ГОУ ВПО «Донецкий национальный университет»

Физико-технический факультет

Кафедра компьютерных технологий

Отчет

об учебной практике

с «07» июня 2021г. по «19» июня 2021г.

ВЫПОЛНИЛ:

Студент II курса, группы ИВТ-4

Очной (заочной) формы обучения

Направление подготовки 09.03.01

«Информатика и вычислительная техника»

Подлужный Станислав Олегович

ПРОВЕРИЛА:

Ст. преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бодряга В. Е.

Донецк 2021

ОГЛАВЛЕНИЕ

[ВАРИАНТ ЗАДАНИЯ 3](#_Toc74752708)

[1. РЕАЛИЗАЦИЯ ЗАДАНИЯ 4](#_Toc74752709)

[1.1 Модель базы данных. 4](#_Toc74752710)

[1.2 Реализация базы данных в Access. 6](#_Toc74752711)

[1.3 Выполнение заданий по варианту, описание контрольных примеров. 9](#_Toc74752712)

[2. РАЕЛИЗАЦИИ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРИЛОЖЕНИЯ 13](#_Toc74752713)

[2.1 Выбор средств реализации 13](#_Toc74752714)

[2.2 Выполнение заданий по варианту, описание контрольных примеров 13](#_Toc74752715)

[ЗАКЛЮЧЕНИЕ 19](#_Toc74752716)

[СПИСОК ЛИТЕРАТУРЫ 20](#_Toc74752717)

# ВАРИАНТ ЗАДАНИЯ

Выполнить вариант задания с использованием базы данных и необходимых запросов, созданных в оболочке Microsoft Access.

**Вариант 16. Данные по штатам Австралии.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Регион** | **Штат** | **Площадь тыс. км^2** | **Население тыс. чел.** |
| Южный | Новый Южный Уэльс | 801,4 | 4382,4 |
| Южный | Виктория | 277,6 | 3324,2 |
| Северный | Квисленд | 1726,7 | 1732,4 |
| Южный | Южная Австралия | 984,4 | 1125,4 |
| Западный | Западная Австралия | 2527,6 | 904,4 |
| Южный | Тасмания | 68,4 | 382,0 |

**Задание 1.** Площадь какого штата минимальная.

**Задание 2.** Население каких штатов выше 3000 тыс. чел.

**Задание 3.** Выдать информацию о штатах, площадью более 500 тыс. км^2.

# РЕАЛИЗАЦИЯ ЗАДАНИЯ

## 1.1 Модель базы данных.

Существует два вида представления модели - логический и физический.

**Физическая модель данных** — это модель базы данных, описанная с помощью средств конкретной СУБД. Физическая модель строится на базе даталогической путем добавления особенностей конкретной СУБД.

**Логическая модель данных** — это модель базы данных, выраженная в понятиях модели данных. Этим отличается от концептуальной модели, описывающей семантику предметной области без указания технологии (конкретных методов реализации), и от физической модели, которая описывает конкретные физические механизмы, применяемые для хранения данных в накопителях.

На логической модели данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире, например "Ученик" или "Класс". Объекты модели, представляемые на логическом уровне, называются сущностями и атрибутами. Логическая модель данных может быть построена на основе другой логической модели, например на основе модели процессов .

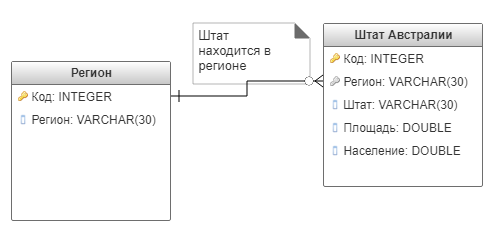
Физическая модель данных, напротив, зависит от конкретной СУБД, фактически являясь отображением системного каталога. В физической модели содержится информация о всех объектах БД. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД.

Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах - таблицах, колонках, индексах, процедурах и т. д. Разделение модели данных на логические и физические позволяет решить несколько важных задач.

Логическая модель данных является универсальной и никак не связана с конкретной реализацией СУБД.

Процесс построения логической модели базы данных должен опираться на определённую модель данных (реляционная, сетевая, иерархическая), которая определяется типом предполагаемой для реализации информационной системы СУБД.

В нашем случае база данных создается в среде Microsoft Access и будет представлять собой реляционную базу данных.



## 1.2 Реализация базы данных в Access.

В реляционных моделях данных объекты и взаимосвязи между ними представляются с помощью таблиц. Каждая таблица представляет один объект и состоит из строк и столбцов. Таблица в реляционной модели называется отношением.

В таблице 1.1 представлены названия регионов

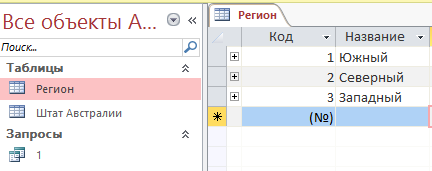


Таблица 1.1 - Регионы

В таблице 1.2 перечислена информация о штатах Австралии

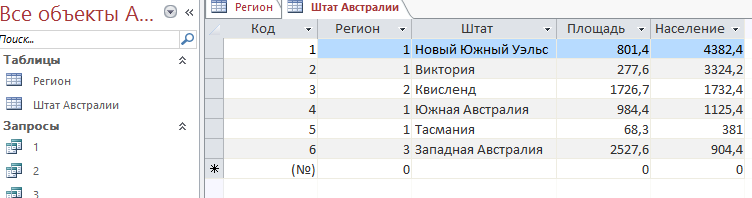


Таблица 1.2 – Штат Австралии

**Задание 1.** Площадь какого штата минимальная.

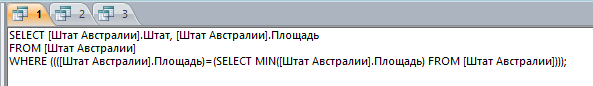


Рисунок 1.3 – Запрос 1

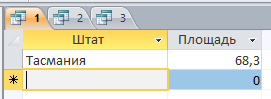


Рисунок 1.4 – Результат запроса 1

**Задание 2.** Население каких штатов выше 3000 тыс. чел.

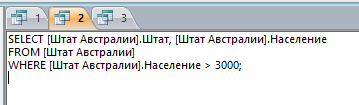


Рисунок 1.5 – Запрос 2

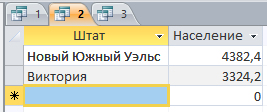


Рисунок 1.6 – Результат запроса 2

**Задание 3.** Выдать информацию о штатах, площадью более 500 тыс. км^2.

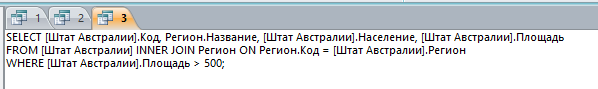


Рисунок 1.7 – Запрос 3

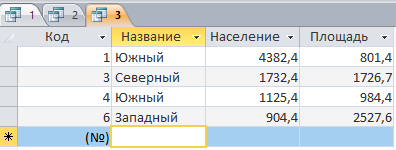


Рисунок 1.8 – Результат запроса 3

Для того чтобы установить связи между таблицами, необходимо установить связи между теми полями, в которых содержится общая информация.

Ключевое поле — это одно или несколько полей, комбинация значений которых однозначно определяет каждую запись в таблице.   
Для сформированных таблиц с установленным ключевым полем в каждой из них возможно создание определенных взаимоотношений. Access использует эти взаимоотношения для связывания данных в каждом новом запросе, форме или отчете, включающем связанные таблицы.

Если для таблицы определены ключевые поля, то Microsoft Access предотвращает дублирование или ввод пустых значений в ключевое поле.

В типе связей один-ко-многим одной записи первой таблицы соответствует несколько записей в другой таблице. В нашем случае будет именно эта связь.

Всего имеем две таблицы, имеющие такую схемой данных и ключевыми полями(записями):

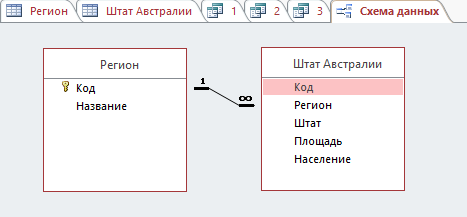


Рисунок 1.9 – Блок-схема

## Выполнение заданий по варианту, описание контрольных примеров.

Форма в Access — это объект базы данных, который можно использовать для создания пользовательского интерфейса для приложения базы данных. "Связанная" форма напрямую связана с источником данных, например таблицей или запросом, и может использоваться для ввода, редактирования или отображения данных из этого источника. Кроме того, можно создать "несвязанную" форму. Она не привязывается непосредственно к источнику данных, но также содержит кнопки команд, метки или другие элементы управления, которые необходимы для работы с приложением.

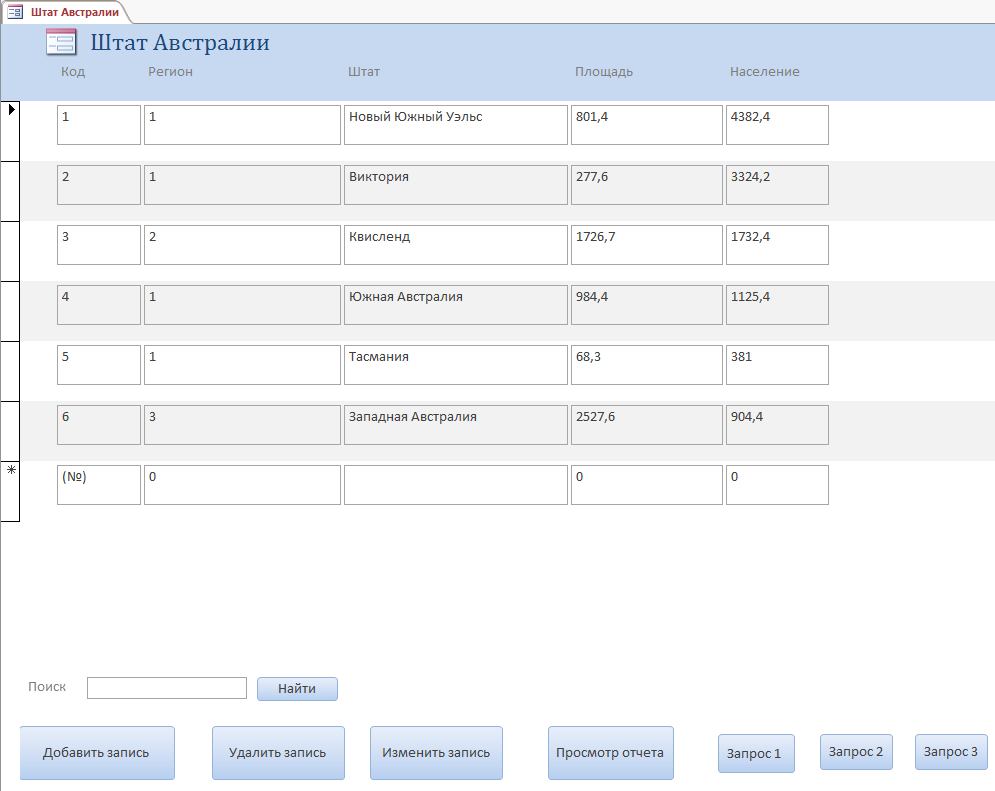


Рисунок 1.10 – Форма «Штат Австралии»

При нажатии на кнопу «Добавить запись» пользователю будет предложены поля для заполнения информации.

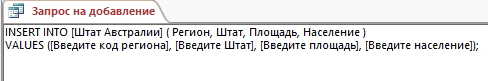


Рисунок 1.11 – Запрос на добавление записи

При нажатии на кнопу «Удалить запись» пользователю будет предложено поле для написания кода удаляемой записи.

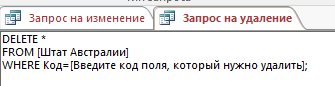


Рисунок 1.12 – Запрос на удаление записи

При нажатии на кнопу «Изменить запись» пользователю будет предложены поля для заполнения информации и также поле для выбора изменяемой записи.

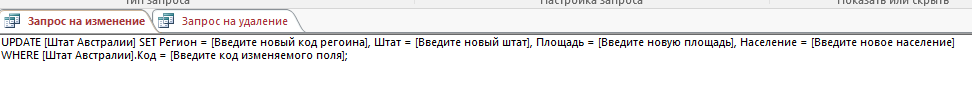


Рисунок 1.13 – Запрос на изменение записи

При нажатии на кнопки «Запрос 1» «Запрос 2» «Запрос 3» будут вызваны запросы 1,2,3 в соответствии.

Поле для поиска при нажатии на кнопку выведет все штаты, код региона которого вы ввели.

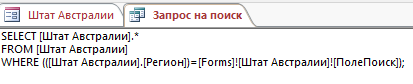


Рисунок 1.14 – Запрос на поиск

При нажатии на кнопку «Отчет» пользователю будет предоставлен отчет по БД.

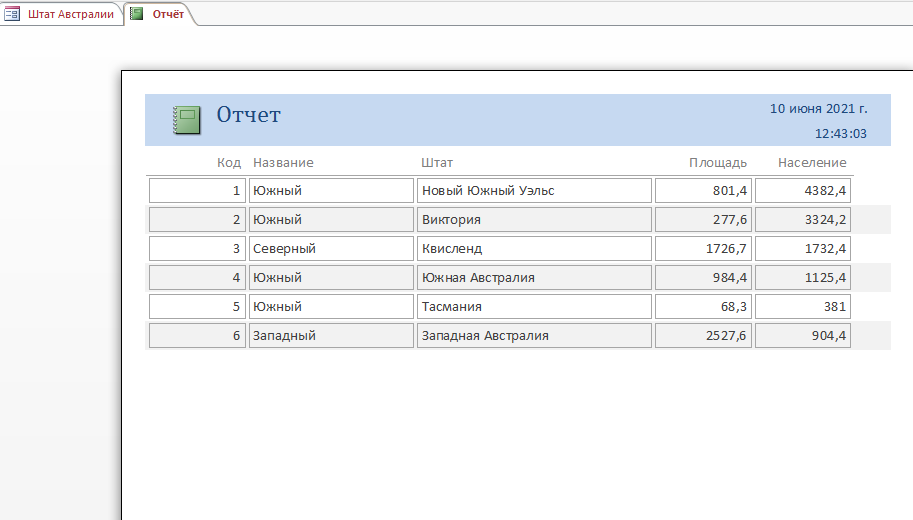


Рисунок 1.15 - Отчет

# РАЕЛИЗАЦИИ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРИЛОЖЕНИЯ

## 2.1 Выбор средств реализации

Для разработки приложения была выбрана технология Microsoft .NET Framework 4.7, обладающая огромным функционалом для графического представления данных и построения окон.

Для создания GUI (графического интерфейса пользователя) был выбран интерфейс программирования приложений Windows Forms, являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде.

В качестве инструментальной среды разработки (IDE) была выбрана Microsoft Visual Studio 2019 Community и язык C#, который является языком разработки приложений для платформы Microsoft .NET Framework.

Для написания кода была выбрана парадигма «ООП» (объектно- ориентированное программирование), так как C# − это объектно- ориентированный язык и каждое окно формы является экземпляром класса.

Для хранения и получения информации была выбрана библиотека CsvHelper. Библиотека CsvHelper – быстрый и простой в использовании способов чтения и записи пользовательских объектов в csv-файлы

## 2.2 Выполнение заданий по варианту, описание контрольных примеров

При выполнении задания было составлено на объектно-ориентированном языке приложение, которое работает с файлами формата csv.

На рисунке 2.1 представлена общая структура проекта.

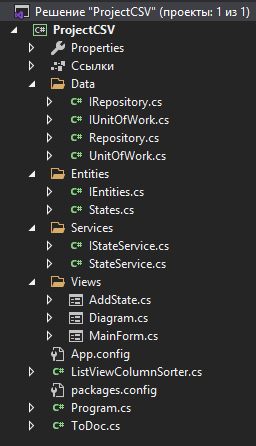


Рисунок 2.1 – Структура проекта

При составлении программы было разработано меню (рис 2.2), содержащие следующие функции:

1. Печать исходного файла.
2. Добавление записей в файл.
3. Корректировка записей файла.
4. Удаление нескольких записей файла.
5. Упорядочение записей.
6. Поиск записей.
7. Печать результирующего файла.
8. Вывод графической информации.

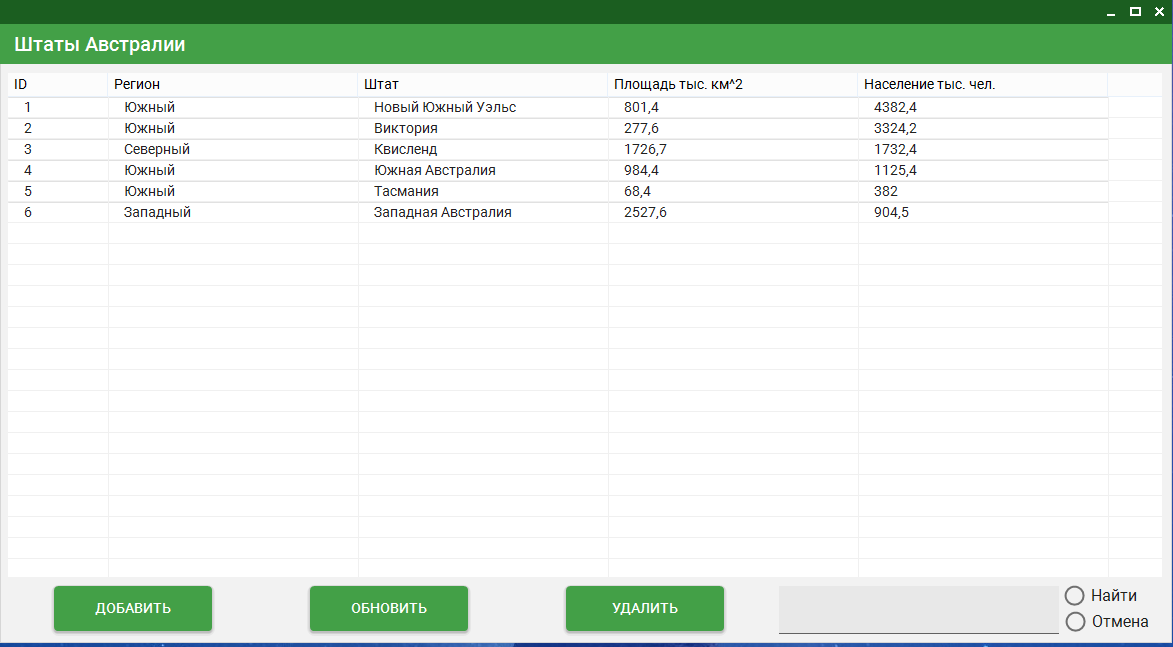


Рисунок 2.2 – Главное окно приложение

При нажатии на кнопку «Добавить» пользователь будет перемещен на форму для заполнения информации.

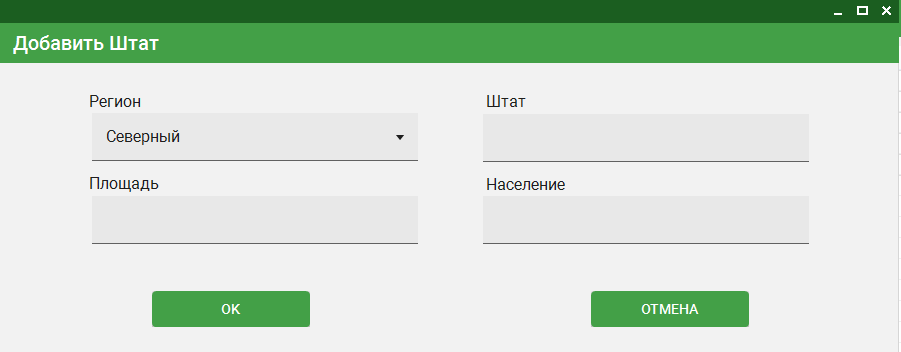


Рисунок 2.3 – Окно добавления штата

При нажатии на кнопку «Обновить» пользователь будет перемещен на форму для корректировки информации выбранной записи.

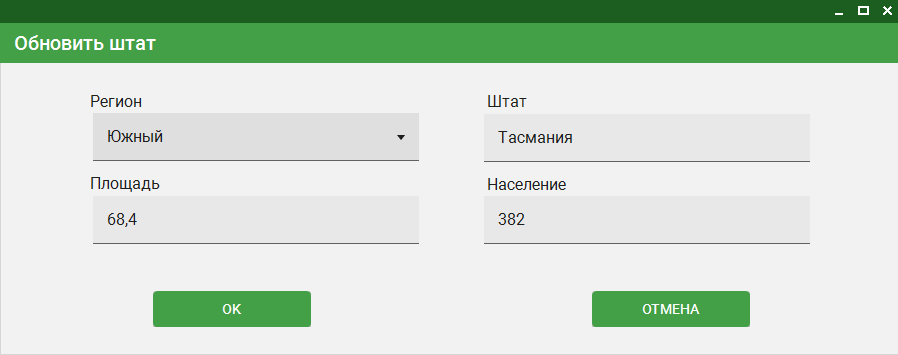


Рисунок 2.4 – Окно обновления штата

При нажатии на кнопку «Удалить» пользователю предоставиться выбор удалить или нет выбранную запись.

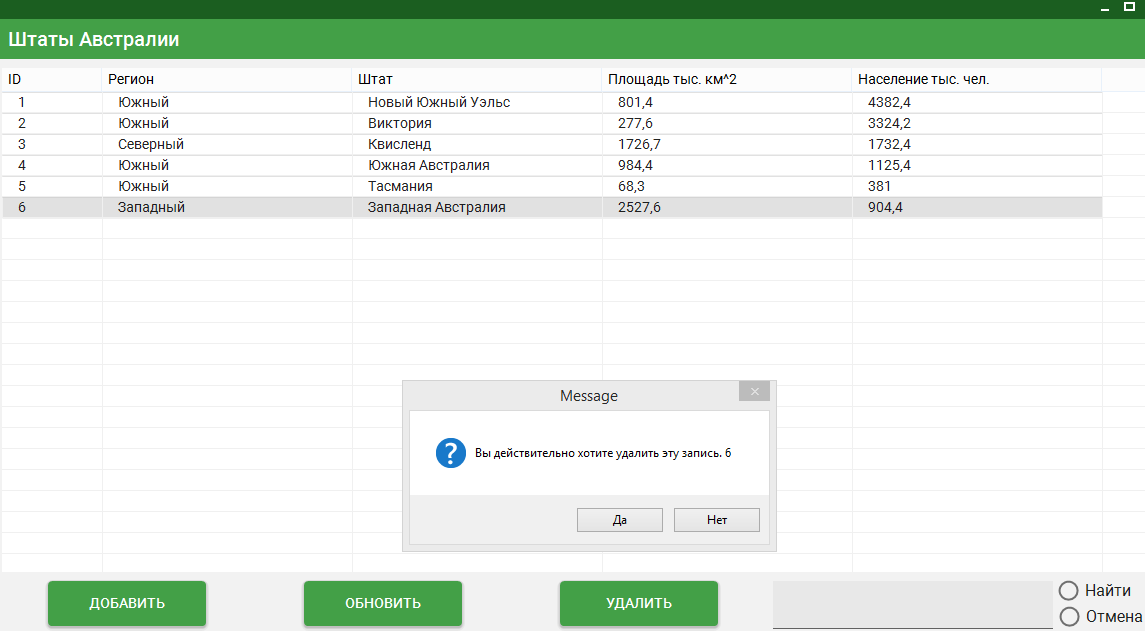


Рисунок 2.5 – Функционал удаления

При заполнении TextBox нажатии на кнопку «Найти» пользователь получит результат поиска, при нажатии кнопка «Отмена» результат обнулится.

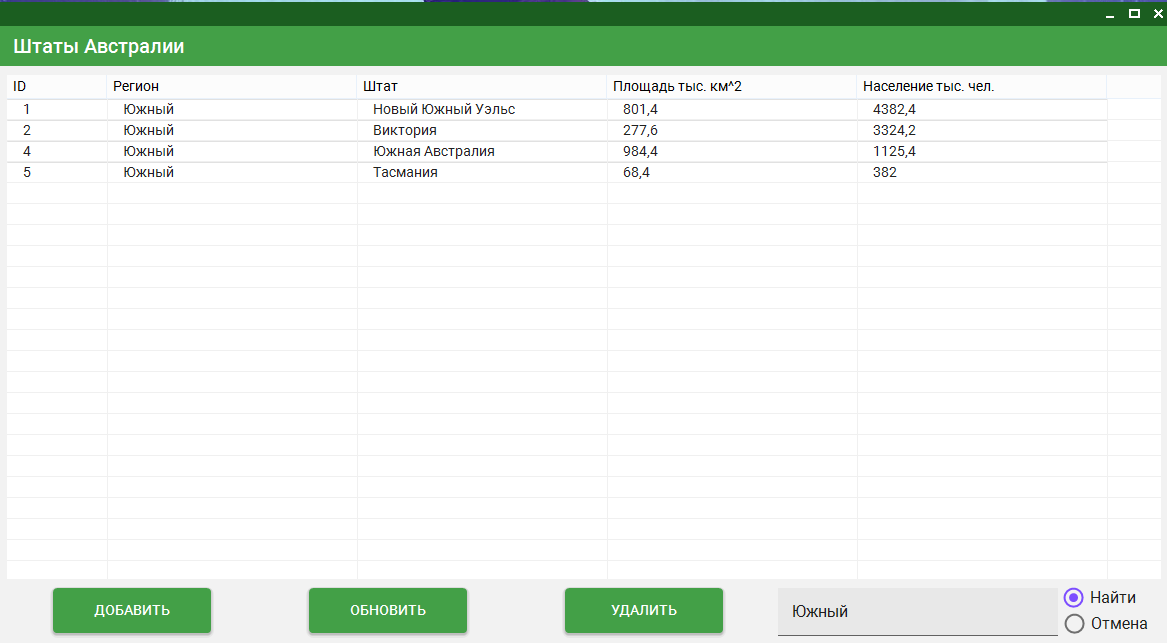
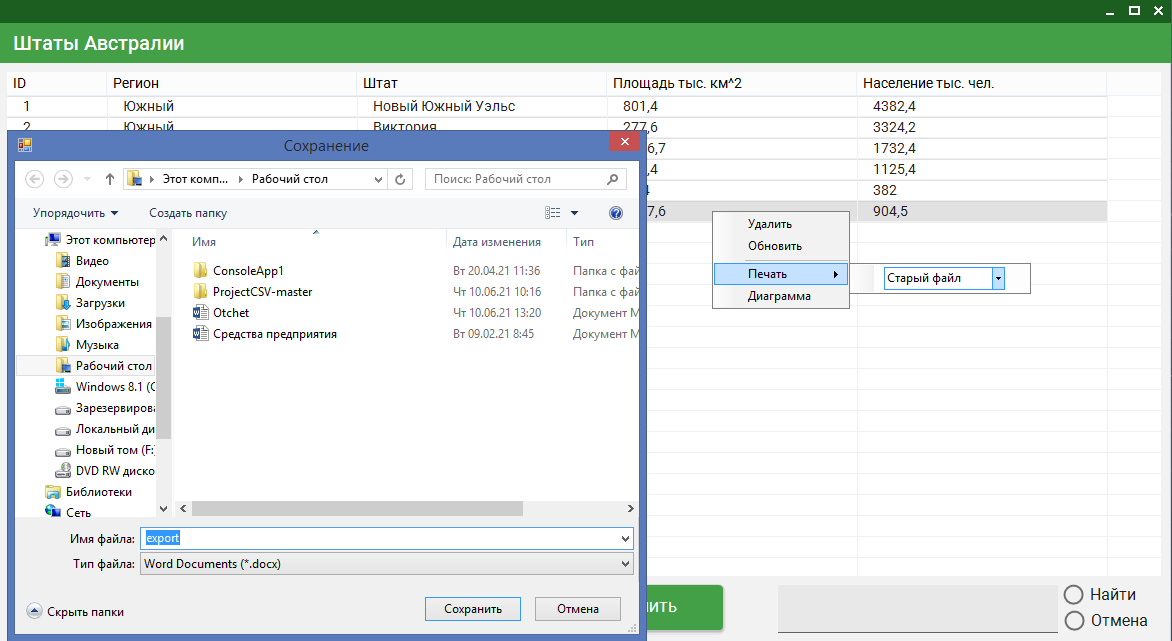


Рисунок 2.6 – Функционал поиска

При вызове ContexMenuStrip вы также может обновлять, удалять записи и печатать исходный и обновленный файл.

Рисунок 2.7 – Функционал печати



При нажатии на кнопу «Диаграмма» в ContextMenuStrip пользователь откроет форму с двумя диаграммами.

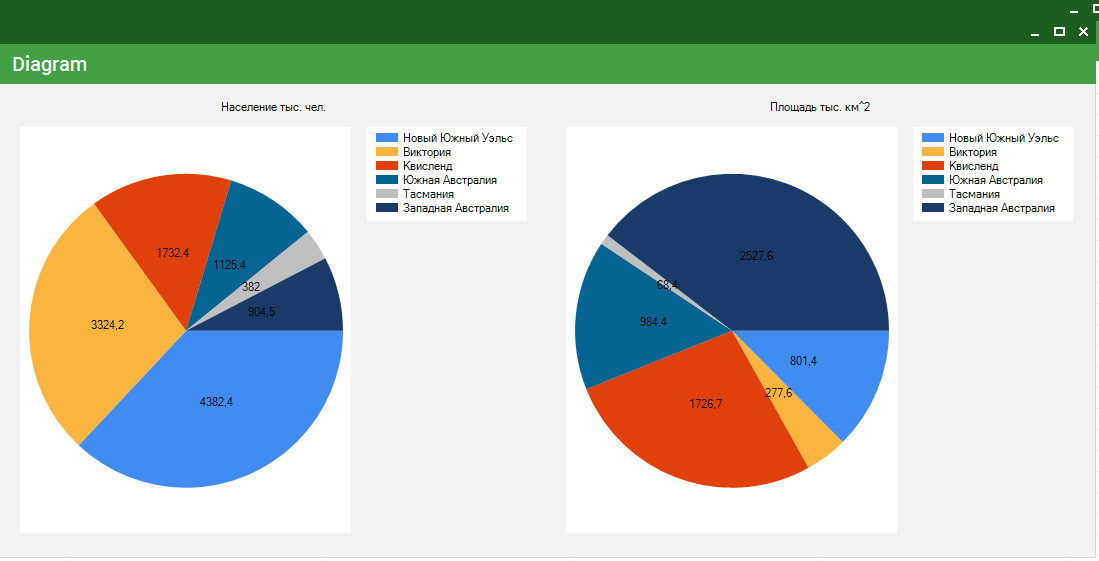


Рисунок 2.8 – Форма с диаграммами

# ЗАКЛЮЧЕНИЕ

В ходе практики была реализованная База Данных «Штат Австралии», ее кнопочная форма, а также объектно-ориентированное приложение для работы с csv файлом. Было выполнено 3 запроса, а также все требования к ООП, с использованием сред реализации MC Access и Visual Studio.

В ходе работы над заданием использовались знания, полученные по таким дисциплинам: «Базы данных», «Программирование на объектно-ориентированном языке C#»

# СПИСОК ЛИТЕРАТУРЫ

1. Гончаров, А. Ю. Access 2003: Самоучитель [Текст] / А. Ю. Гончаров. М: Кудиц-образ, 2004. 270 с.
2. Базы данных: учебник для высших учебных заведений / под ред. А.Д.Хомоненко. - СПб.: КОРОНА принт, 2002.
3. Кузин, А. В. Базы данных: учеб. пособие для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" [Текст] / А. В. Кузин, С. В. Левонисова. – М.: Академия, 2010. 315 с.
4. Виллариал Б. Программирование Access 2002 в примерах. Пер. с англ. – М.: Кудиц-образ, 2003. – 496с.
5. Корнеев В.В. и др. Базы данных. Интеллектуальная обработка информации // М.:Нолидж, 2000.- 352 с.
6. Карпова Т.С. Базы данных. Модели, разработка, реализация/СПб.: Питер, 2002. - 304 с.
7. Windows Forms Metanit // Сайт Metanit. URL: <https://metanit.com/sharp/windowsforms> (дата обращения 10.06.2021).
8. CsvHelper // Сайт Joshclose.github. URL: <https://joshclose.github.io/CsvHelper> (дата обращения 09.06.2021)