» (таблицы 1 -6).

Таблица 1 – Породы

Наименование полей	Тип	Примечание
Код	Счетчик	Ключевое поле
Порода	Текстовый	

Таблица 2 – Животные

Наименование полей	Тип	Примечание
Код_животного	Счетчик	Ключевое поле
Кличка	Текстовый	
Возраст	Числовой	
Код_породы	Числовой	
Прививки	Логический	
Тип_животного	Числовой	

Таблица 3 – Заявки

Наименование полей	Тип	Примечание
Код_заявки	Счетчик	Ключевое поле
Код_животного	Числовой	
ID_клиента	Числовой	
Дата_заявки	Дата время	
Код_сотрудника	Числовой	

Таблица 4 – Клиенты

Наименование полей	Тип	Примечание
Код_клиента	Счетчик	Ключевое поле
Фамилия	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Адрес	Текстовый	
Телефон	Текстовый	

Таблица 5 – Данные о сотрудниках

Наименование полей	Тип	Примечание
Код_сотрудника	Счетчик	Ключевое поле
Фамилия	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Адрес	Текстовый	
Должность	Текстовый	
Оклад	Числовой	

Таблица 6 – Тип животного

Наименование полей	Тип	Примечание
Код	Счетчик	Ключевое поле
Тип_животного	Текстовый	

3. Создание таблиц в СУБД Access.

Выбираем объект *Таблицы* (пункт из списка «Объекты») и кнопку *Создать*, определяем режим *Конструктор*, выбираем пункт *Создание таблицы в режиме конструктора* рабочей области окна, после чего создаем таблицу *Животные* с указанием заголовка столбцов, типов полей и ключевого поля. Ключевым полем в данной таблице является поле – *Код_животного*. Результат показан на рис.2-3.

Имя поля	Тип данных
Код_животного	Счетчик
Кличка	Короткий текст
Возраст	Короткий текст
Код_породы	Числовой
Прививки	Логический
Тип_животного	Числовой

Рисунок 2 – Таблица «Животные» в режиме конструктора

Код животн	Кличка	Возраст	Код породы	Прививки	Тип животн	Ще
1	Басар	2	2	<input checked="" type="checkbox"/>	1	
2	Мухтар	1	1	<input type="checkbox"/>	1	
3	Ричард	2	1	<input type="checkbox"/>	2	
4	Пушок	1	1	<input type="checkbox"/>	2	
5	Самсон	3	1	<input checked="" type="checkbox"/>	2	
6	Сеня	2	1	<input checked="" type="checkbox"/>	1	
7	Симбат	4	1	<input type="checkbox"/>	1	
(№)			0	<input type="checkbox"/>	0	

Рисунок 3 – Таблица «Животные» в режиме ввода данных

Аналогичным образом создаем остальные таблицы (рис 4-13).

Имя поля	Тип данных
Код_заявки	Счетчик
ID_клиента	Числовой
Код_животного	Числовой
Дата_заявки	Дата и время
Код_сотрудника	Числовой

Рисунок 4 – Таблица «Заявки» в режиме конструктора

Код заявки	ID клиента	Код животн	Дата заявки	Код сотрудн
1	1	1	18.01.2024	1
2	1	2	16.01.2024	2
3	3	3	14.01.2024	3
(№)	0	0		0

Рисунок 5 – Таблица «Заявки» в режиме ввода данных

Имя поля	Тип данных
Код_клиенты	Счетчик
Фамилия	Короткий текст
Имя	Короткий текст
Отчество	Короткий текст
Адрес	Короткий текст
Телефон	Короткий текст

Рисунок 6 – Таблица «Клиенты» в режиме конструктора

Код_клиент	Фамилия	Имя	Отчество	Адрес	Телефон
+	1 Иванов	Алексей	Андреевич	Ленинна 12	89223444312
+	2 Исмаилов	Тимур	Мамбетович	Энгельса 13	89883331245
+	3 Шибалов	Иракий	Прокофиевич	Леннина 55	89275543377
+	4 Валерев	Александр	Иванович	Кирова 12	88991234456
+	5 Мартынов	Артем	Андреевич	Суслова 33	89234437765
*	(№)				

Рисунок 7 – Таблица «Клиенты» в режиме ввода данных»

Имя поля	Тип данных
Код	Счетчик
Порода	Короткий текст

Рисунок 8 – Таблица «Порода» в режиме конструктора»

Код	Порода
+	1 Не выявлена
+	2 Алабай
+	3 Немецкая овчарка
	(№)

Рисунок 9 – Таблица «Порода» в режиме ввода данных

Имя поля	Тип данных
Код_сотрудника	Счетчик
Фамилия	Короткий текст
Имя	Короткий текст
Отчество	Короткий текст
Адрес	Короткий текст
Должность	Короткий текст
Оклад	Короткий текст

Рисунок 10 – Таблица «Сотрудники» в режиме конструктора

Код сотруд-	Фамилия	Имя	Отчество	Адрес	Должность	Оклад
1	Иванов	Александр	Тимурович	Кирова 12	Ветеринар	30000
2	Любимцева	Кристина	Анзоровна	Энгельса 34	Дрессировщик	35000
3	Качурин	Денис	Станиславович	Карла-маркса	Кипер	27000
4	Эсаулов	Вацлав	Ипполитович	Кирова 32	Кипер	27000
5	Гребенщиков	Петр	Герасимович	Лермонтова 56	Ветеринал	30000
6	Арнаутов	Магомед	Асланович	Луговая 32	Охранник	15000
7	Исламов	Хасан	Алиевич	Лазо 61	Дрессировщик	35000

Рисунок 11 – Таблица «Сотрудники» в режиме ввода данных

Имя поля	Тип данных
Код	Счетчик
Тип_животного	Короткий текст

Рисунок 12 – Таблица «Тип животное» в режиме конструктора

Код	Тип_животного
1	Собака
2	Кошка
3	Морская свинка
4	Хомяк
5	Попугай

Рисунок 13 – Таблица «Тип животное» в режиме ввода данных

4. Создание схемы данных.

Для создания связи между таблицами, можно использовать пункт меню *Access Сервис~Схема данных*.

Связывание полей осуществляется перетаскиванием поля из одной таблицы в другую.

Например, поле из таблицы **Сотрудники** *код_* сотрудника необходимо переместить в таблицу **Заявки** на аналогичное поле.

Следует запомнить, что поля, по которым выполняется связь, должны быть одинаково описаны.

Примечание. Допускается связь полей с типами Счетчик и Числовой. В окне установки связей определяются настройки связей (рис. 14).

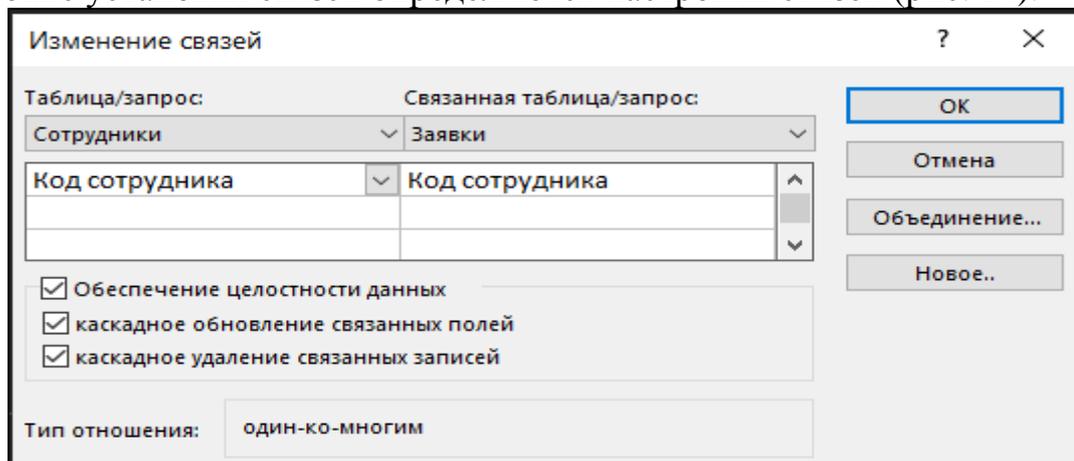


Рисунок 14 – Установка связей между таблицами

Результат установления связей в базе данных представлен на рис. 15.

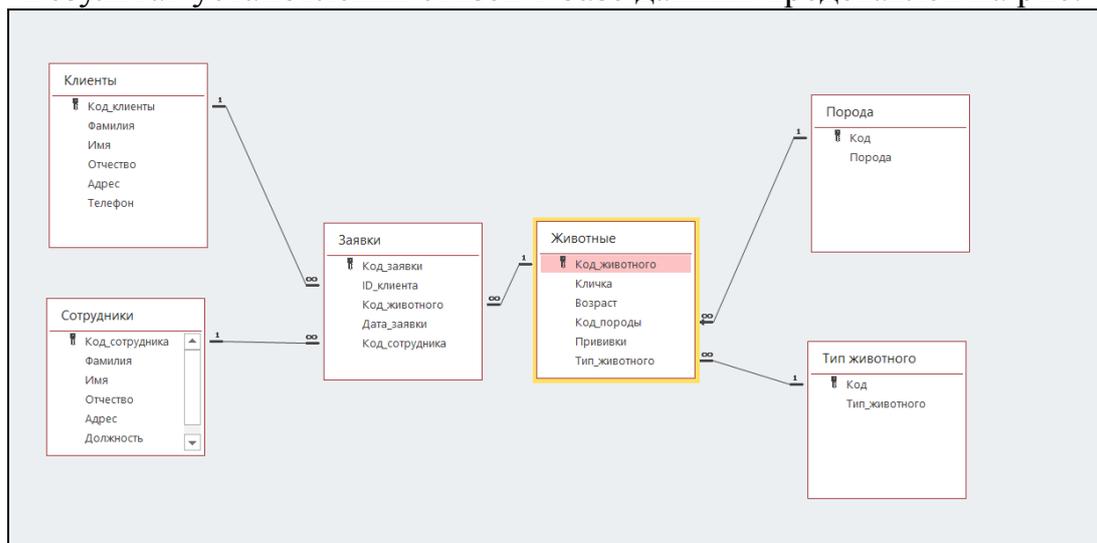


Рисунок 15 – Схема данных

Индивидуальное вариантное задание

1) Определить список необходимых таблиц и дополнительных атрибутов.

2) Создать дополнительную таблицу или добавить новые атрибуты в модель. Если необходимо, откорректировать схему данных («привязать» новую таблицу).

3) Заполнить таблицы данными: не менее 5 строк для таблиц справочников и 10 – для таблиц заявок.

Для связи таблицы Клиенты с таблицами Заявки и Сотрудники были добавлены поля - Код_животного, Id_клиента, Код_сотрудника; для связи таблицы Животные с таблицами Заявки, Порода и Тип животного были добавлены следующие атрибуты, которые являются внешними ключами для главной таблицы: Код_породы, Тип животного, которые приведены в таблице 7.

Таблица 7 – Дополнительные атрибуты для таблиц

Номер	Наименование поля	Тип (формат) поля
1	2	3
Заявки		
1	Код_животного	Числовой
2	Id_клиента	Числовой
3	Код_сотрудника	Числовой
Животные		
1	Код_породы	Числовой
2	Тип животного	Числовой

Контрольные вопросы

1. Опишите процесс создания таблицы в СУБД Access.
2. Перечислите какие бывают типы данных в Access.
3. Поясните на примере как можно связать две таблицы, какие дополнительные атрибуты для этого задаются?
4. Для чего применяется схема данных в Access?

ЛАБОРАТОРНАЯ РАБОТА № 2

Тема: Подключение базы данных в Visual Studio

Цель выполнения работы. Подключить созданную базу данных в программной среде Visual Studio 2022.

Контрольное задание

Необходимо открыть созданную базу данных Питомник в среде Visual Studio 2022.

Шаг 1. Нужно сохранить файл ACCESS в формате .mdb, для этого перейдите во вкладку ФАЙЛ-Сохранить как—2002-2003(*.mdb) (рис. 16). Необходимо сделать данное сохранение так как среда Visual Studio открывает файлы базы данных в формате (*.mdb).

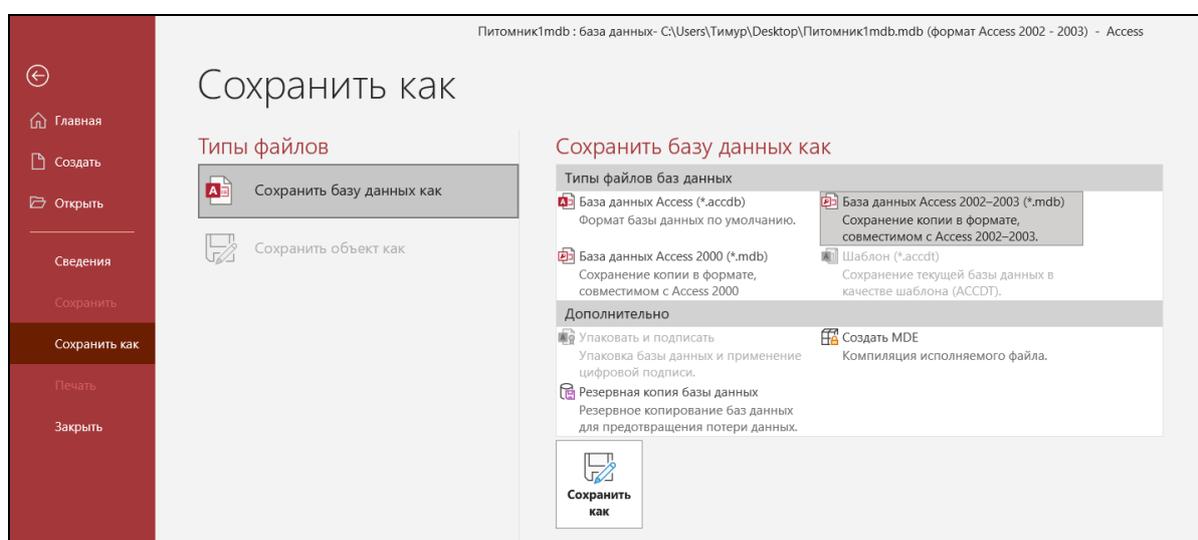


Рисунок 16 – Сохранение базы данных в Access

Шаг 2. Установка среды разработки на компьютер. «Можно использовать бесплатную и полнофункциональную среду разработки - Visual Studio Community 2022 (рис.17-рис. 18). Загрузим установщик Visual Studio по адресу: <https://www.visualstudio.com/en-us/downloads>. Чтобы добавить в Visual Studio поддержку проектов для Windows Forms и C# и .NET, в программе установки среди рабочих нагрузок нужно выбрать только пункт Разработка классических приложений .NET.»[3].

Шаг 3. Создание проекта графического приложения. При старте программы выберем пункт создать новый проект(рис. 19). На следующем окне в качестве типа проекта выберем Windows Forms App(рис.20). Выберем каталог и имя нашего проекта(рис.21). В поле Project Name дадим проекту какое-либо название – например, HelloApp.

Выберем версию, которую будет использовать для проекта(рис.21). Выберем последнюю на данный момент версию - .NET и нажмем на кнопку Create (Создать) для создания проекта. После этого Visual Studio откроет наш проект с созданными по умолчанию файлами(рис.23).

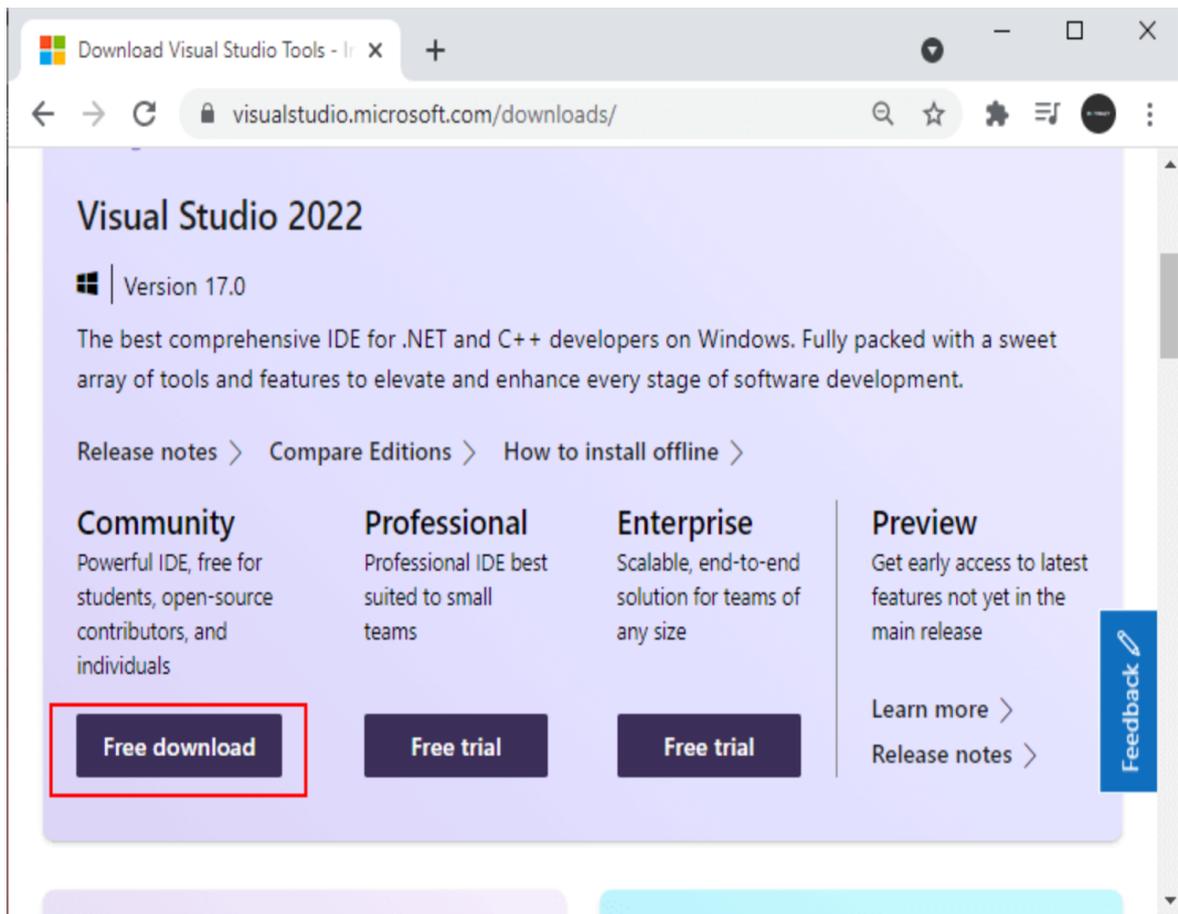


Рисунок 17 – Скачивание среды разработки с сайта

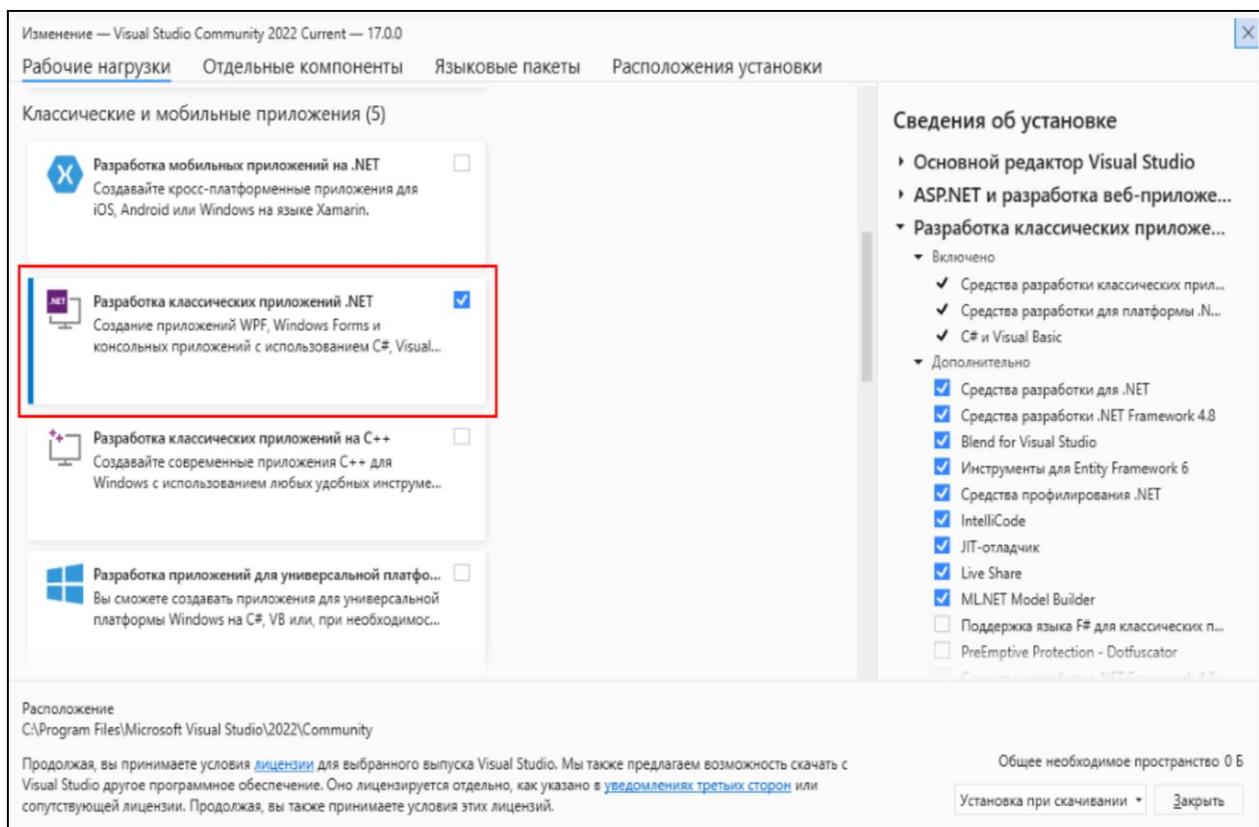


Рисунок 18 – Выбор поддержки проектов в качестве пункта при установке

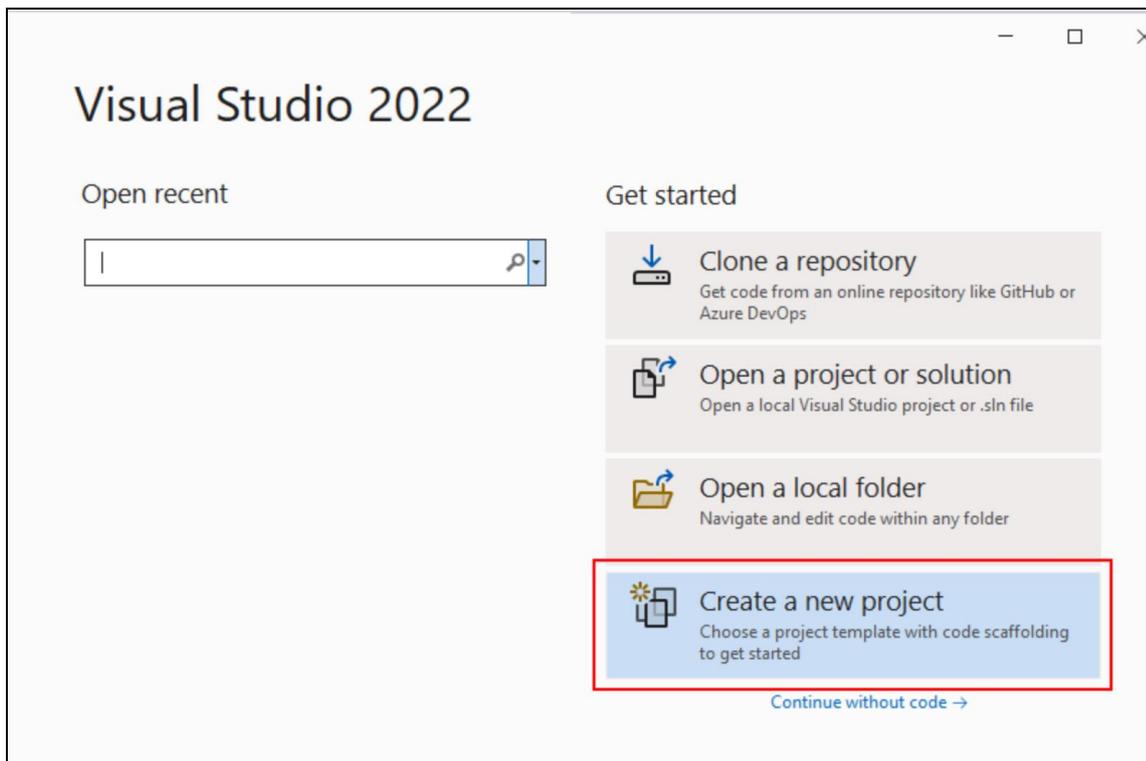


Рисунок 19 – Стартовое окно среды разработки

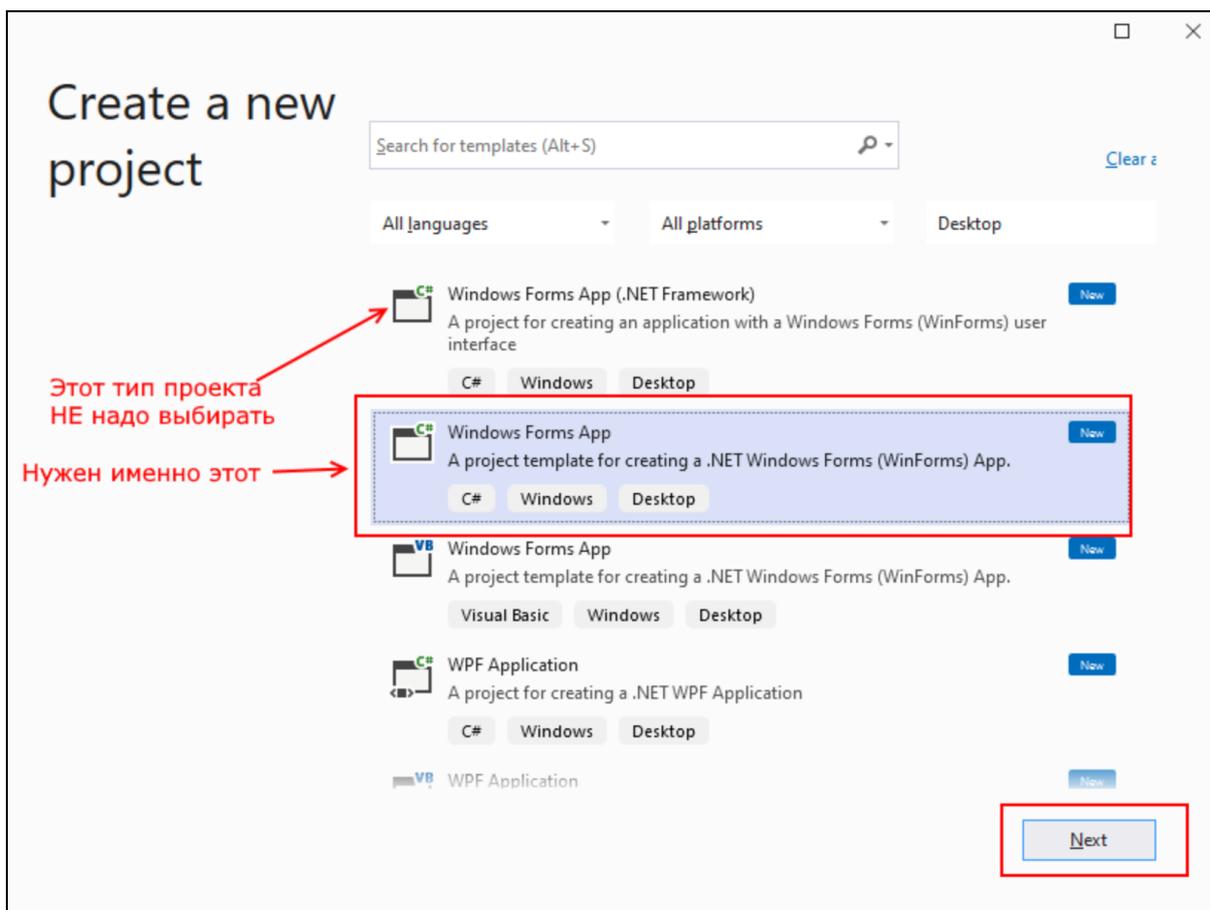


Рисунок 20 – Выбор типа проекта

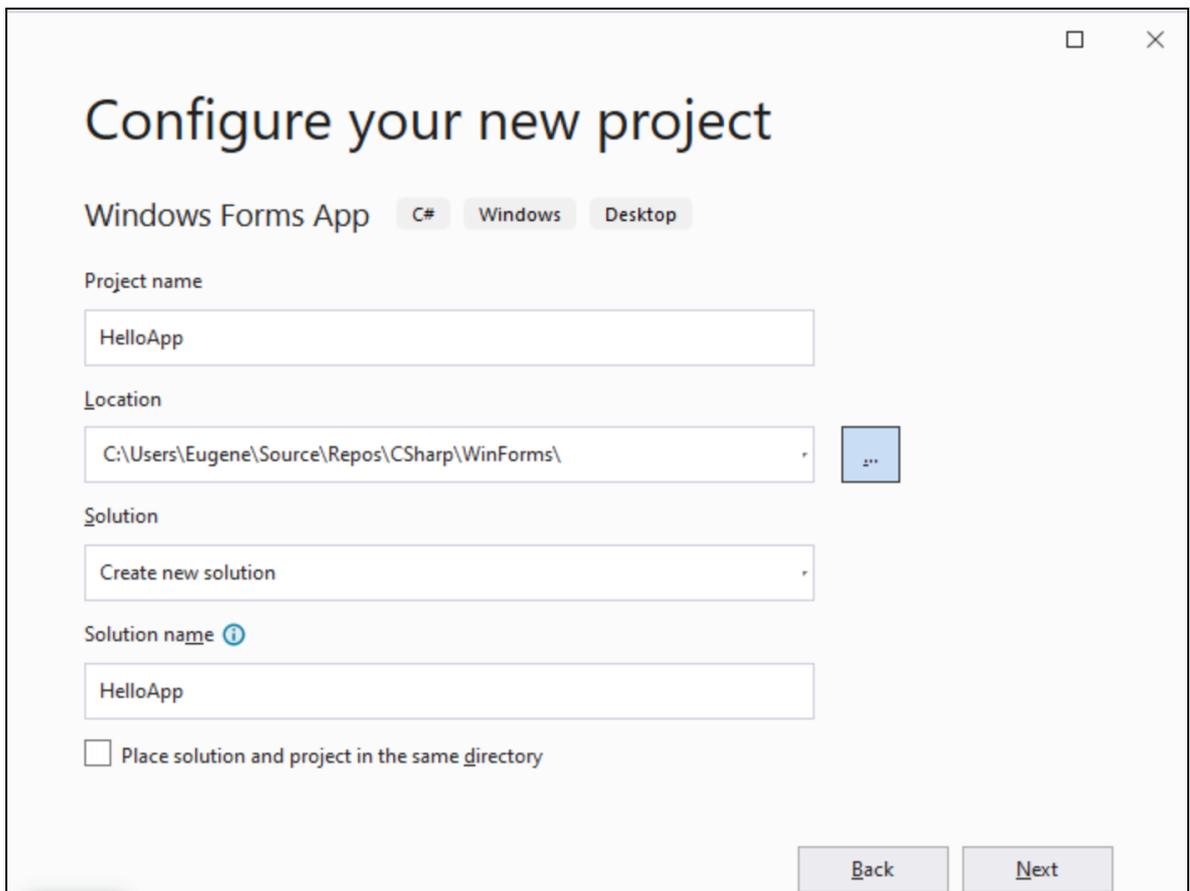


Рисунок 21 – Окно выбора места хранения проекта

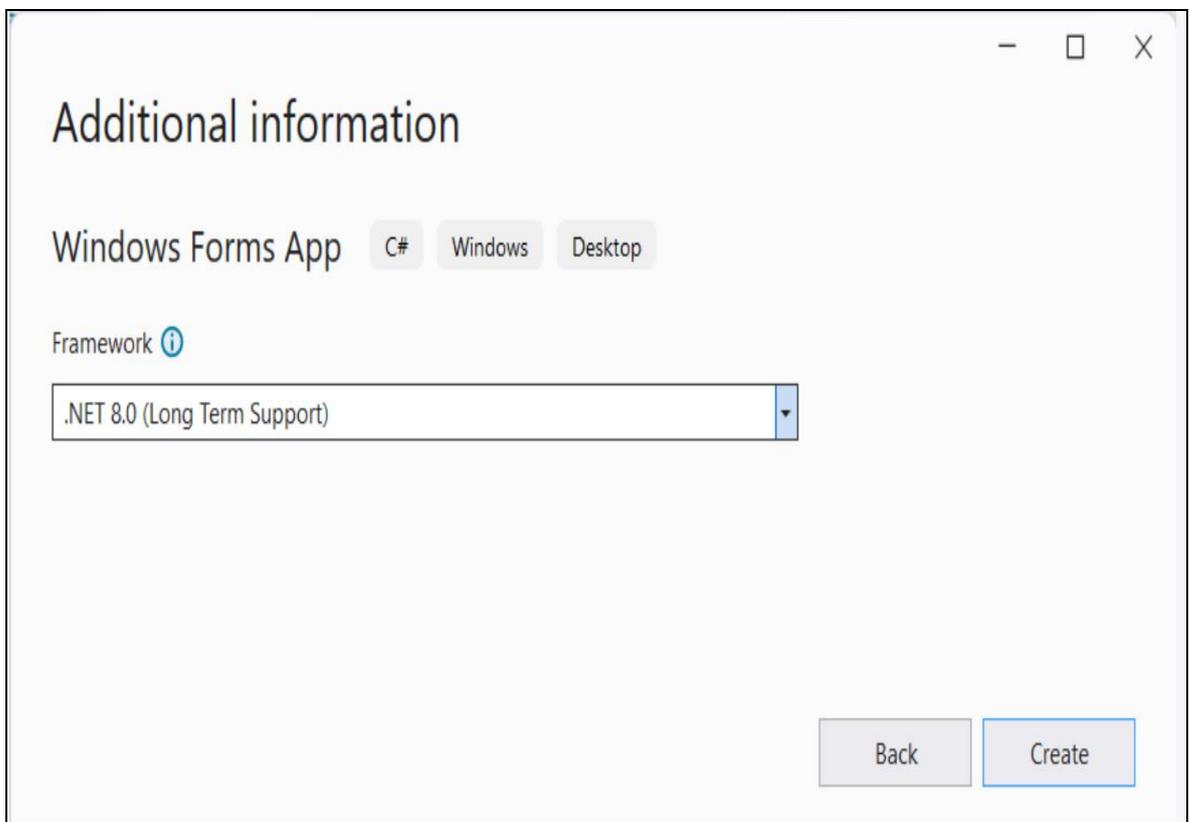


Рисунок 22 – Окно выбора версии проекта

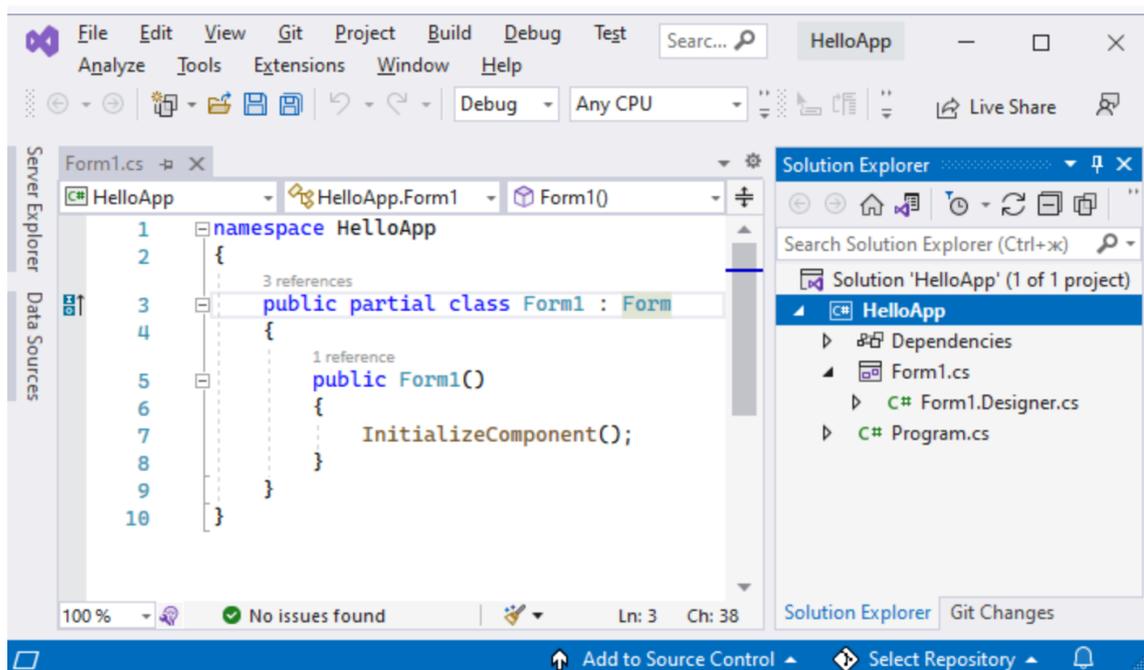


Рисунок 23 – Окно созданного проекта

Справа находится окно Solution Explorer определяющее структуру нашего проекта. «Практически этот тот же проект, который создается с помощью .NET CLI:

Dependencies - это узел содержит сборки dll, которые добавлены в проект по умолчанию. Эти сборки как раз содержат классы библиотеки .NET, которые будет использовать C#

Form1.Designer.cs: он содержит определение компонентов формы, добавленных на форму в графическом дизайнера

Далее идет файл единственной в проекте формы - Form1.cs, который по умолчанию открыт в центральном окне.

Program.cs определяет точку входа в приложение»[3].

Для отладки приложения выбираем клавишу F5 или зеленую стрелочку на панели инструментов(рис.24).

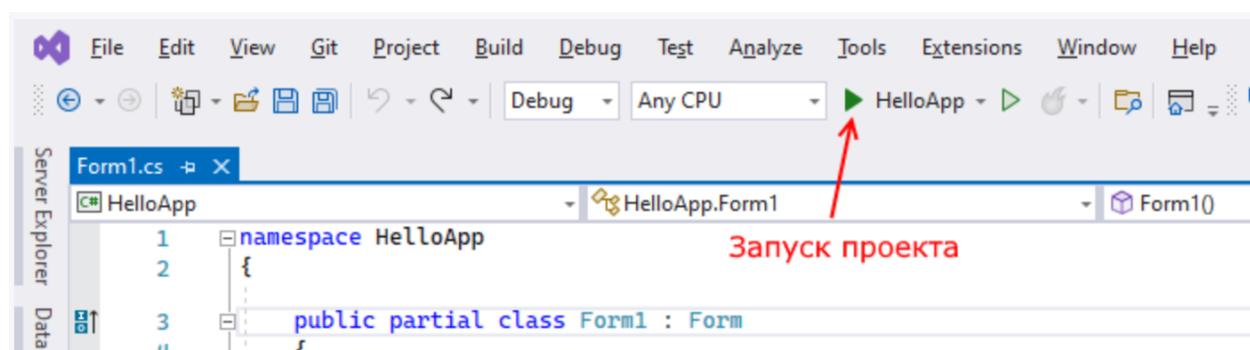


Рисунок 24 – Окно запуска проекта в режиме отладки

После запуска приложения среда компилирует его в файл с расширением exe. Найти данный файл можно, зайдя в папку проекта и далее в каталог \bin\Debug\net8.0-windows(рис.25).

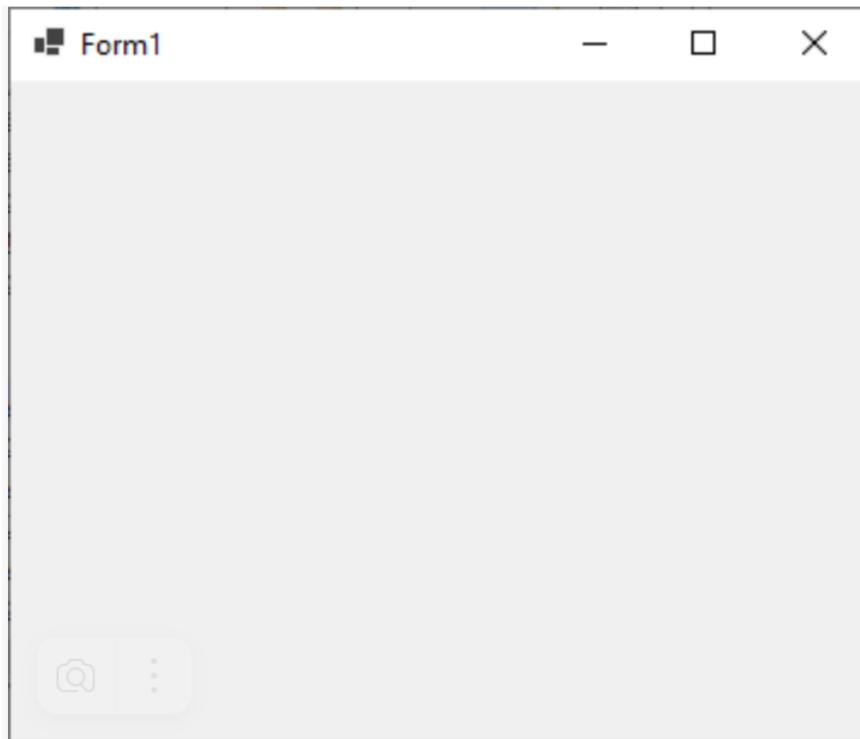


Рисунок 25 – Окно пустой формы в режиме запуска проекта

Подключим созданную базу, для этого нужно выбрать проект в Visual Studio, для этого нужно выбрать команду на вкладке проект «Добавить новый источник данных» (рис.26).

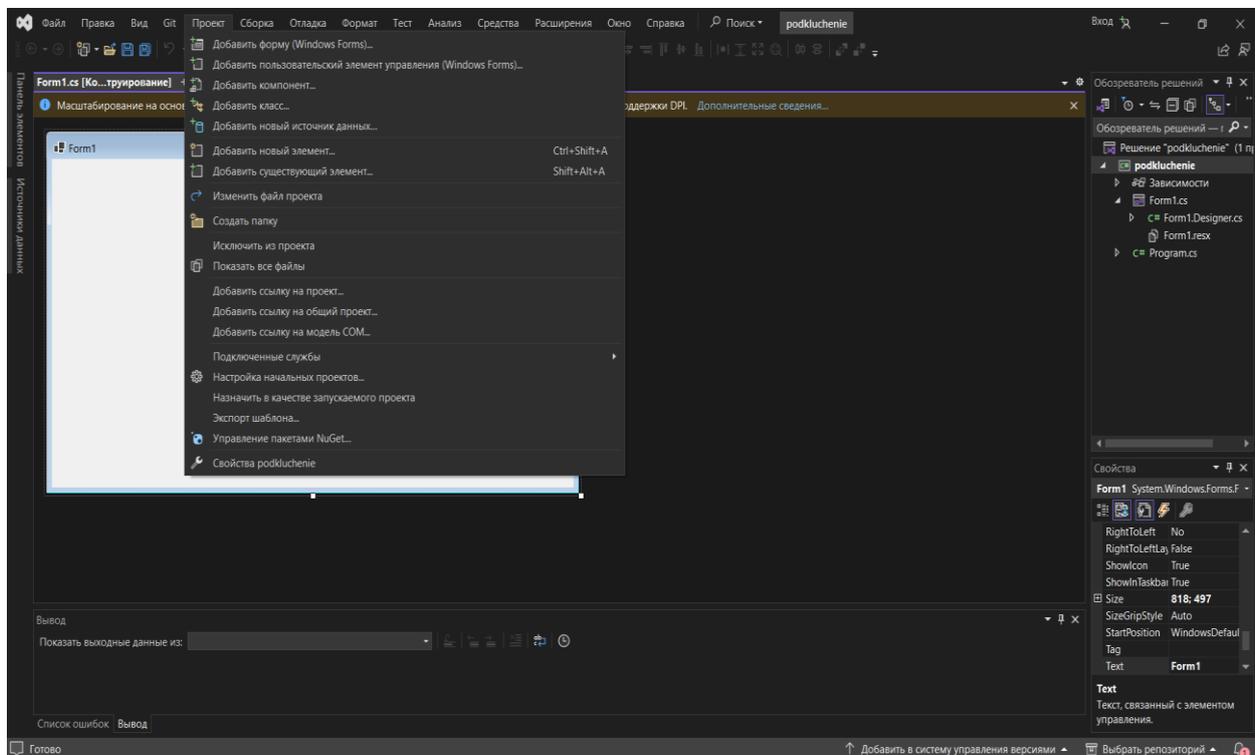


Рисунок 26 – Добавление нового источника данных

Появится окно мастера настройки источника данных (рис. 27).

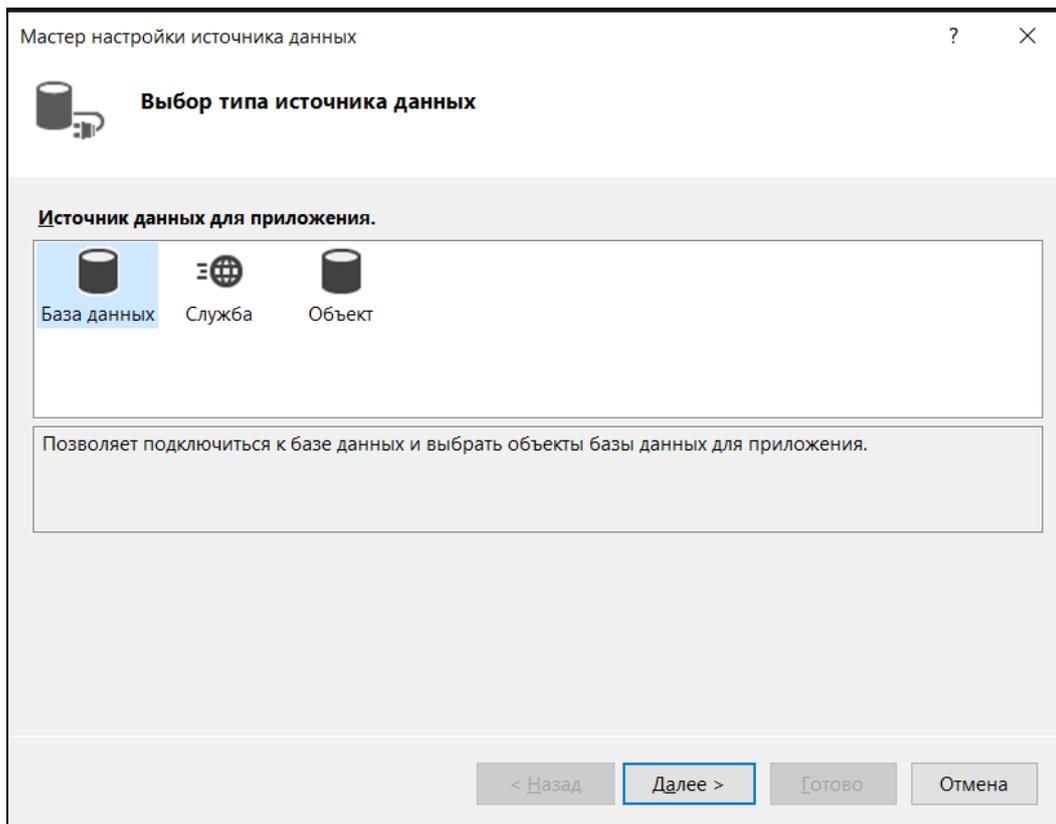


Рисунок 27 – Окно мастера настройки источника данных

В открытом окне нажимаем создать подключение, далее выбираем нашу созданную базу - Файл базы данных Microsoft Access (рис. 28).

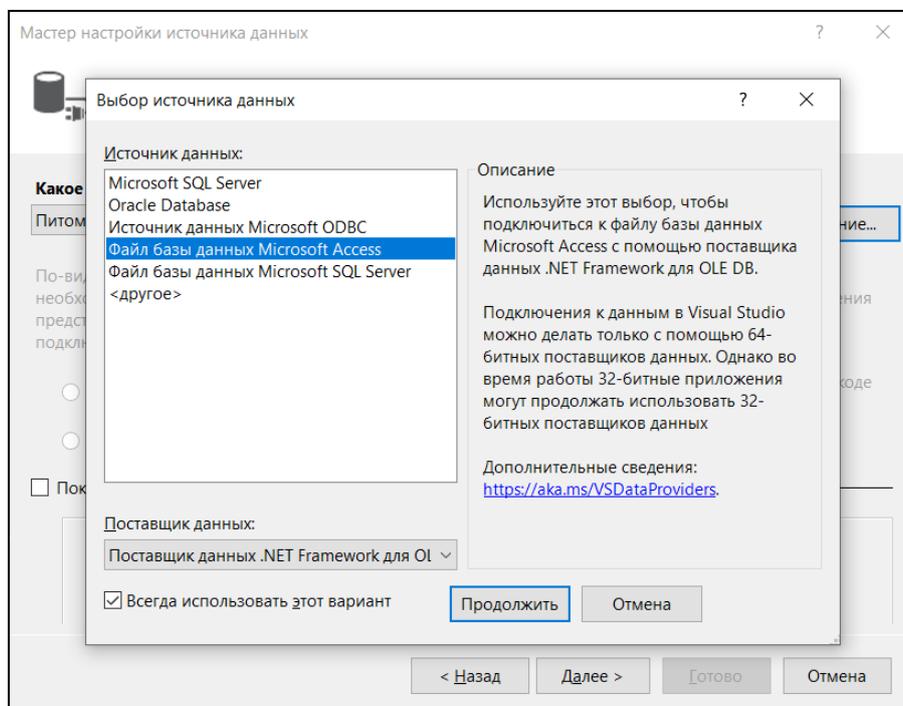


Рисунок 28 – Окно выбора источника данных

Выбираем файл с созданной базой (*.mdb) (рис. 29).

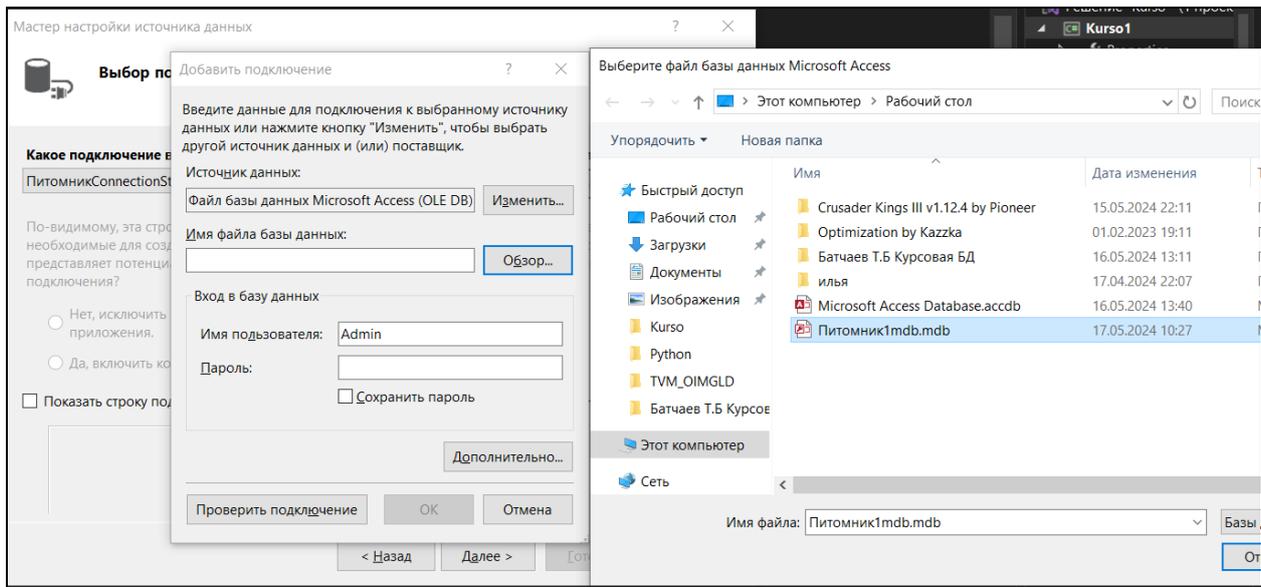


Рисунок 29 – Окно добавления подключения

В следующем окне ничего не меняем и идем далее (рис. 30).

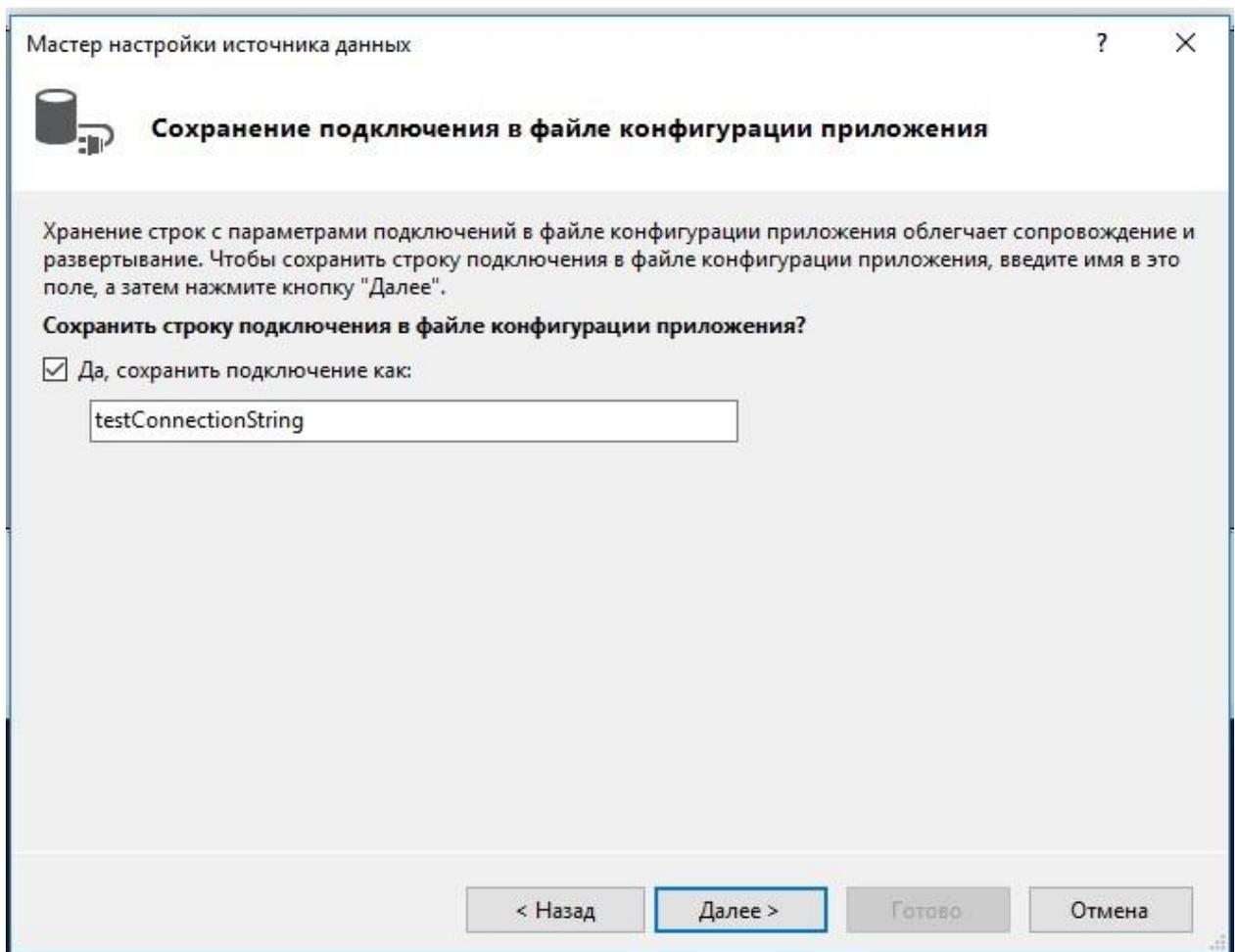


Рисунок 30– Окно сохранения подключения в файле конфигурации

В следующем окне выбираем таблицы подключаемой базы и ждем «Готово» (рис. 31).

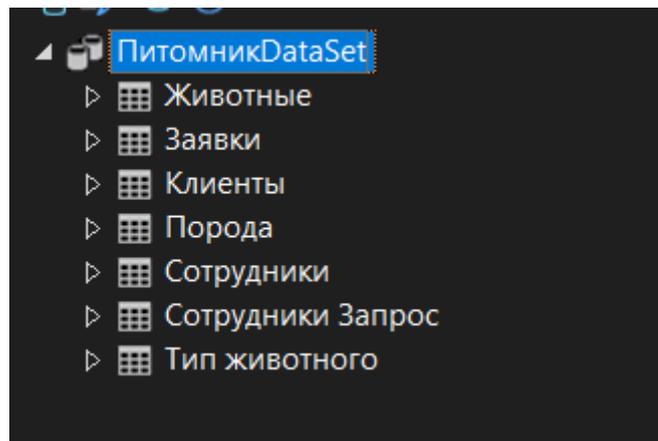


Рисунок 31 – БД в Visual Studio 2022 с подключенными таблицами

Из панели Источник данных перетаскиваем таблицу на форму (рис. 32).

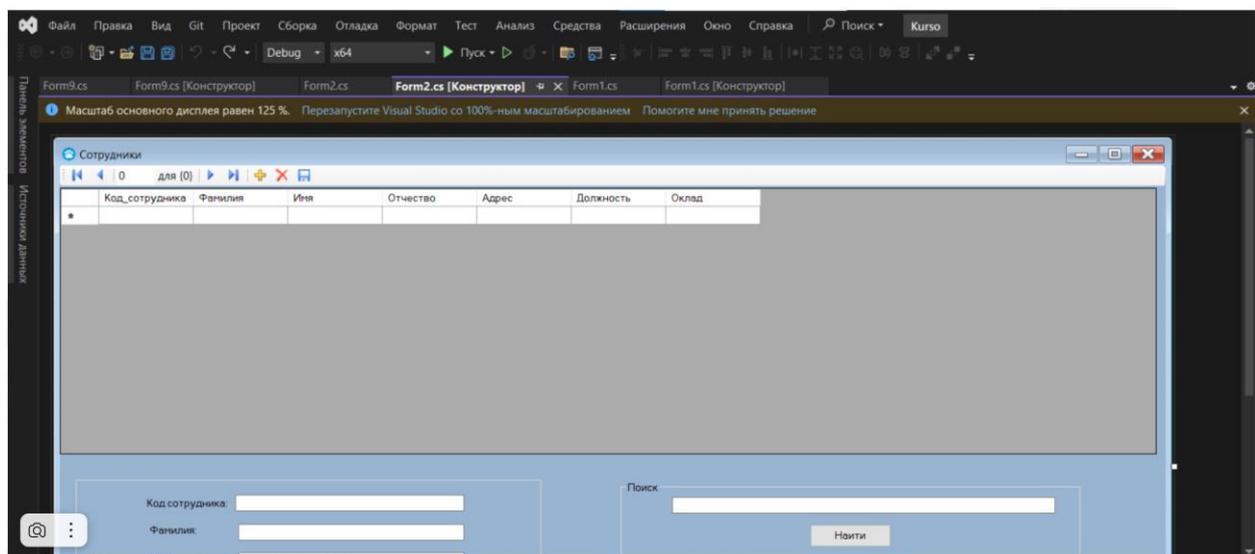


Рисунок 32 – Таблица «Сотрудники» в Visual Studio

Контрольные вопросы

1. Как создать проект в среде Visual Studio?
2. Какой формат базы данных в Access нужен для подключения в Visual Studio?
3. Опишите технологию подключения базы данных в Visual Studio.

ЛАБОРАТОРНАЯ РАБОТА № 3

Тема: Разработка пользовательского интерфейса в Visual Studio: создание форм

Цель выполнения работы. Разработать формы на основании созданных таблиц.

Для комфортной работы с базой данных приступим к разработке графического приложения. Формы позволят пользователю вносить, изменять или удалять информацию из базы данных.

Контрольное задание

Необходимо создать формы для базы данных Питомник в среде Visual Studio 2022.

Окно графического редактора Windows Forms показано на рис.33.

«Одним из преимуществ разработки в Visual Studio приложений Windows Forms является наличие графического редактора, который позволяет в графическом виде представить создаваемую форму и в принципе упрощает работу с графическими компонентами.

Для открытия формы в режиме графического дизайнера нажмем в структуре проекта на файл Form1.cs либо левой кнопкой мыши двойным кликом, либо правой кнопкой мыши и в появившемся контекстном меню выберем View Designer (также можно использовать комбинацию клавиш Shift+F7)»[4].

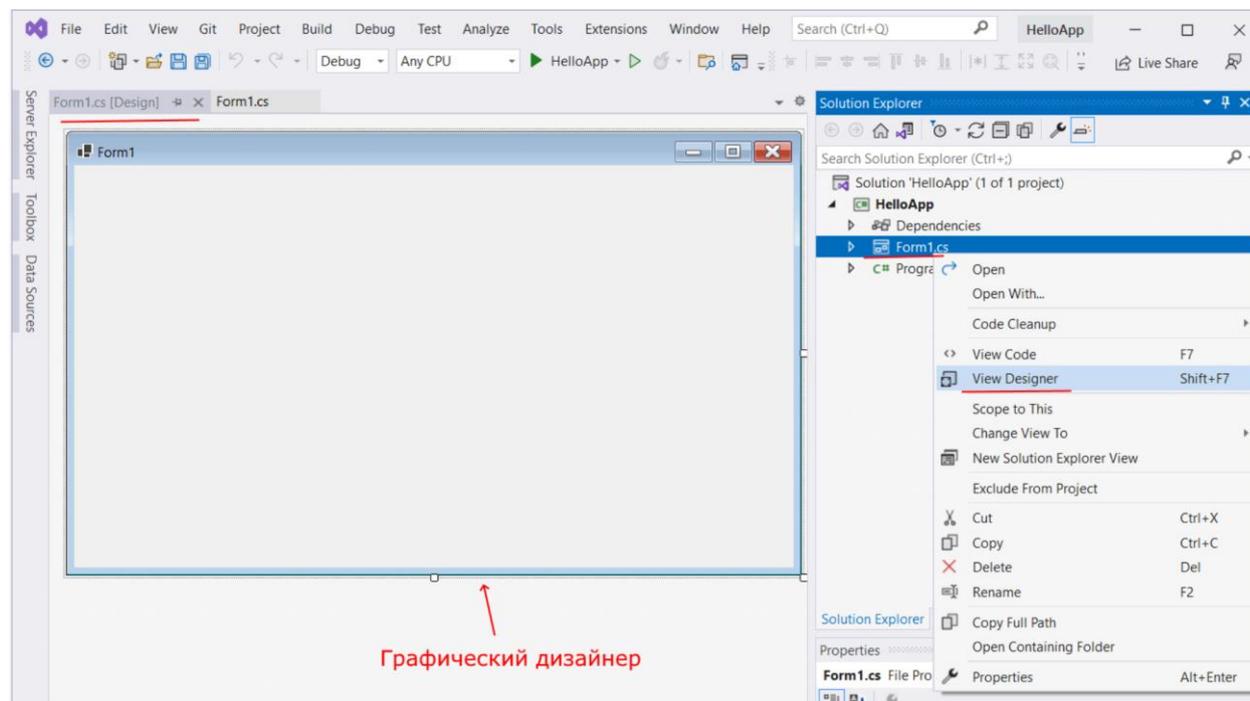


Рисунок 33 – Режим графического дизайнера

«После этого в Visual Studio откроется выбранная форма в графическом виде.

В то же время следует отметить, что на более слабых компьютерах окно дизайнера можно открываться довольно долго, а на некоторых - довольно

слабых компьютерах может и вообще не открыться. При выборе формы в окне дизайнера внизу справа под структурой проекта мы сможем найти окно Properties(Свойства)»[4](рис.34).

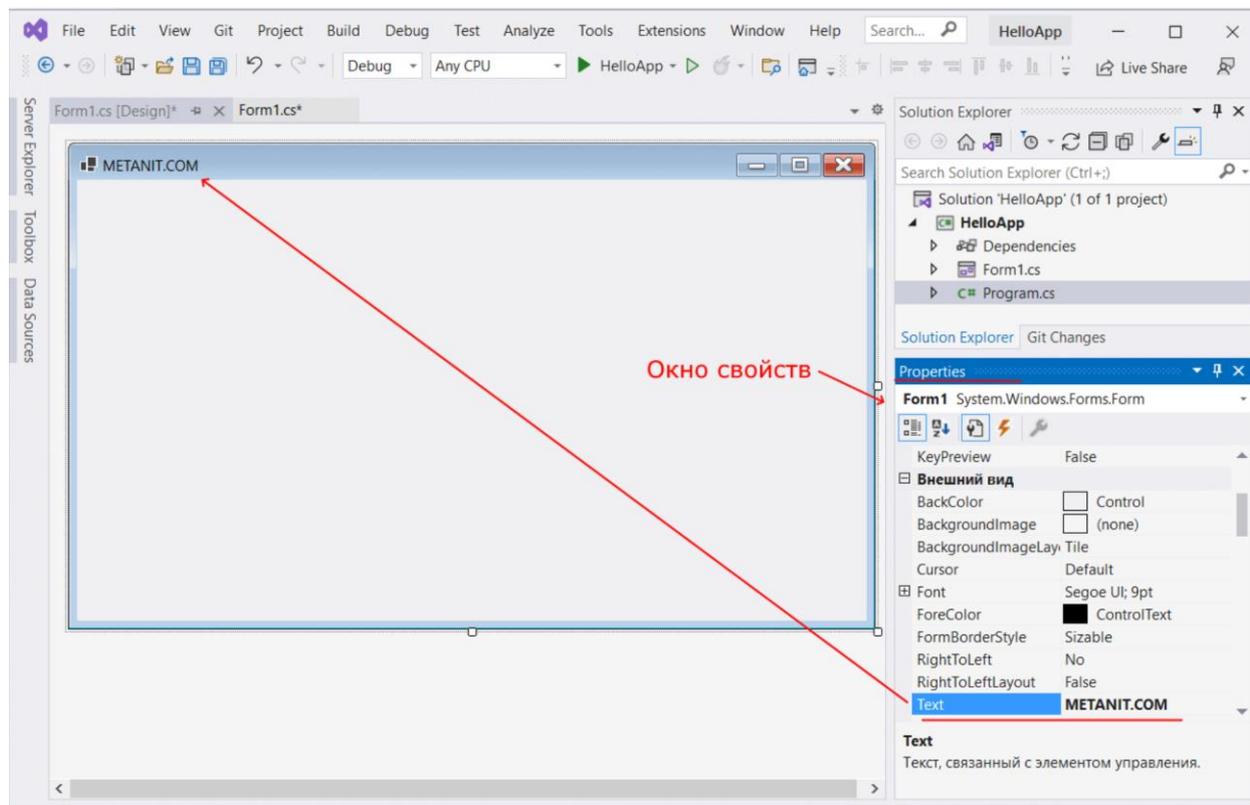


Рисунок 34 – Задание свойств для выбранного элемента форма

«Но Visual Studio имеет еще одну связанную функциональность. Она обладает панелью графических инструментов. И мы можем, вместо создания элементов управления в коде C#, просто переносить их на форму с панели инструментов с помощью мыши. Так, перенесем на форму какой-нибудь элемент управления, например, кнопку.

Для этого найдем в левой части Visual Studio вкладку Toolbox (Панель инструментов). Нажмем на эту вкладку, и у нас откроется панель с элементами, откуда мы можем с помощью мыши перенести на форму любой элемент»[4](рис.35).

«Найдем среди элементов кнопку и, захватив ее указателем мыши, перенесем на форму(рис.36).

Причем при выборе кнопки она открывается в окне свойств и, как и для всей формы, для кнопки в окне свойств мы можем изменить значения различных свойств(рис.37)»[4].

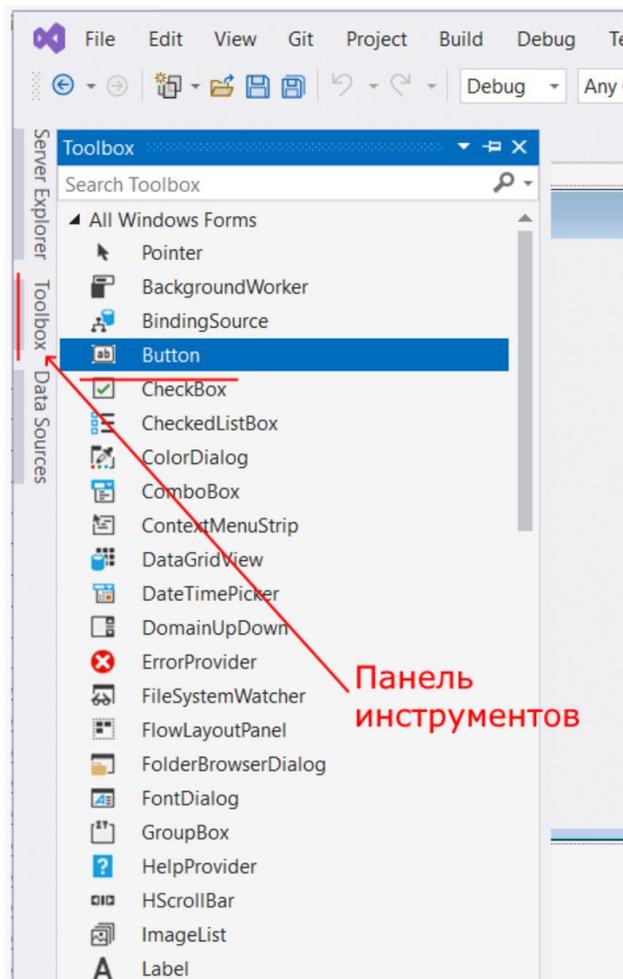


Рисунок 35 – Окно панели инструментов среды разработки

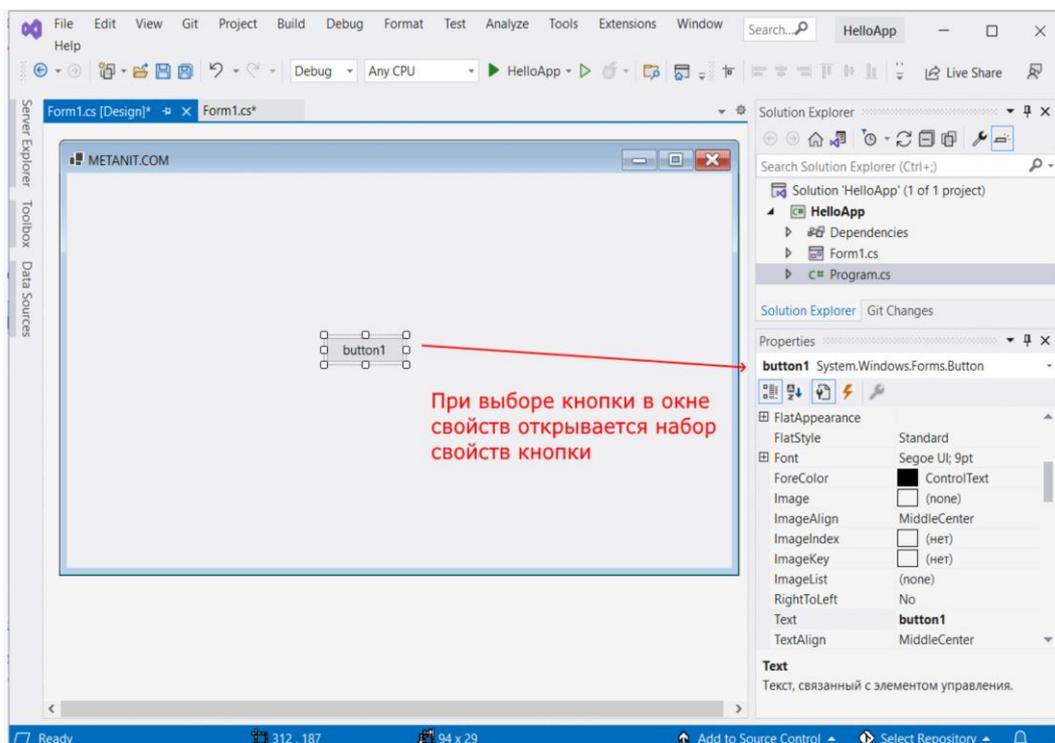


Рисунок 36 – Окно панели инструментов среды разработки

```

// button1
//
this.button1.Location = new System.Drawing.Point(312, 187);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(94, 29);
this.button1.TabIndex = 0;
this.button1.Text = "button1";
this.button1.UseVisualStyleBackColor = true;
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 20F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(800, 450);
this.Controls.Add(this.button1);
this.Name = "Form1";
this.Text = "METANIT.COM";
this.ResumeLayout(false);

```

Рисунок 37 – Код для кнопки button 1

«В класс Form1 была добавлена переменная button1 типа Button и для этой переменной, как и для объекта формы, задан ряд свойств.

И если в окне свойств мы поменяем значения этих свойств, то в этом файле также изменятся их значения. Как в случае с текстом формы:

```
1 this.Text = "METANIT.COM";
```

Это визуальная часть. Теперь приступим к самому программированию. Добавим простейший код на языке C#, который бы выводил сообщение по нажатию кнопки.

Для этого перейдем в файл кода Form1.cs, который связан с этой формой. По умолчанию после создания проекта он имеет код типа следующего»[4](рис.38).

При нажатии кнопки возникает событие **Click**. «В данном случае в конструкторе формы мы подвязываем к кнопке button1 в качестве обработчика события нажатия метод button1_Click, в котором с помощью метода **MessageBox.Show** выводит сообщение. Текст сообщения передается в метод в качестве параметра.

Стоит отметить, что графический дизайнер позволяет автоматически сгенерировать обработчик нажатия кнопки.

Для этого надо в окне дизайнера нажать на кнопку на форме двойным щелчком мыши.

Теперь запустим проект и мы увидим форму с кнопкой, на которую мы можем нажать и получить сообщение»[4](рис. 39).

```

1 namespace HelloApp
2 {
3     public partial class Form1 : Form
4     {
5         public Form1()
6         {
7             InitializeComponent();
8         }
9     }
10 }

```

Изменим этот код следующим образом:

```

1 namespace HelloApp
2 {
3     public partial class Form1 : Form
4     {
5         public Form1()
6         {
7             InitializeComponent();
8
9             button1.Click += button1_Click;
10        }
11
12        private void button1_Click(object? sender, EventArgs e)
13        {
14            MessageBox.Show("Привет");
15        }
16    }
17 }

```

Рисунок 38 – Код проекта

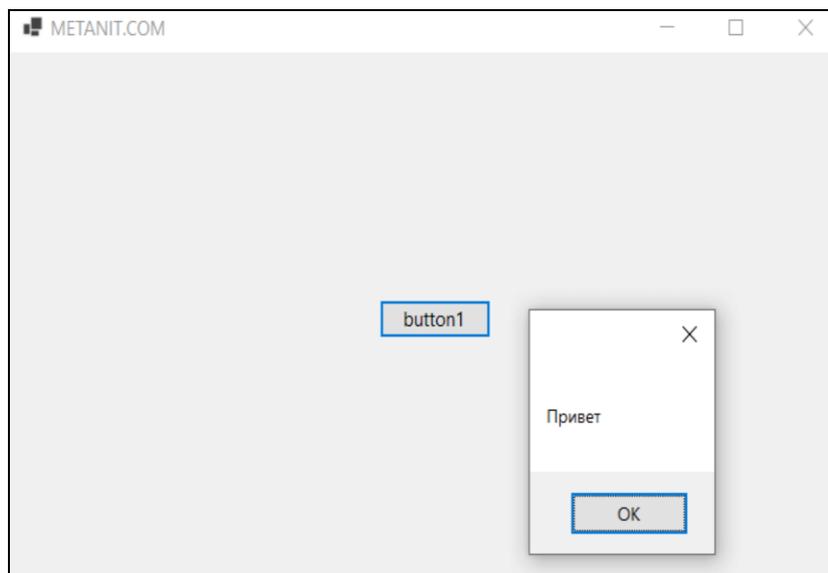


Рисунок 39 – Окно запуска проекта с событием нажатие на кнопку

«Внешний вид приложения является нам преимущественно через формы. Формы являются основными строительными блоками. Они предоставляют контейнер для различных элементов управления. А механизм событий позволяет элементам формы отзываться на ввод пользователя, и, таким образом, взаимодействовать с пользователем.

При открытии проекта в Visual Studio в графическом редакторе мы можем увидеть визуальную часть формы - ту часть, которую мы видим после запуска приложения и куда мы переносим элементы с панели управления. Но на самом деле форма скрывает мощный функционал, состоящий из методов, свойств, событий и прочее. Рассмотрим основные свойства форм»[5](рис.40).

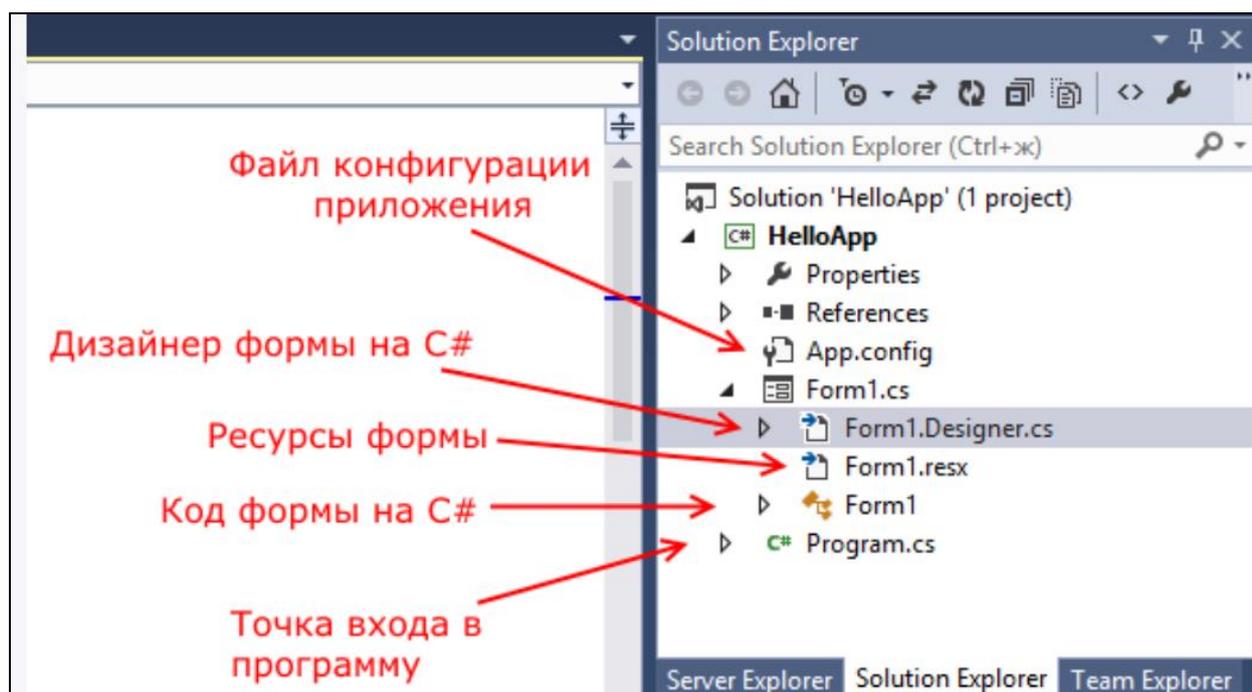


Рисунок 40 – Основные свойства формы

«Еще один файл - *Form1.resx* - хранит ресурсы формы. Как правило, ресурсы используются для создания однообразных форм сразу для нескольких языковых культур.

И более важный файл - *Form1.cs*(рис.41), который в структуре проекта называется просто *Form1*, содержит код или программную логику формы. По умолчанию здесь есть только конструктор формы, в котором просто вызывается метод *InitializeComponent()*, объявленный в файле дизайнера *Form1.Designer.cs*»[5].

Основные свойства формы представлены на сайте <https://metanit.com/sharp/windowsforms/2.2.php>.

Для выбранной темы на основании таблиц были созданы несколько форм(рис. 42-рис.44).

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace HelloApp
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19     }
20 }

```

Рисунок 41 – Код проекта Form1

«Чтобы добавить еще одну форму в проект, нажмем на имя проекта в окне Solution Explorer (Обозреватель решений) правой кнопкой мыши и выберем Add(Добавить)->Windows Form..»[5](рис.43).

Главная форма (рис. 44) содержит кнопки, открывающие созданные формы и логотип питомника в виде изображения.

Для работы с приложением была создана форма Инструкция(рис.45).

На рис. 46 показана подчиненная форма Сотрудники, которая позволяет редактировать, добавлять или удалять записи в таблице Сотрудники. На форму для перемещения по записям был добавлен компонент bindingNavigator в левом верхнем углу.

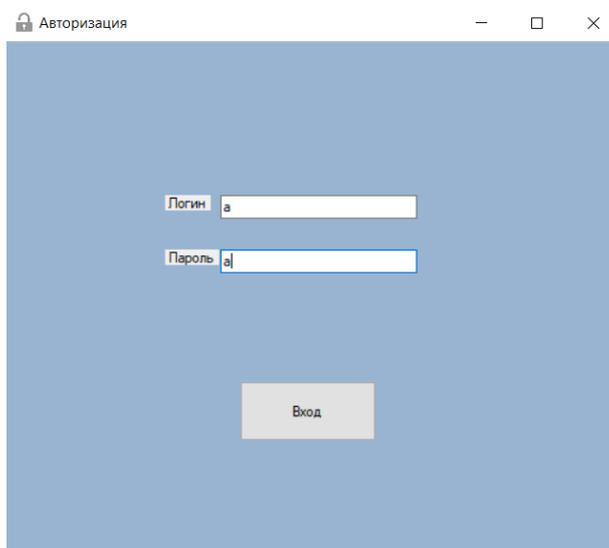
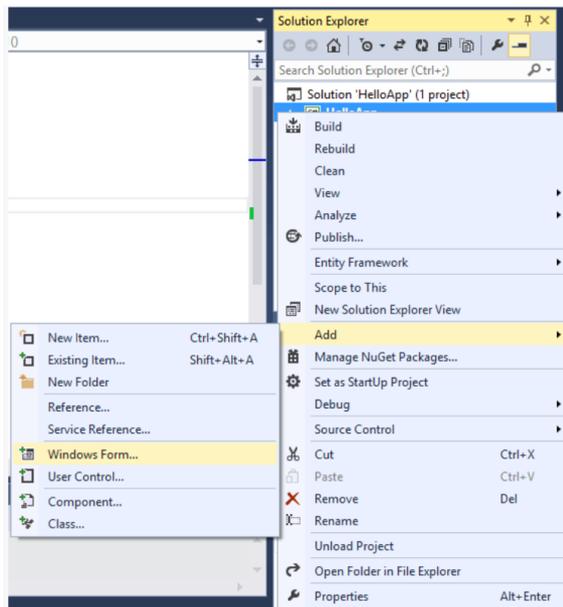


Рисунок 42 – Форма авторизация для входа в БД



Дадим новой форме какое-нибудь имя, например, *Form2.cs*:

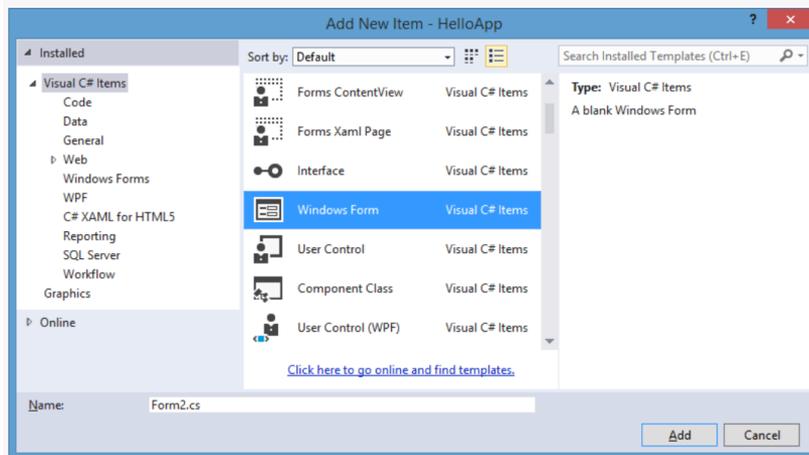


Рисунок 43 – Добавление новой формы в проект

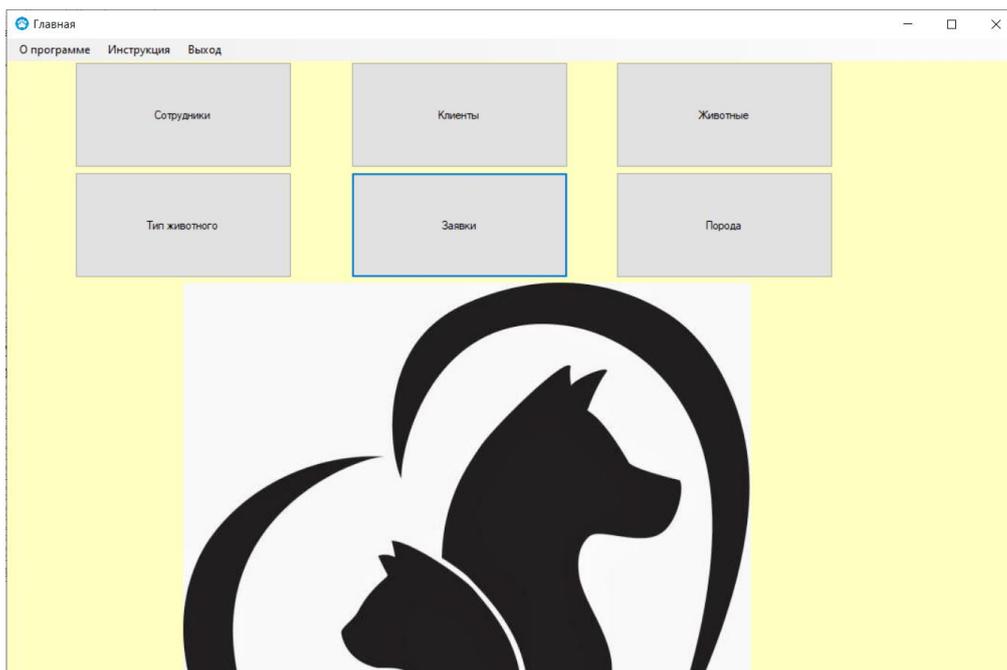


Рисунок 44 – Форма Главная

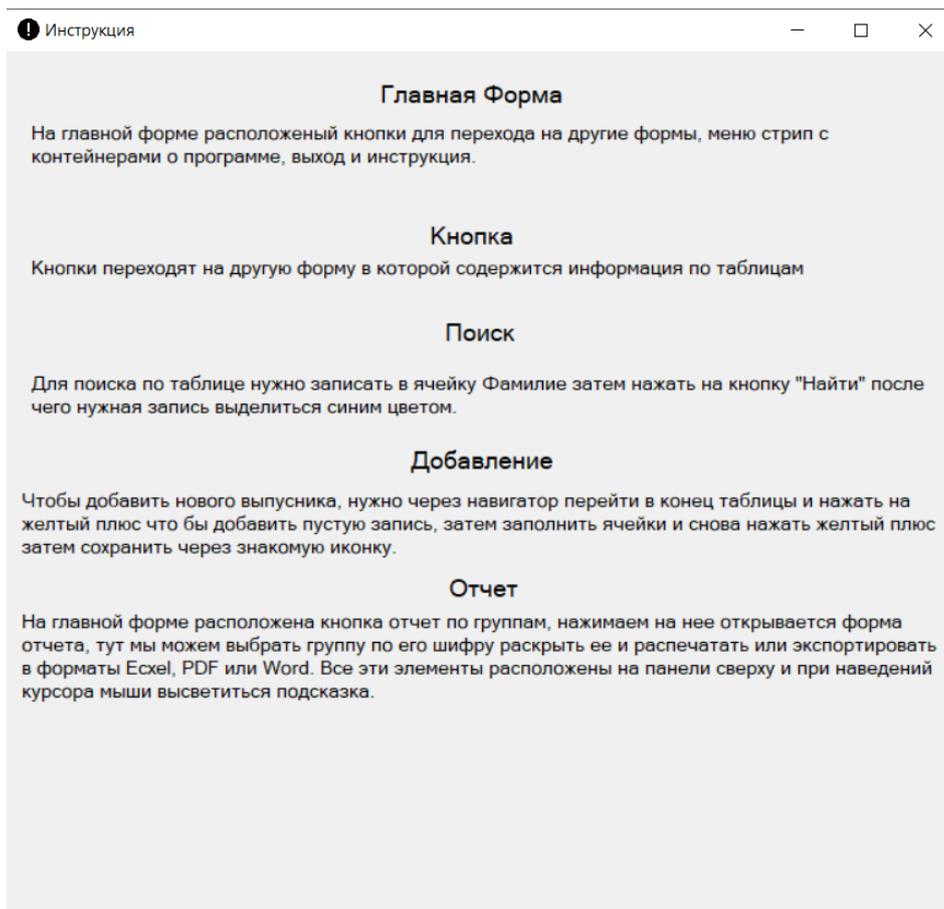


Рисунок 45 – Форма Инструкция

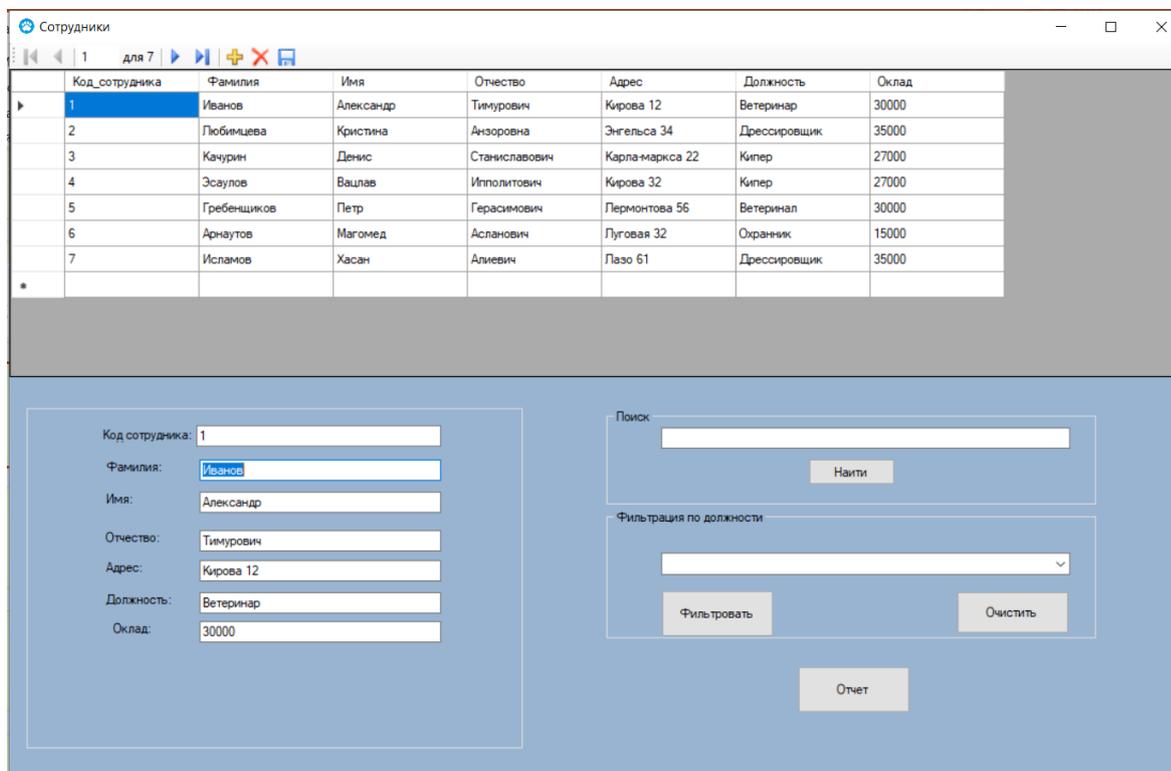


Рисунок 46 – Форма Сотрудники

Добавление новых записей происходит с помощью компонента TextBox(рис. 47). Для этого нужно перейти в конец таблицы, нажать на кноп-

ку добавить в виде кнопки плюс +, ввести информацию в TextBox, еще раз нажать на кнопку добавить.

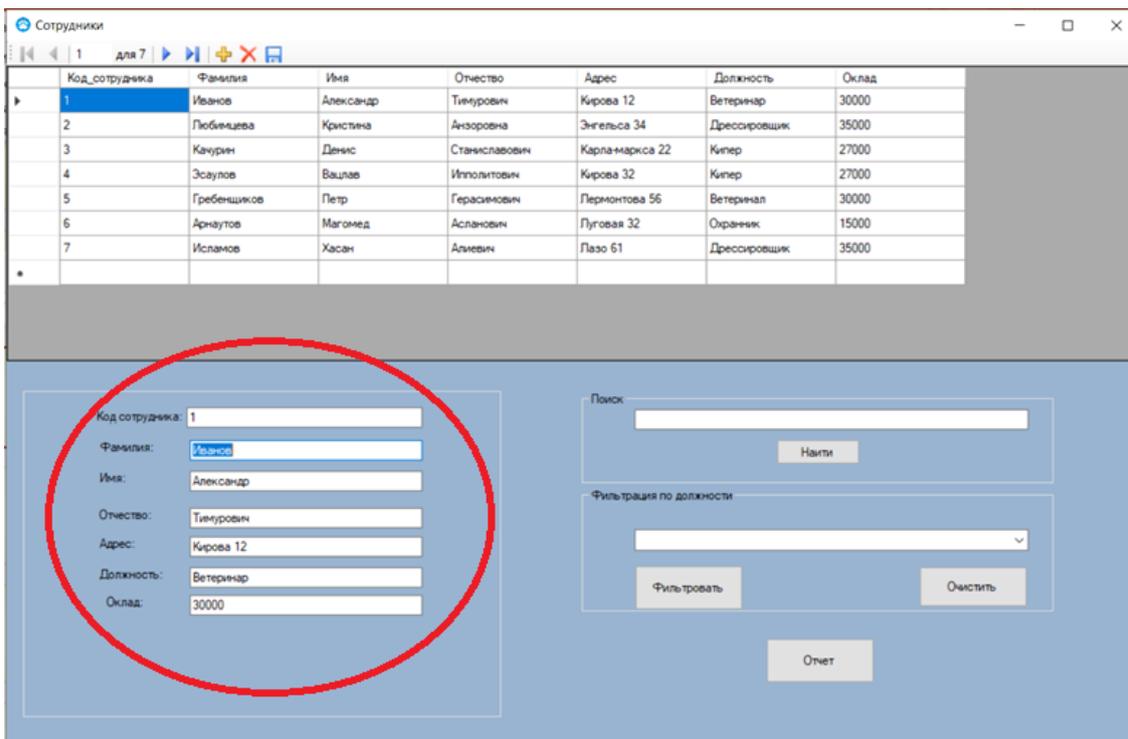


Рисунок 47 – Редактирование записей на форме Сотрудники

Индивидуальное вариантное задание

1. Для выбранной предметной области создать на каждую таблицу форму.
2. Обеспечить просмотр, модификацию данных, добавление новых записей с помощью созданных форм.

Контрольные вопросы

1. Для чего предназначены формы?
2. Какие компоненты включает панель инструментов среды?
3. Опишите процесс создания формы для редактирования данных из таблиц

ЛАБОРАТОРНАЯ РАБОТА № 4.

Тема: Поиск данных в БД.

Цель выполнения работы: научиться осуществлять поиск данных средствами программной среды.

Для осуществления поиска в БД необходимо добавить на форму следующие компоненты:

1. groupBox1.
2. textBox1.
3. button1.

В groupBox поместить элементы textBox1 и button1 и переименовываем button1 (рис. 48).



Рисунок 48 - Оформление поиска.

Далее добавляем в Кнопку «Найти» следующий код:

```
for (int i = 0; i < сотрудникиDataGridView.RowCount; i++)
{
    сотрудникиDataGridView.Rows[i].Selected = false;
    for (int j = 0; j < сотрудникиDataGridView.ColumnCount; j++)
        if (сотрудникиDataGridView.Rows[i].Cells[j].Value != null)
            if
                (сотрудникиData-
GridView.Rows[i].Cells[j].Value.ToString().Contains(textBox10.Text))
                {
                    сотрудникиDataGridView.Rows[i].Selected = true;
                    break;
                }
}
```

Данный код позволяет выделять на форме данные, соответствующие поиску (рис. 49). Поиск по таблице осуществляется с помощью TextBox.

Нужно внести информацию (имя или должность) в поле и нажать на кнопку «Найти», после чего нужная информация выделится синим цветом (рис. 49).

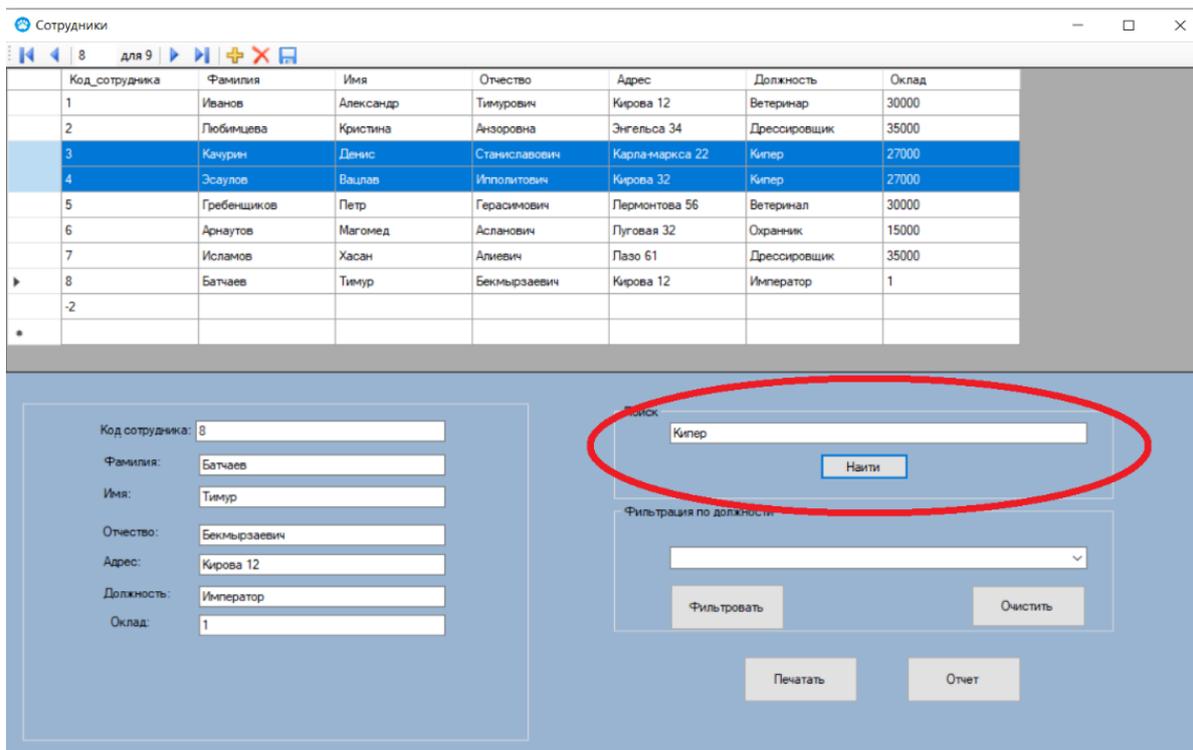


Рисунок 49– Поиск данных по должности сотрудников

Для фильтрации данных необходимо добавить следующие компоненты:

1. groupBox2
2. comboBox2
3. button2
4. button3

Также, как и в предыдущем задании, нужно в groupBox поместить элементы comboBox2, button2 и button3, переименовываем button2 и button3 (рис. 50).



Рисунок 50 – Фильтрация по должности

Для задания фильтрации из выпадающего списка (рис. 51) выбираем нужную должность, например «Дрессировщик» и нажать кнопку фильтровать, и в таблице останется должности «Дрессировщик» (рис. 52).

При нажатии кнопки очистить таблица возвращается в исходное состояние(рис. 53).

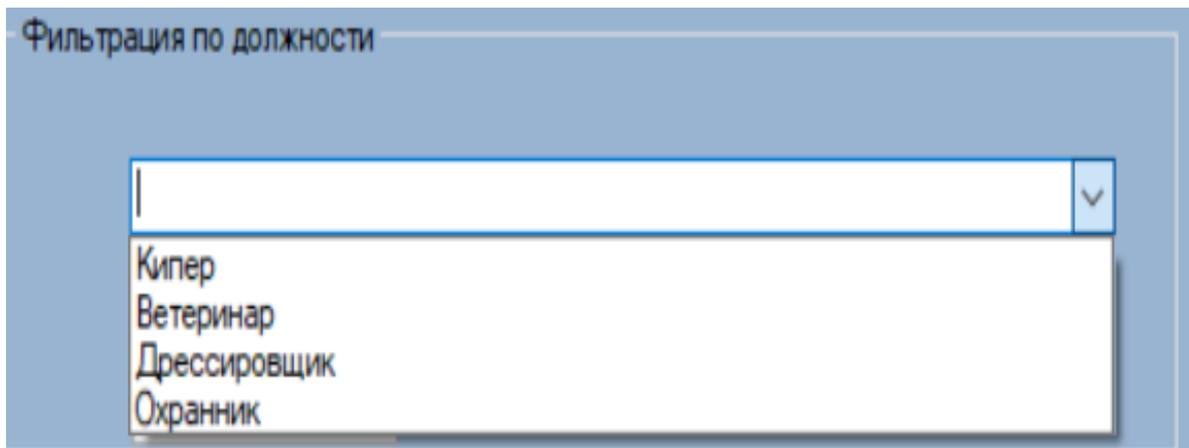


Рисунок 51– Окно выбора фильтра

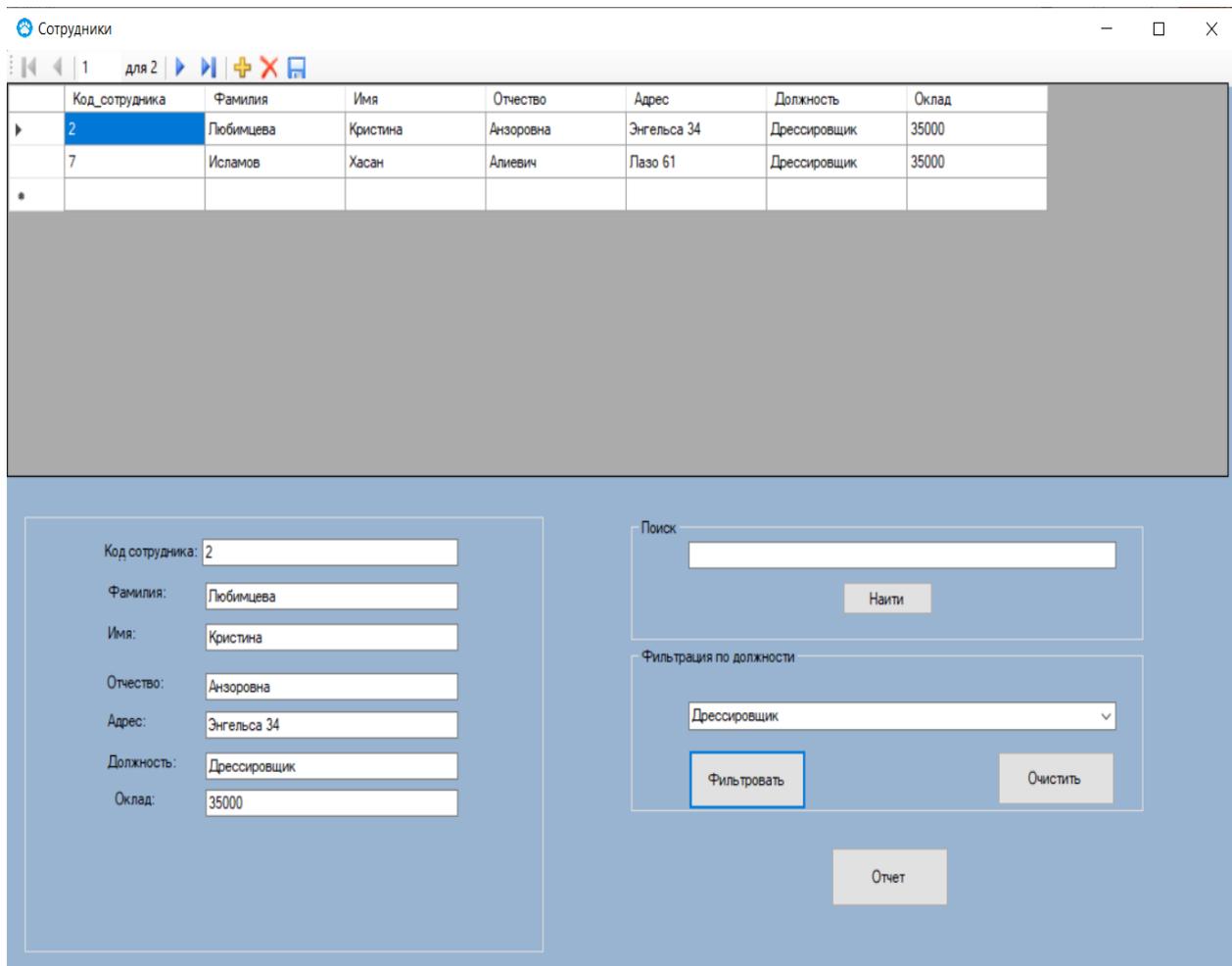


Рисунок 52– Окно результата применения фильтра

Код кнопки «Фильтровать».

сотрудникиBindingSource.Filter = "Должность=" + comboBox3.Text + """;

Код кнопки «Очистить».

сотрудникиBindingSource.Filter = "";

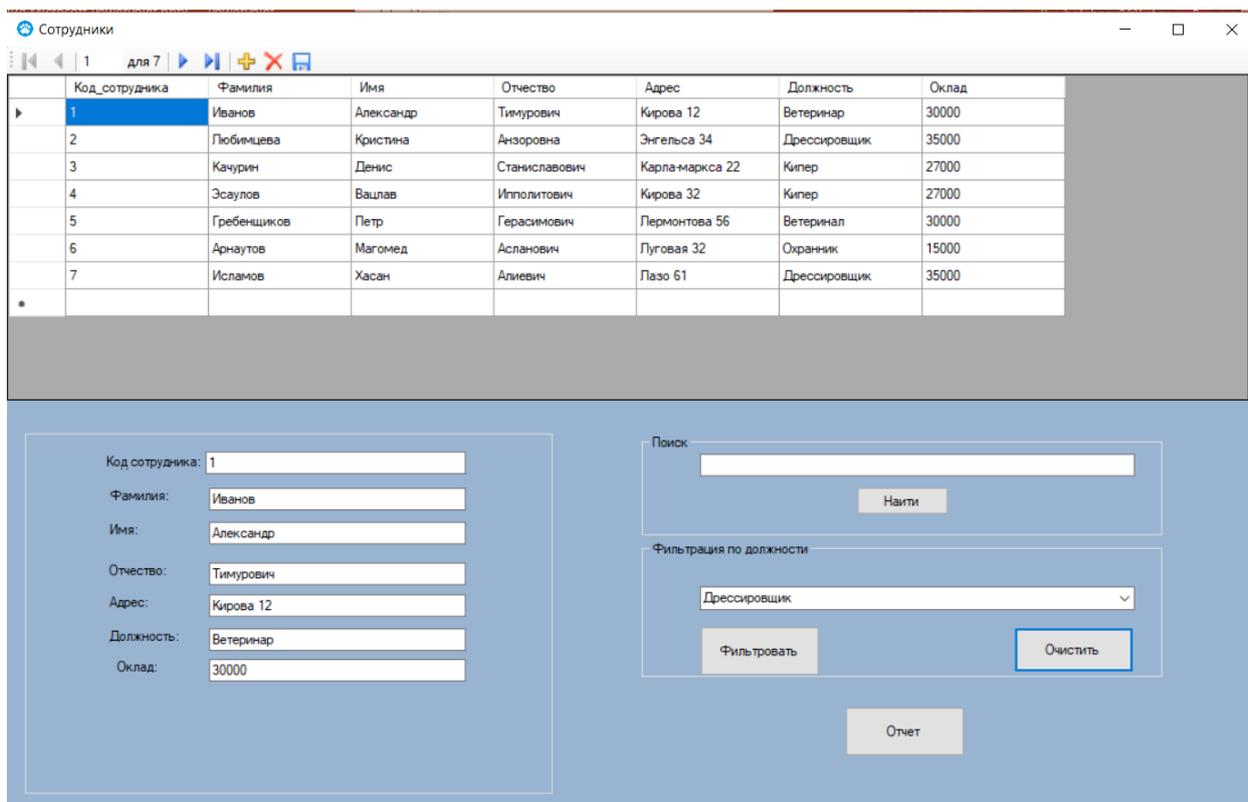


Рисунок 53 – Окно поиска сотрудников после нажатия кнопки очистить

Индивидуальное вариантное задание

Задать отбор по дате или наименованию по двум таблицам предметной области.

Контрольные вопросы

1. Какие компоненты программной среды используются для поиска данных?
2. Как осуществляется фильтрация данных?

ЛАБОРАТОРНАЯ РАБОТА № 5.

Тема: Создание отчетов по выбранным параметрам

Цель работы. Создать отчет по данным из базы.

Контрольное задание

1. Создание отчета по таблице Сотрудники.

Для создания отчета выбираем команду меню Проект--> управление пакетами NuGet(рис. 54).

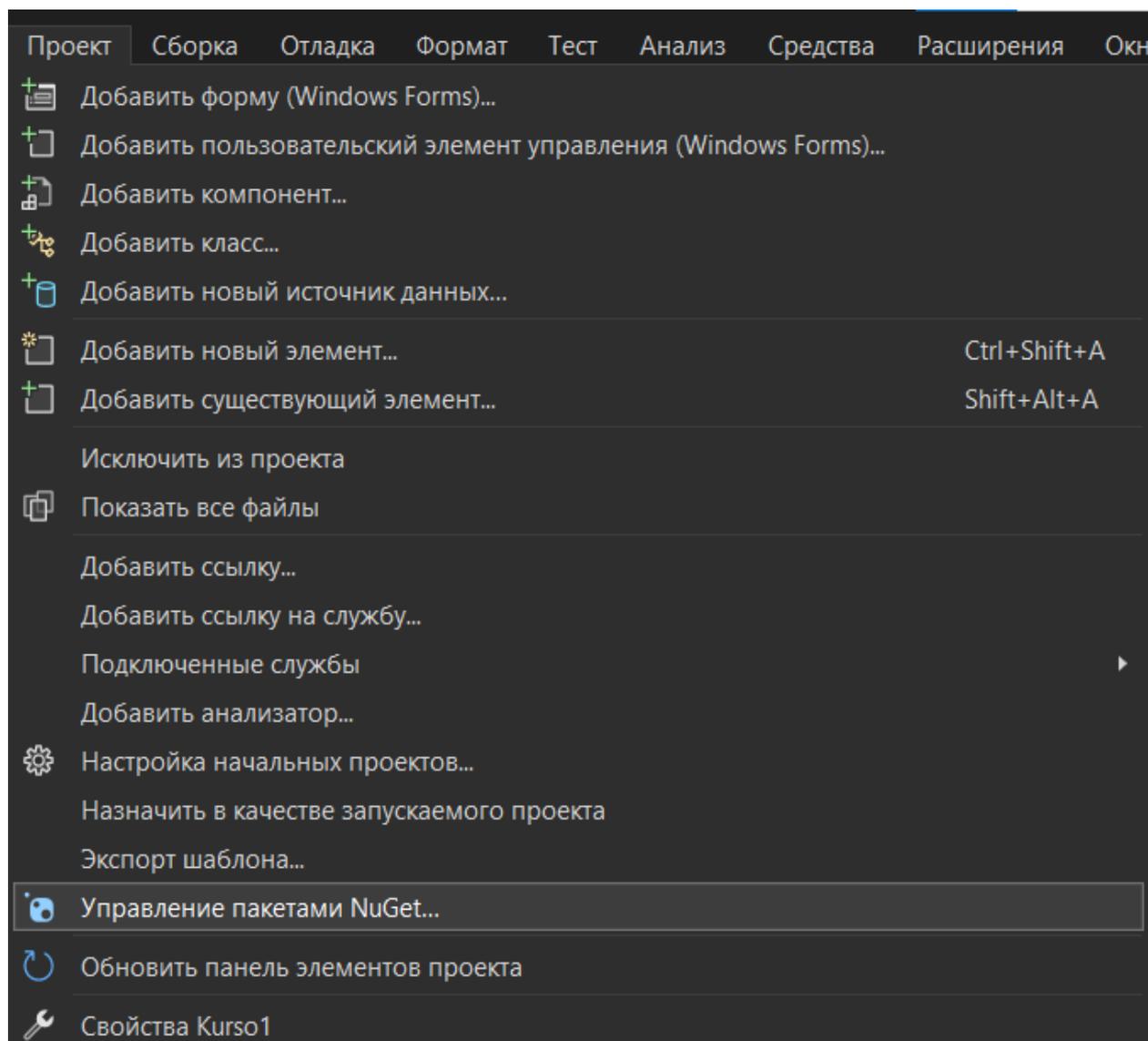


Рисунок 54 – Выбор команды для создания отчета

Основные инструменты просмотра отчетов отсутствуют как в Windows forms, так и в шаблоне приложения WPF.

Необходимо дополнительно скачать

1. Microsoft.ReportingServices.ReportViewerControl.WebForms.
2. Microsoft.SqlServer.Types(рис.55).

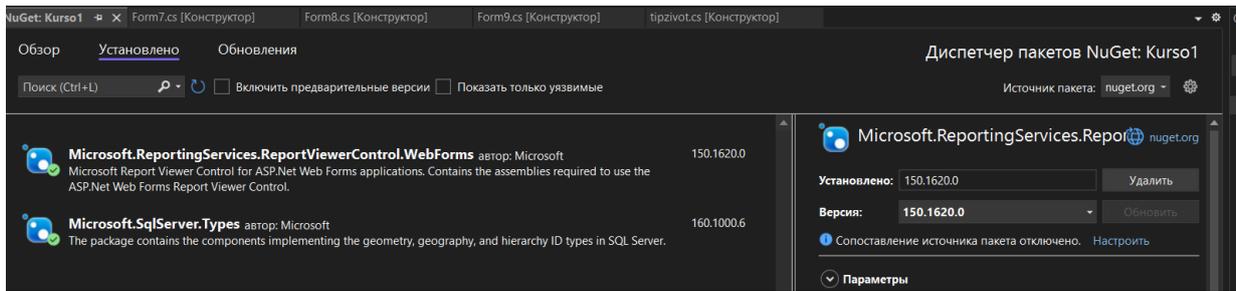


Рисунок 55 – Выбор компонентов для отчета

На пустую форму добавляем элемент ReportViewer(рис.56).

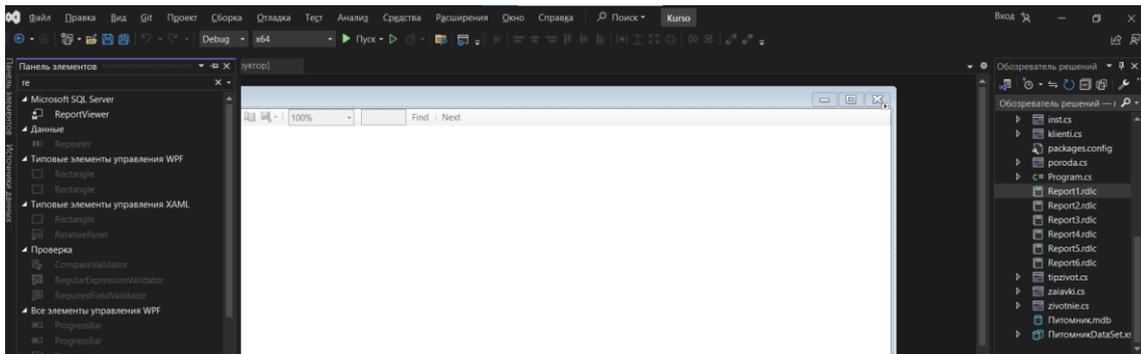


Рисунок 56 – Окно формы с компонентом отчета

Для добавления мастера отчета выбираем команду Добавление нового элемента --> Мастер отчетов (рис.57).

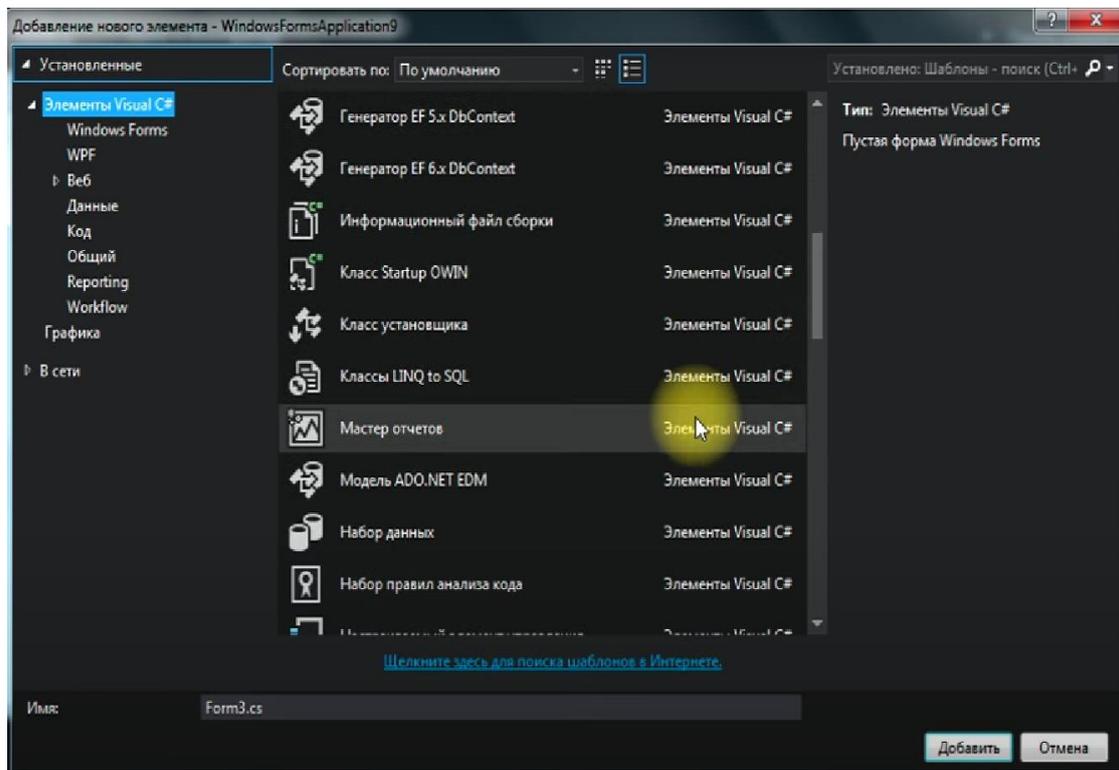


Рисунок 57 – Окно выбора нового элемента

Выбираем источник данных ПитомникDataSet и источники данных-СотрудникиЗапрос, отбираем необходимые поля(рис.58).

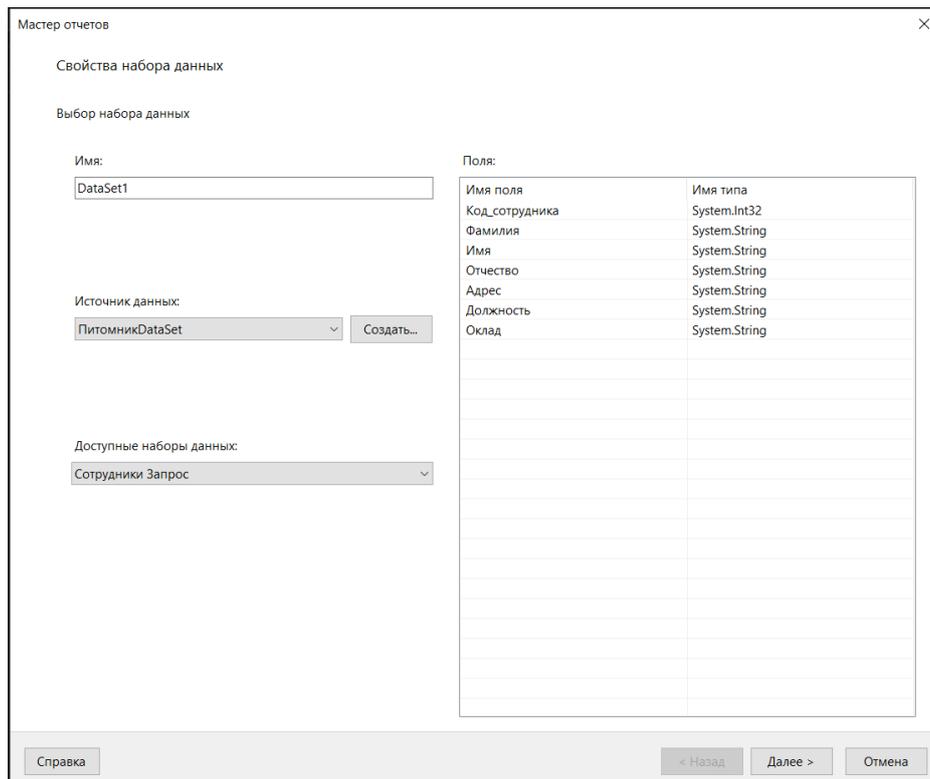


Рисунок 58 – Окно мастер отчета

Далее размещаем поля, которые будут в отчете, задаем группировки(рис.59).

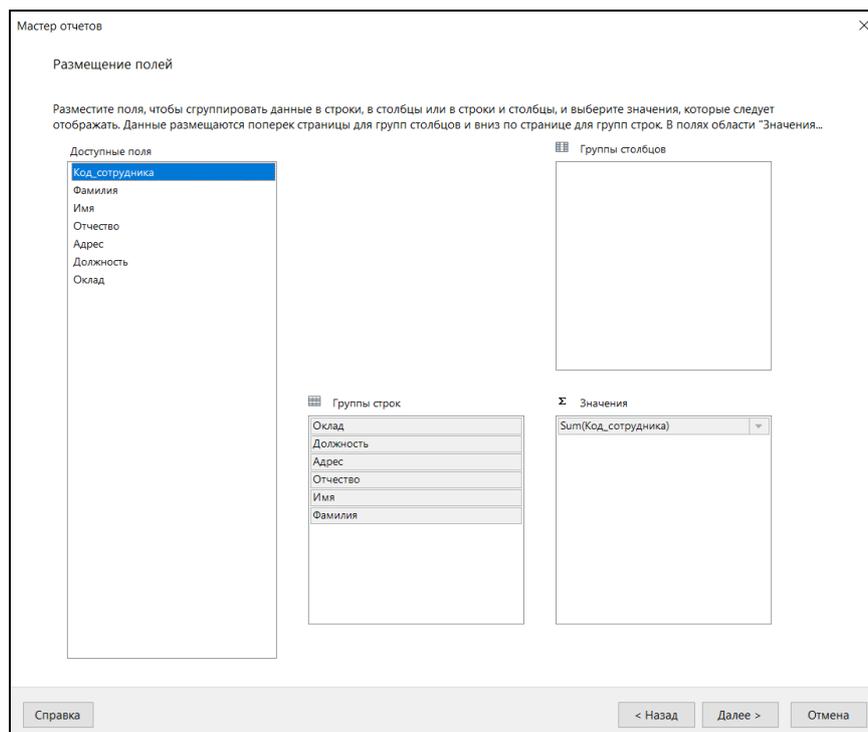


Рисунок 59 – Задание размещения полей в отчете

Остается выбрать макет отчета (рис.60).

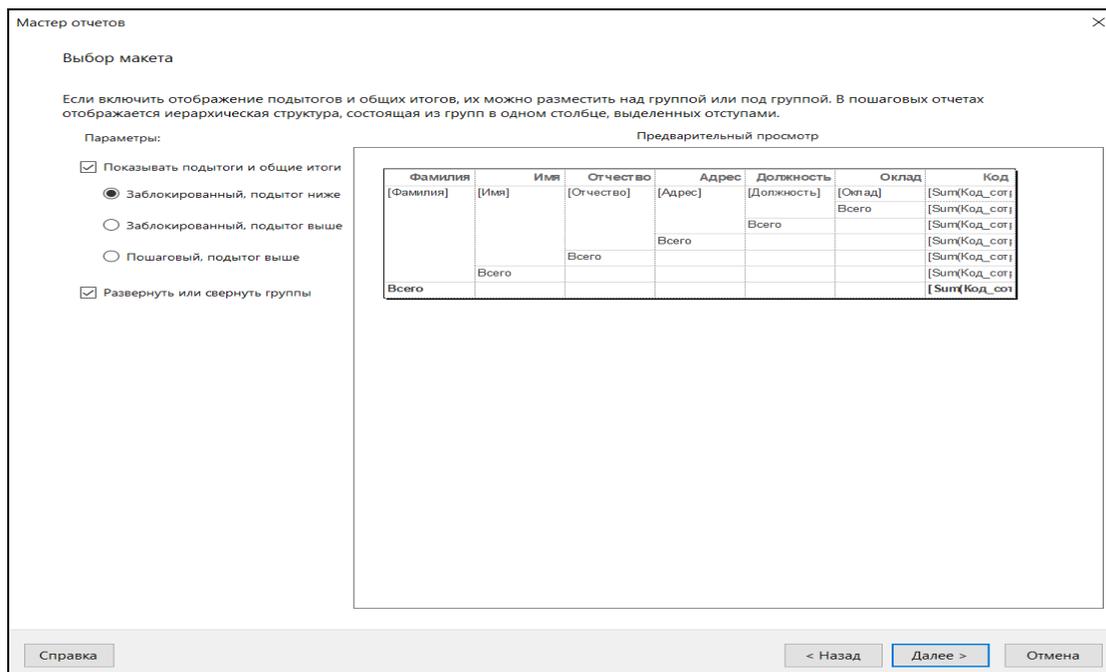


Рисунок 60 – Задание размещения полей в отчете

К элементу ReportViewer подключаем отчет, который мы получили с помощью мастера отчетов(рис.61).

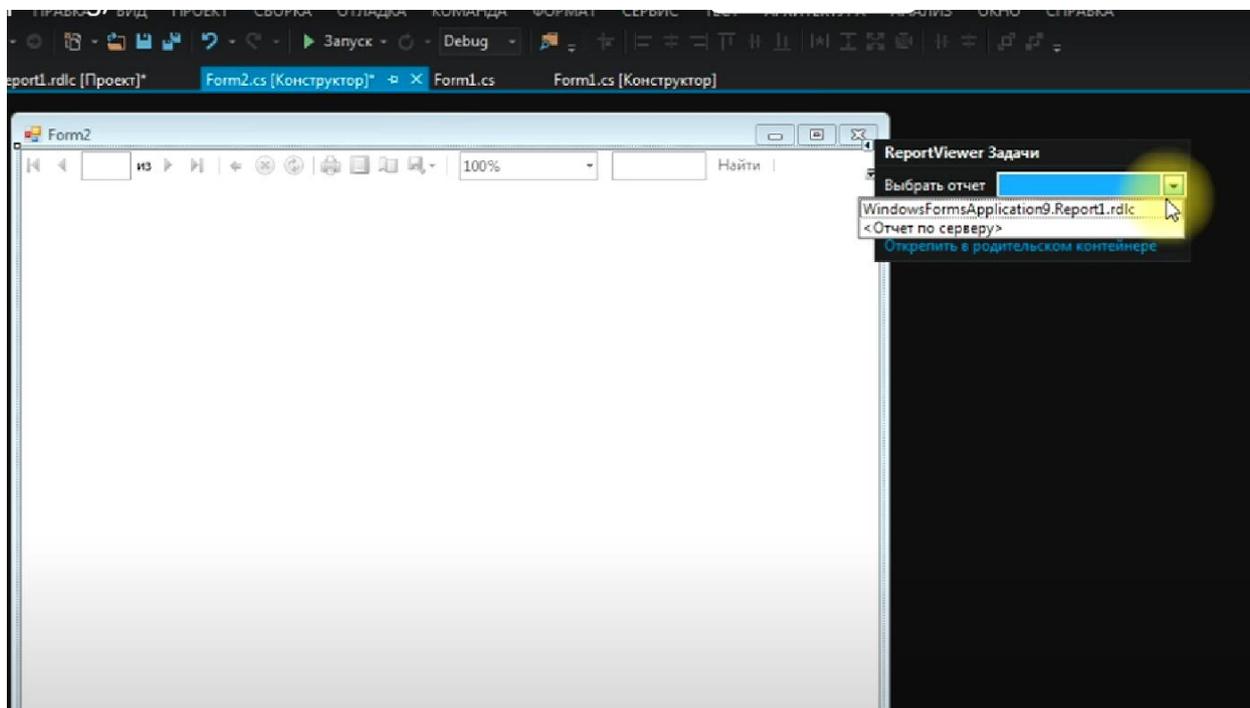


Рисунок 61 – Подключение созданного отчета на форме

Результат полученного отчета показан на рис. 62.

Фамилия	Имя	Отчество	Адрес	Должность	Оклад	Код сотрудника
Арнаутов	Магомед	Асланович	Луговая 32	Охранник	15000	6
Гребенщиков	Петр	Герасимович	Лермонтова 56	Ветеринал	30000	5
Иванов	Александр	Тимурович	Кирова 12	Ветеринар	30000	1
Исламов	Хасан	Алиевич	Лазо 61	Дрессировщи к	35000	7
Качурин	Денис	Станиславови ч	Карла-маркса 22	Кипер	27000	3
Любимцева	Кристина	Анзоровна	Энгельса 34	Дрессировщи к	35000	2
Эсаулов	Вацлав	Ипполитович	Кирова 32	Кипер	27000	4

Рисунок 62 – Готовый отчет

Можно сохранить отчет в Excel (рис. 64-65), выбрав соответствующий пункт(рис.63).



Рисунок 63 – Выбор пункта меню для сохранения в Excel

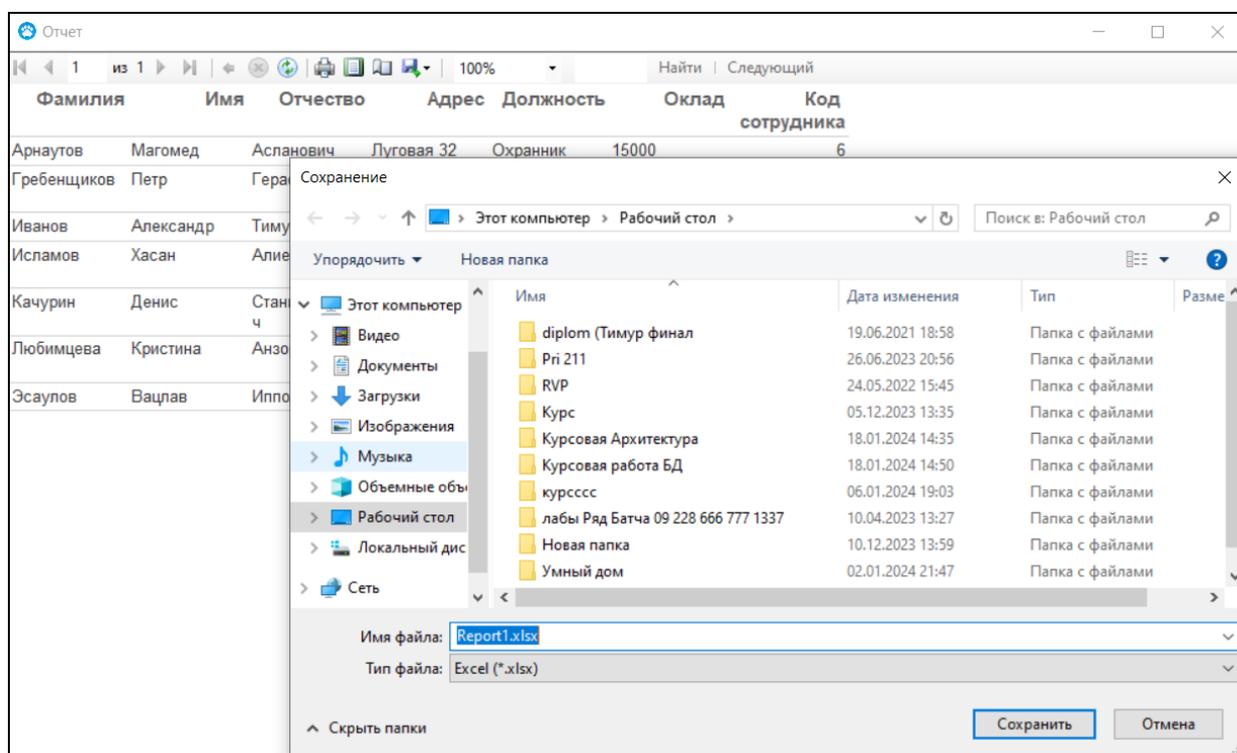


Рисунок 64 – Окно сохранения отчета в Excel

Фамилия	Имя	Отчество	Адрес	Должность	Оклад	Код сотрудника
Арнаутов	Магомед	Асланович	Луговая 32	Охранник	15000	6
Гребенщиков	Петр	Герасимович	Лермонтова 56	Ветеринал	30000	5
Иванов	Александр	Тимурович	Кирова 12	Ветеринар	30000	1
Исламов	Хасан	Алиевич	Лазо 61	Дрессировщик	35000	7
Качурин	Денис	Станиславович	Карла-маркса 22	Кипер	27000	3
Лобимцева	Кристина	Анзоровна	Энгельса 34	Дрессировщик	35000	2
Эсвулов	Вацлав	Ипполитович	Кирова 32	Кипер	27000	4

Рисунок 65 – Выгруженный в Excel отчет

По аналогии можно создать разные виды отчетов на основании выбранных источников данных. В результате в данном пособии приведены примеры подключения базы данных в Visual Studio, создания форм, поиска данных, создания отчетов.

Индивидуальное вариантное задание

Создать отчеты по выбранной предметной области.

Контрольные вопросы

1. Для чего предназначен пакет NuGet?
2. Какие компоненты необходимо скачать для создания отчета?
3. Опишите процесс создания отчета.

Список использованных источников

1. Благодаров А.В. Клиент-серверные приложения баз данных: учеб. пособие / А.В. Благодаров, Н.Н. Гринченко, А.Ю. Громов; Рязан. гос. Ун-т. Рязань, 2017.-72с.
2. Евдокимов П.В. С# НА ПРИМЕРАХ — СПб.: Наука и Техника, 2016. — 304 с.
3. Первое приложение в Visual Studio:<https://metanit.com/sharp/windowsforms/1.1.php>
4. Создание графического приложения:<https://metanit.com/sharp/windowsforms/1.2.php>
5. Работа с формами:<https://metanit.com/sharp/windowsforms/2.1.php>

МОРОЗОВА Надежда Витальевна
БАТЧАЕВА Зурида Борисовна

БАЗЫ ДАННЫХ

Практикум для обучающихся 3 курса
по направлению подготовки 09.03.04 Программная инженерия

Корректор Чагова О. Х.
Редактор Чагова О.Х.

Сдано в набор 23.08.2024 г.
Формат 60x84/16
Бумага офсетная
Печать офсетная
Усл. печ. л.3,02
Заказ №4619
Тираж 100 экз.

Оригинал-макет подготовлен
в Библиотечно-издательском центре СКГА
369000, г. Черкесск, ул. Ставропольская, 36

