

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО «Тувинский государственный университет»
ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
КАФЕДРА ИНФОРМАТИКИ**

Выпускная квалификационная работа
(бакалаврская работа)

**СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ «ЛЕКСИКА
ПОХОРОННО-ПОМИНАЛЬНОГО ОБРЯДА В ТУВИНСКОМ ЯЗЫКЕ»**

Работа допущена к защите
И.о. зав. кафедрой информатики
Тарыма А.К., к.п.н., доцент
(фамилия, и.о., уч.степень, звание)

Студентки 4 курса 4 группы
направления подготовки 02.03.02
Фундаментальная информатика и
информационные технологии
Ламажык Сай-Суу Оюн-ооловны

Работа защищена «19»июня 2020 г.
С оценкой _____
Председатель ГЭК _____
(подпись)

(подпись)
« ____ » _____ 2020 г.

Сенатов В.И. д.ф-м.н., профессор
ведущий научный сотрудник
Института
вычислительного моделирования
СО РАН, Красноярск

Научный руководитель:

Члены комиссии _____

(подписи)

(подпись)
Арапчор Т.А., ст. преподаватель
кафедры информатики
(фамилия, и.о., должность, уч.степень, звание)

Кызыл – 2020

Содержание

<u>Введение</u>	3
<u>Глава I. Технология баз данных в среде программирования</u>	4
1.1. <u>Основные понятия баз данных</u>	4
1.2. <u>Язык программирования и базы данных</u>	9
1.3. <u>Принципы работы с базами данных в C#</u>	12
<u>Глава II. Электронный словарь «Лексика похоронно-поминального обряда в тувинском языке»</u>	15
2.1. <u>Структура базы данных</u>	15
2.2. <u>Структура программы</u>	16
2.3. <u>Организация поиска в базе данных</u>	23
<u>Заключение</u>	25
<u>Список источников и использованной литературы</u>	26
<u>Приложения</u>	28
<u>Приложение 1 «Заставка программы»</u>	28
<u>Приложение 2 «О программе»</u>	28
<u>Приложение 3 «Помощь»</u>	30
<u>Приложение 4 «Основная часть программы»</u>	31
<u>Приложение 5 «Добавление новой записи»</u>	39
<u>Приложение 6 «Ввод пароля»</u>	40

Введение

В наше время на помощь интересующимся филологией приходит много различных словарей. И все эти словари можно найти в интернете в электронном варианте. Все большее количество организаций предоставляют свои услуги в on-line среде, и базы данных не исключение.

Тем не менее, информацию о лексемах в тувинском языке в электронном варианте невозможно найти. Для решения данной задачи, необходимо приложение, которое бы предоставляло доступ к такому словарю.

Целью этой работы является разработка системы управления базами данных «Лексика похоронно-поминального обряда в тувинском языке», которая позволяет просматривать слова, редактировать и добавлять новые записи.

Предметом работы является разработка СУБД на языке программирования C# в среде программирования Microsoft Visual Studio 2010 «Лексика похоронно-поминального обряда в тувинском языке».

Объектом работы является процесс создания программного продукта.

Для реализации цели были поставлены следующие **задачи:**

- Изучить литературу по теории баз данных;
- Разработать структуру будущей базы данных;
- Создать программную оболочку в среде разработки «Microsoft Visual Studio 2010»;
- Создать административный модуль, который позволяет администратору и редактору доступ к базе данных и дает возможность добавлять, редактировать и удалять записи.

В электронном варианте нет такого словаря, что и является **актуальностью моей работы.**

Глава I. Технологии баз данных в среде программирования

1.1. Основные понятия баз данных

Одной из важнейших областей применения компьютеров является переработка и хранение больших объемов информации в различных сферах деятельности человека: в экономике, банковском деле, торговле, транспорте, медицине, науке и т. д. Для того чтобы понимать разработку программного продукта на основе базы данных, приведу понятие базы данных.

База данных (БД) – это совокупность структурированных данных, относящихся к определенной предметной области. Или база данных – это информационная модель, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств.

Базы данных состоят из таблиц, которые представляют широкий диапазон категорий данных. Таблицы имеют определенную структуру, и данные, хранящиеся в них, соответствуют этой структуре.

Система программ, позволяющая создавать БД, обновлять хранимую в ней информацию, обеспечивающая удобный доступ к ней с целью просмотра и поиска, называется **системой управления базами данных (СУБД)**.

СУБД делятся на однозвенные, двухзвенные и трехзвенные.

Классификация баз данных

1. По способу хранения данных делятся на централизованные и распределённые.

Централизованная БД - вся информация хранится на одном компьютере. Это может быть автономный ПК или сервер сети, к которому имеют доступ пользователи-клиенты. Архитектуры систем централизованных *БД* с сетевым доступом делятся на два типа: на *файл-сервер* и *клиент-сервер*.



Рис. 1.1. БД с сетевым доступом (файл-сервер)

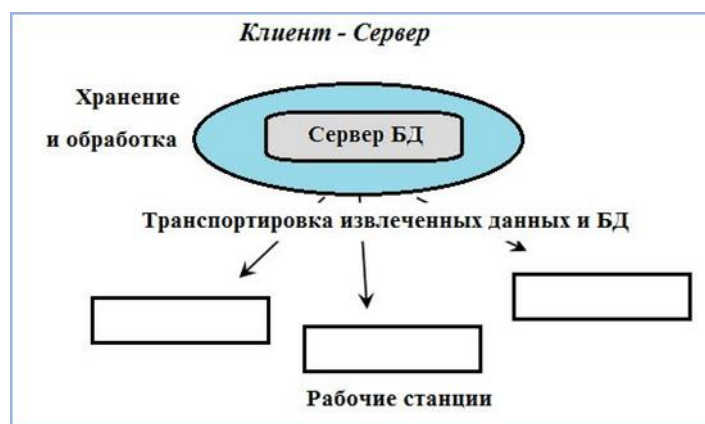


Рис. 1.2. БД с сетевым доступом клиент – сервер

Примером БД может быть расписание движения поездов. Здесь каждая строчка – запись отражает данные одного объекта. База включает поля: номер рейса, маршрута следования, время отправления и т.д.

Распределённая БД используются в локальных и глобальных компьютерных сетях. И еще разные части базы хранятся на разных компьютерах.

По способу доступа к данным БД подразделяются на два типа с локальным и с удаленным доступом.

Общая классификация баз данных представлена на рис. 1.3.

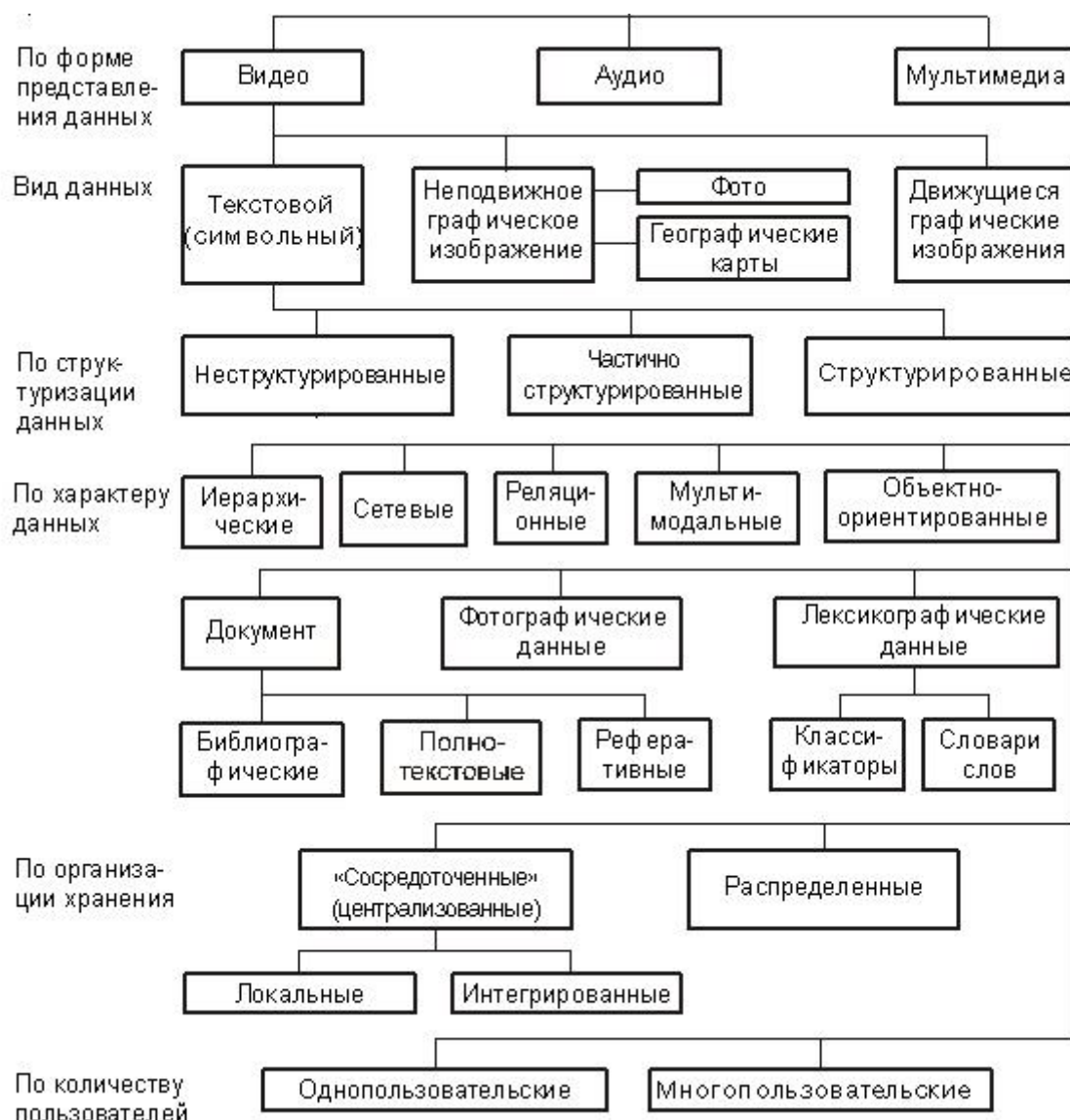


Рис. 1.3. Классификация баз данных.

Модели данных

Модель данных – это совокупность структур данных и операции их обработки. Модели используются для представления данных в информационных системах.

Различают три типа моделей данных: иерархическая, сетевая и реляционная.

Основное назначение моделей данных: возможность посмотреть в целом информационную картину без излишних деталей.

Иерархическая модель данных

Иерархическая модель данных строится по принципу иерархии типов объектов: один тип объекта является главным, а другие – подчиненными. К основным понятиям иерархической модели данных относятся: уровень, узел и связь. Узел – это совокупность атрибутов данных, описывающих некоторый объект.

Иерархическая модель позволяет быстро получать данные, но не отличается гибкостью.

Сетевая модель данных

Сетевая модель данных основана на тех же основных понятиях (уровень, узел, связь), что и иерархическая модель, но в сетевой модели каждый узел может быть связан с любым другим узлом.

Примером сетевой структуры может служить структура базы данных, содержащей сведения о студентах, занимающихся в спортивных секциях. Возможно участие одного студента в нескольких секциях, возможно также участие нескольких студентов в одной секции.

Сетевая модель данных, имеет более гибкую структуру, чем иерархическая, и поддерживает отношения «многие ко многим». Но быстро становится слишком сложной и неудобной для управления.

Реляционная модель данных

Реляционная модель данных использует организацию данных в виде двумерных таблиц. Каждая такая таблица, называемая реляционной таблицей или отношением, представляет собой двумерный массив и обладает следующими свойствами:

- все элементы в одном столбце имеют одинаковый тип и максимально допустимый размер;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;

- порядок следования строк и столбцов в таблице не имеет значения.

Основными структурными элементами реляционной таблицы являются поле и запись. Поле (столбец реляционной таблицы). Запись (строка реляционной таблицы).

Этапы проектирования баз данных:

1. Системный анализ предметной области
2. Инфологическое проектирование
3. Выбор СУБД
4. Дatalogическое проектирование
5. Физическое проектирование

I. Системный анализ предметной области

В первом этапе проектирования баз данных рассматриваются цели и задачи, с помощью которого они будут решаться.

II. Инфологическое проектирование

Во втором этапе выполняется моделирование данных. Моделирование данных – процесс создания логической структуры данных.

Существует два подхода к моделированию данных:

1. Модель «сущность-связь»
2. Семантическая объектная модель

Модель «Сущность-Связь»

Сущность – это объект, идентифицируемый пользователем в предметной области. Класс сущностей – это совокупность сущностей, и описывается определенной структурой или форматом сущностей, составляющих этот класс.

Экземпляр сущности – соответствует объектам реальной действительности. Это конкретные персоны студентов, преподавателей и т.д. То есть представляет собой конкретную сущность.

Атрибуты сущности – это свойства сущности.

Связи – это записи в таблице могут зависеть от одной или нескольких записей другой таблицы, то есть взаимоотношения сущностей.

Типы связей:

- 1). «Один к одному» (1:1) – каждая запись (сущность) родительской таблицы связана только с одной записью дочерней.
- 2). «Один ко многим» (1:M) – каждая запись родительской таблицы связана с одной или несколькими записями дочерней.
- 3). «Многие ко многим» (M:N) – несколько записей одной таблицы связаны с несколькими записями другой.

III. Выбор СУБД

При выборе СУБД руководствуются следующими соображениями:

- системное программное обеспечение, с которым будет работать проектируемая БД и соответствующие ей приложения;
- методология и подходы к программированию, реализованные в той или иной СУБД;
- модель данных, которая будет встроена в конкретную СУБД;

IV. Дatalogическое проектирование

При формировании дatalogической схемы, каждая из определённых в схеме сущностей отображается в таблицу, которая является одним отношением.

V. Физическое проектирование

В этом этапе необходимо на конкретной СУБД, реализовать базу данных по той информации, которую собрали, обработали и подготовили. Описываются модули, их назначение, а также структура модулей.

1.2. Язык программирования и базы данных

Чтобы более подробно рассмотреть данный раздел работы, я рассмотрю язык программирования C#. Но сначала нужно определить понятие универсального интерфейса для доступа к базам данных.

Протокол ODBC

ODBC (Open Database Connectivity – открытый доступ к базам данных) – это протокол, используемый для подключения базы данных Microsoft Access к внешнему источнику данных, например Microsoft SQL Server. Целью разработки протокола ODBC считается стандартизация механизмов взаимодействия с различными СУБД. Основная идея заключалась в том, что в разработке универсального интерфейса на уровне семейства операционных систем Windows, который мог бы быть поддержан в разных СУБД.

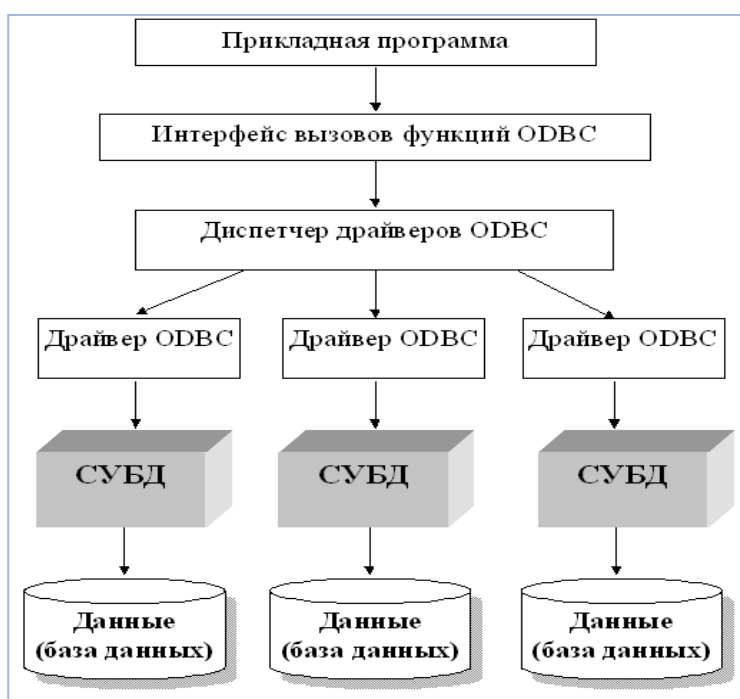


Рис. 1.4. Схема выполнения программы с использованием протокола ODBC для доступа к данным

Язык программирования C#

Для C# приложений есть несколько универсальных методов доступа, с помощью которых приложения могут быть связаны с базами данных. Все эти методы можно поделить на две группы, одна из них представляет прямой или косвенный вызов СУБД через программные интерфейсы, а вторая – использование объектных интерфейсов.

Вызов через программный интерфейс:

1. Использование программного интерфейса ODBC.
2. Прямой вызов программного интерфейса.

Объектные интерфейсы СУБД:

1. Интерфейс OLE DB
2. Интерфейс ADO (ActiveX Data Objects)

Интерфейс OLE DB представляет собой открытый стандарт, и предназначен для универсального доступа приложений к базам данных. В качестве данных могут быть *базы данных*, простые документы, таблицы *Excel* и любые другие источники данных. В отличие от доступа, предоставляемого посредством драйверов ODBC, *OLE DB* позволяет реализовывать *доступ* к источникам данных. И еще можно отметить три главных компонента OLE DB: потребители (**consumers**), провайдеры данных (**data providers**) и сервисные компоненты (**service components**).

Интерфейс ActiveX Data Objects (ADO) - представляет собой программный интерфейс для доступа к данным из приложений. С точки зрения программирования ADO и его расширения являются упрощенным высокоуровневым объектно-ориентированным интерфейсом к OLE DB;

Если пользователей БД много, то каждый из них будет во время своей работы держать как минимум одно соединение с сервером БД. Это отнимает большое количество ресурсов сервера, и потому было введено понятие многоуровневой системы.

Метод доступа ADO.NET, доступный приложениям на платформе Microsoft.NET, позволяет создавать разновидность многоуровневых систем - называемые *рассоединенные (disconnected)* системы. Эти системы позволяют получить данные локальным путем, извлеченные из базы данных, выполнить их обработку, а затем отправить обновленные данные обратно на сервер.

Данные, извлеченные из сервера СУБД методом ADO.NET, сохраняются в объекте класса DataSet (может хранить в себе одновременно несколько таблиц данных, в том числе связанных между собой (relatedtables)).

1.3. Принципы работы с базами данных в C#

Для решения моей задачи я использовала базу данных, созданную в Microsoft Access 2010 версии, и присоединила её как локальную БД для будущего проекта. Так как в моей работе использовалась технология ADO.NET, поэтому будем рассматривать только ее.

Метод доступа ADO.NET позволяет создавать рассоединенные системы.



Рис.1.6. Роль ADO.NET в обеспечении доступа к данным.

Проведу небольшой обзор объектов ADO.NET.

1. **Источник данных DataSet** – это основной объект ADO. Этот объект состоит из объектов типа DataTable (таблицы) и объектов DataRelation (каждый объект DataRelation выражает отношение между двумя таблицами.)
2. **Объект DataRow** представляется собой набор записей из таблицы DataTable.
3. **Объект DataAdapter** - осуществляет взаимодействие между DataSet и источником данных (в моем случае с БД в MS Access).
4. **Объекты DBConnection и DBCommand** - фундаментальные объекты, они позволяют редактирование БД в DataSet, не прибегая к самому источнику. Объект DBConnection реализует связь с источником данных. А объект DBCommand позволяет послать БД команду (команду SQL или хранимую процедуру).

Подключение БД MS Access (версии 2010 года) к приложению C# с помощью мастера среды программирования

Во вкладке «Источники данных» выбрать «Добавить новый источник данных» → «База данных» → «Набор данных» → «Создать подключение». В

качестве поставщика данных нужно выбрать *Provider=Microsoft.ACE.OLEDB.12.0*, а не *Provider=Microsoft.Jet.OLEDB.4.0*, потому что последний не поддерживает MS Access версии выше 2003 года.

При создании подключения БД автоматически копируется в папку, которая хранит исполняемый файл приложения. Команды удаления, добавления и обновления БД также будут прописаны автоматически мастером среды программирования.

Использование SQL-запросов в языке C# при редактировании БД

Команды изменения таблицы

Команда добавления полей в таблицу:

ALTER TABLE <имя таблицы> ADD <имя поля1> <тип данных>

[NULL | NOT NULL], <имя поля2> <тип данных> [NULL | NOT NULL]

и команда удаления полей таблицы:

ALTER TABLE <имя таблицы> DROP <имя поля1>, <имя поля2>

Команда удаления таблицы

DROPTABLE <имя таблицы> [RESTRICT | CASCADE]

Данная команда удалит таблицу и все входящие в неё записи. Если в операторе используется ключевое слово *RESTRICT*, то при наличии в БД хотя бы одного объекта, выполнение оператора *DROP TABLE* будет отменено. Если же указано ключевое слово *CASCADE*, автоматически удаляются также все объекты базы данных.

Команда добавления записей в таблицу

INSERT INTO имя таблицы VALUES (значение поля1,..., значение поляN)

Команда редактирования записи в таблице

UPDATE имя таблицы SET имя поля1 = значение1,..., имя поля N = значениеN

WHERE условие

где условие задает, какую запись редактировать.

Команда удаления записи в таблице

DELETE FROM имя таблицы WHERE условие

где *условие* задает, какую запись удалять.

Глава II. Электронный словарь «Лексика похоронно-поминального обряда в тувинском языке»

2.1. Структура базы данных

База данных состоит из четырех таблиц формата .acscdb. Первая таблица – основа программы, которая содержит словарный запас словаря, она состоит из семи полей.

Имя поля	Тип данных	Значение
L_S_G	Числовой	Значение лексемы
L_S_PG	Числовой	Значение лексемы
Лексема	Текстовый	Лексема на тувинском языке
Перевод	Текстовый	Лексема на русском языке
Буквальный перевод	Текстовый	Лексема на русском языке
Диалект	Текстовый	
Источник	Текстовый	

Вторая таблица содержит всего два поля, в котором содержит значения лексем типа «Лексико-семантическая группа».

Имя поля	Тип данных	Значение
ID	Числовой	Ключевое поле
L_S_G	Текстовый	Значение лексемы

Третья таблица содержит всего два поля, в котором содержит значения лексем типа «Лексико-семантическая подгруппа».

Имя поля	Тип данных	Значение
ID	Числовой	Ключевое поле
L_S_PG	Текстовый	Значение лексемы

Четвертая таблица содержит всего два поля и всего одну запись – пароль для входа в режим администрирования.

Имя поля	Тип данных	Значение
Код	Счетчик (длинное целое)	Ключевое поле
pass	Текстовый	Пароль

2.2. Структура программы

Программа рассчитана на студентов и преподавателей, преимущественно для филологического направления. Данная программа предназначена для изучения разбора слов тувинского языка.

Программа содержит 8 форм:

- Form1(Fon) – заставка программы;
- Form2 (Form_goal) – основная часть программы;
- Form3 (Form_new_rec)– форма добавления новой записи;
- Form4 (Form_pass) – форма ввода пароля для входа в режим администрирования;
- Form5 (aboutProgr) – форма, в которой выводится информация об авторе;
- Form6 (Form_z)– форма для просмотра таблицы «Лексико-семантическая группа (L_S_G)»;
- Form7 (Form6)– форма для просмотра таблицы «Лексико-семантическая подгруппа (L_S_PG)»;
- Form8 (Form7) – форма, в которой выводится информация о программе и об авторе.

Схематическая структура программы (рис.2.1):



Рис.2.1. Структура программы

После запуска программы, появляется заставка программы (рис.2.2), после которой и запускается сама программа (рис.2.3).

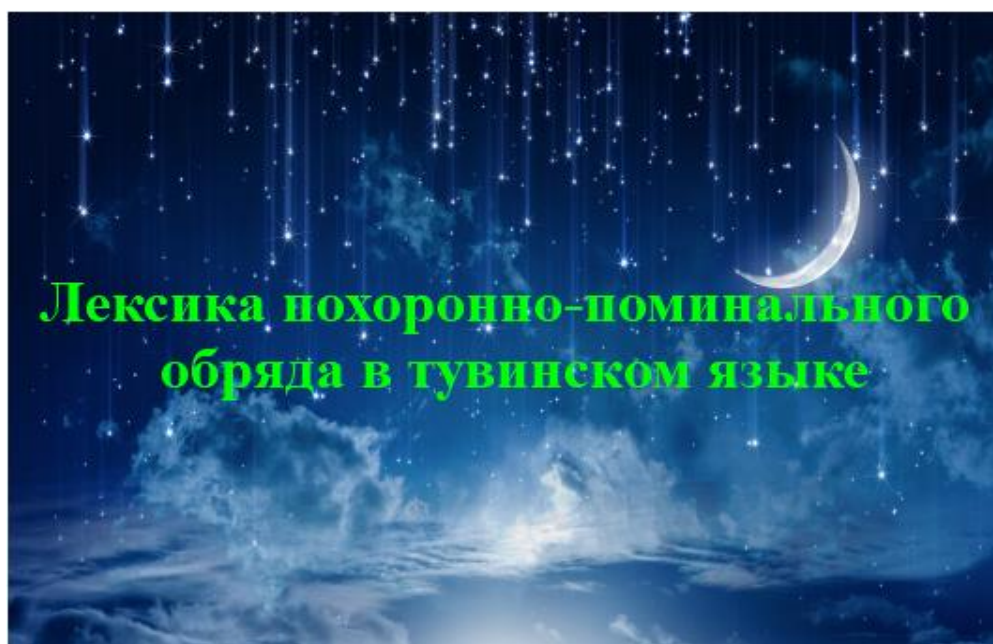


Рис.2.2. Заставка программы



Рис.2.3. Титульный лист программы

На рис.2.3. представлен титульный лист программы. На главном меню (компонент MenuStrip) имеются четыре пункта: «Словарь», «Типы групп», «Справка» и «Заккрыть». Пункт меню «Словарь» имеет подпункты «Просмотр», «Войти в режим администратора», «Редактирование», «Добавление новой записи». Пока пользователь не вошел в режим администратора возможность редактирования и добавления новых записей в словарь отключена.

Раздел просмотра словаря (рис.2.4) содержит область вывода всех слов (компонент DataGridView), входящих в словарь; область поиска (компонент TextBox, ComboBox).

Лексика похоронно-поминального обряда в тувинском языке

Словарь Типы групп Справка Закрыть

Просмотреть лексемы

Посмотреть весь список

Введите слово для поиска

для 159 Сортировка по возрастанию

Группа	Подгруппа	Лексема	Перевод	Буквальный перевод	Диалект	Источник
1	1	мөчүтүр	умереть	скончаться	лит., заруб.	ТРС 1968: 302; И-8; И-6
1	1	чарлып чоруткан	разлучившись ушел	разлучившись ушел	лит.	ТРС 1968: 519
1	2	чок алаар	умереть	не стать	лит., заруб.	И1, И3, И5
1	2	калыр	умереть	оставаться	лит., заруб.	ТРС 1968: 223; И-3
1	2	багай алаар	умереть	стать плохим	лит.	И4, И3
1	2	дүвүрээр	заканчивать жизнь	тревожиться	лит.	И3, И3
1	2	сылдызы өжер	погас звезда	утасла звезда	лит.	И7, И4
1	2	ужа бээр	упасть	умереть	лит.заруб.	Кенин-Топсан 2006: 54; И12
1	2	чой баар	ухлеть	уйти	цзнг., заруб.	Кенин-Топсан 2006: 54; Серен 2014: 35
1	2	донгаар	наклоняться	нагибаться	цзнг.	Монгуш 2004: 66; И13
1	2	чер ыгарар	кусать землю	кусать землю	цзнг.	Монгуш 2004: 66; И13

Рис.2.4. Просмотр лексем

Вход в режим администрирования

Для того чтобы редактировать данные словаря необходимо иметь доступ в режим администратора. Что войти в этот режим необходимо ввести пароль в отдельном окне (рис.2.5.), который выводится на экран после нажатия пункта меню «Вход в режим администратора». Пароль: **fiit9fmf** для входа в режим администратора. Пользователь вводит пароль в область ввода (компонент TextBox). Проверка правильности введенного пароля происходит по событию щелчка на кнопке (Button) или нажатию клавиши «ENTER».

Ввод пароля

☐ Показать пароль

Принять

Рис.2.5. Ввод пароля

Раздел редактирования данных (рис.2.6.) дает пользователю возможность редактировать данные сразу же на области вывода информации, что дает возможность быстрого редактирования. После

редактирования/удаления записей необходимо сохранить произведенные изменения (кнопка «Сохранить», компонент Button), в противном случае, данные не сохранятся. Пользователь имеет возможность поменять существующий пароль для доступа в режим администрирования. Для этого ему достаточно ввести новый пароль в области ввода (компонент TextBox) в правом верхнем углу и нажать на кнопку «Изменить» (компонент Button), тогда пароль будет изменен.

Раздел добавления новой записи (рис.2.7.) достаточно прост в работе. На новой форме пользователь добавляет данные (в компоненты TextBox) и нажимает на кнопку «Добавить» (компонент Button) после чего введенные данные сохраняются в базу. После добавления новой записи программа не выходит из режима добавления записи, позволяя тем самым пользователю выбирать добавить ещё запись или закрыть данный раздел.

Лексика похоронно-поминального обряда в тувинском языке

Словарь Типы групп Справка Закрыть

Редактирование данных

Изменить пароль администратора

Посмотреть весь список

Запись

Сохранить Удалить

Поиск

Введите слово для поиска

Группа	Подгруппа	Лексема	Перевод	Буквальный перевод	Диалект	Источник
1	1	мечүүр	умереть	скончаться	лит., заруб.	ТРС 1968: 302; И-8; И - 6
1	1	чарлыт чоруткан	разлучившись ушел	разлучившись ушел	лит.	ТРС 1968: 519
1	2	чок апаар	умереть	не стать	лит., заруб.	И1, И3, И5
1	2	кальр	умереть	оставаться	лит., заруб.	ТРС 1968: 223; И - 3
1	2	багай апаар	умереть	стать покойным	лит.	И4, И 3
1	2	дүжүрээр	заканчивать жизнь	тревожиться	лит.	И 5, И 3
1	2	сыцдылы өжөр	потас звезда	утасла звезда	лит.	И7, И 4
1	2	ужа бээр	упасть	умереть	лит.заруб.	Кенин-Топсан 2006: 54; И 12
1	2	чой баар	уехать	уйти	цэнг., заруб.	Кенин-Топсан 2006:54; Серен 2014:35
1	2	донгалар	наклоняться	нагибаться	цэнг.	Монгуш 2004: 66; И 13
1	2	чер ызырар	кусать землю	кусать землю	цэнг.	Монгуш 2004: 66; И 13

Рис.2.6.Раздел редактирования.

Добавить новую запись

Группа 0 Подгруппа 0

Лексема Перевод

Буквальный перевод Диалект

Источник

Добавить Отменить

Рис.2.7. Раздел добавления новой записи.

Раздел «Типы групп» имеет подпункты «Лексико-семантическая группа» и «Лексико-семантическая подгруппа», в котором описаны значение лексем (компонеты Panel, DataGridView, BindingNavigator, BindingSource, DataSet).

Лексико-семантическая группа (L_S_G)

0 для 0

№	Группа
1	общая лексика, обозначающая смерть человека
2	лексика, обозначающая смерть младенца
3	лексика, связанная со смертью молодого человека
4	лексика, обозначающая смерть пожилого человека
5	лексика, обозначающая причину смерти
6	Лексика обрядов и обычаев, связанная с моментом смерти и до погребения умершего
*	

Рис.2.8. Лексико-семантическая группа

Лексико-семантическая подгруппа (L_S_PG)

0 для 0

№	Подгруппа
1	нейтральные
2	эвфемизмы
3	основанные на физиологии и анатомии человека
4	на народной этимологии
5	кочевой жизни
6	на буддийских представлениях
7	убийство:
8	болезнь:
9	несчастные случаи:
10	самоубийство:
11	извещение о смерти:
12	выражение соболезнования:
13	действия и выражения по отношению умершему:
14	буддийские обряды:
15	названия действий:

Рис.2.9. Лексико-семантическая подгруппа

Раздел «Справка» имеет 2 подпункта «Помощь» и «О программе». «О программе» передает общую информацию об авторе (компоненты AboutBox1, RichTextBox1):

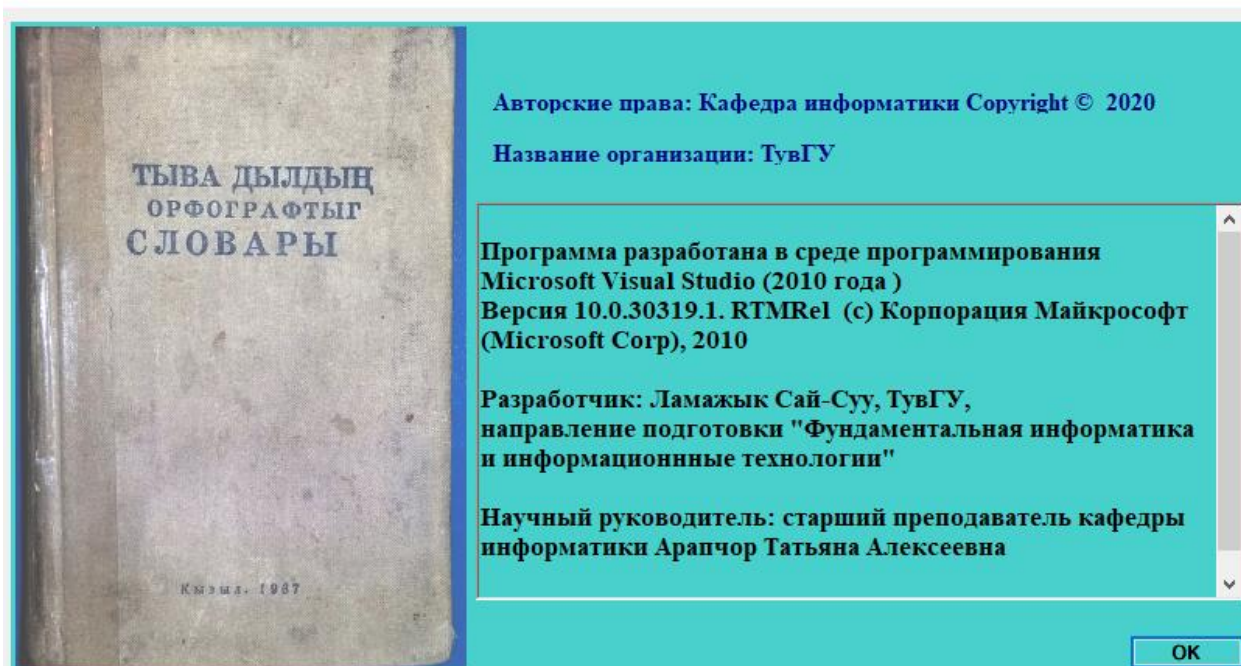


Рис.2.10. О программе

«Помощь» - передает информацию о словаре и об авторе (компонент RichTextBox1):

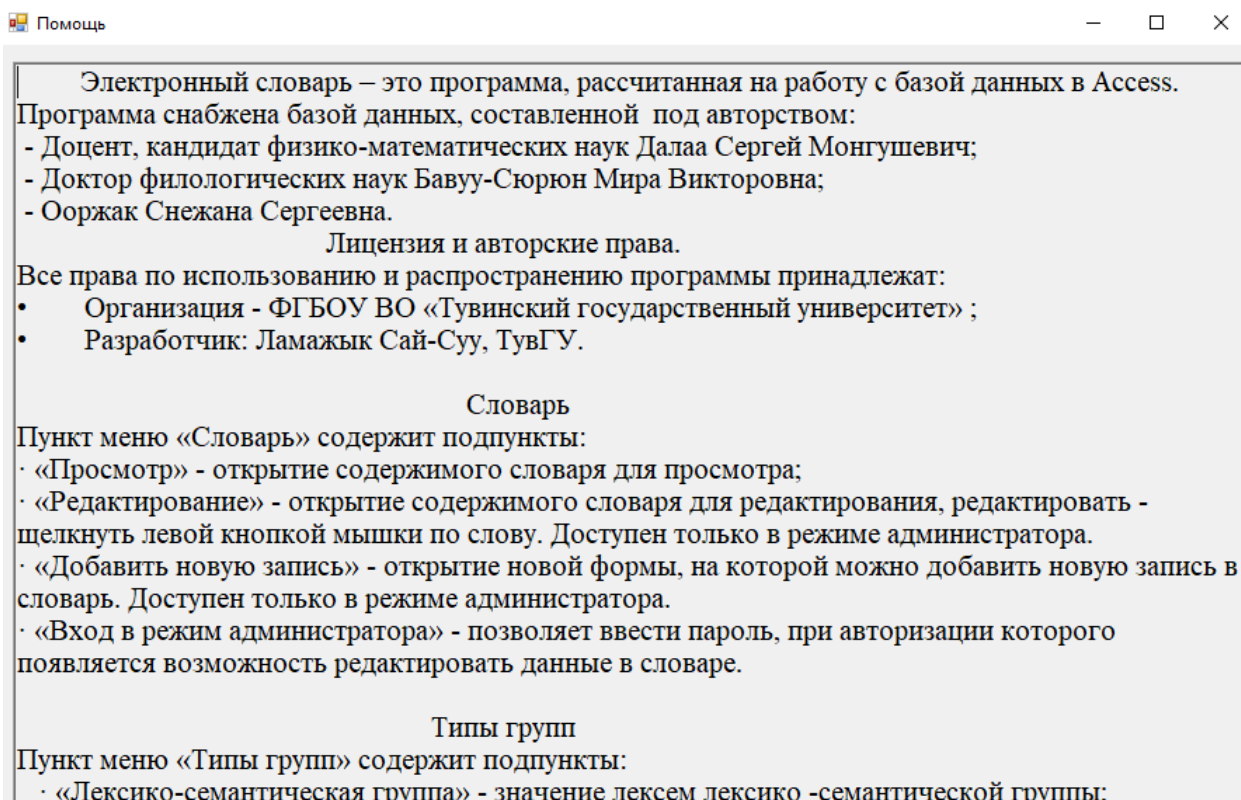


Рис.2.11. Помощь

2.3. Организация поиска в базе данных

В программе есть два режима поиска: непосредственный поиск (по мере поступления данных) и поиск, который работает при нажатии на кнопку «Поиск».

Принцип работы непосредственного поиска

Данный режим доступен в режиме просмотра словаря, его задача состоит в том, чтобы найти совпадение в любой части слова; начинает работу сразу после того, как пользователь ввел первую букву слова (отрезка слова).

Фрагмент кода, содержащий работу данного поиска:

```
private void tb_search_TextChanged(object sender, EventArgs e)
{
    bNavigator.Visible = false;

    if (red == false) // Данное условие проверяет, не открыт ли режим просмотра
    {
        dataGrid.Visible = true;
        string s2, s3, s4, s5, s6, s7, s8;
        string f;
```

//Для начала проверяется пуста ли область ввода

```
if (String.IsNullOrEmpty(tb_search.Text))  
f = "Leksema like """;  
else  
f = "Leksema like '%" + tb_search.Text + "%'"; //если не пуста выполняется запрос в
```

БД

```
dataGrid.DataSource = null; //Очистка области вывода БД для того, чтобы  
//вывести данные, удовлетворяющие условию поиска
```

```
DataRow[] R = ppoDataSet1.Main_PPO.Select(f); //Здесь временно будут храниться  
//найденные записи
```

```
dataGrid.Rows.Clear();
```

//Заполняем область вывода полученными данными

```
for (int i = 0; i < R.Length; i++)  
{  
    s2 = Convert.ToString(R[i]["L_S_G"]);  
    s3 = Convert.ToString(R[i]["L_S_PG"]);  
    s4 = Convert.ToString(R[i]["Leksema"]);  
    s5 = Convert.ToString(R[i]["Perevod"]);  
    s6 = Convert.ToString(R[i]["Bukvperevod"]);  
    s7 = Convert.ToString(R[i]["Dialekt"]);  
    s8 = Convert.ToString(R[i]["Istochnik"]);  
    dataGrid.Rows.Add(new String[] { s2, s3, s4, s5, s6, s7, s8 });  
} }
```

Принцип работы поиска, который работает при нажатии на кнопку

«Поиск»

Этот режим доступен в режиме редактирования словаря, его задача состоит в том, чтобы найти совпадение, начиная с начала слова; работает только по вызову пользователя. Единственное отличие в том, что поиск производится с начала слова, а в остальном принцип работы такой же, как и предыдущий.

Фрагмент кода, содержащий работу данного поиска:

```
private void l_search_Click(object sender, EventArgs e)  
{  
    listb = false;  
    bNavigator.Visible = false;  
    binSource.Filter = string.IsNullOrEmpty(tb_search.Text) ? null : "Leksema  
LIKE '" + tb_search.Text + "%'";  
}
```

Заключение

В результате выполнения работы была спроектирована и разработана, а также заполнена база данных приложения «Лексика похоронно-поминального обряда в тувинском языке».

В ходе выполнения работы мною была достигнута *цель*: создания системы управления базами данных словаря с помощью языка программирования C# в среде разработки Microsoft Visual Studio 2010 с использованием базы данных Microsoft Access.

Для этого выполнены следующие задачи:

- Изучить литературу по теории баз данных;
- Разработать структуру будущей базы данных;
- Создать программную оболочку в среде разработки «Microsoft Visual Studio 2010»;
- Создать административный модуль, который позволяет администратору и редактору доступ к базе данных и дает возможность добавлять, редактировать и удалять записи.

Мой выбор среды программирования Microsoft Visual Studio 2010 и языка программирования C# обусловлен тем, что данный язык имеет гибкую функцию форматирования текста, его перехода из одной кодировки в другую, что особенно важно для тувинского языка, который имеет кодировку Unicode.

Список источников и использованной литературы

1. Далаа С.М. Разработка баз данных в среде Visual Studio с помощью языка С#: [электронное учебное пособие]. /Далаа С.М. - Кызыл : Тувинский государственный университет (ТувГУ), 2016. – URL: <http://cdo.tuvsu.ru:8080/kafedry/informatika/oopr/subd/book.html> (дата обращения: 15.12.2019). – Текст : электронный.
2. Дейт К.Дж. Введение в системы баз данных. 8-ое издание.: Пер.с англ. - М.: Вильямс, 2005. - 1328 с. – Текст : непосредственный.
3. Информатика: Учеб.пособие для студ. пед. вузов / А.В. Могилев, Н.И. Пак, Е.К.Хеннер; Под ред. Е.К.Хеннера. -М.: Изд.центр "Академия",2000, -816 с. – Текст : непосредственный.
4. Крёнке. Д. Теория и практика построения баз данных. 8-ое изд./Д. Крёнке. - СПб.:Питер, 2003. - 800с.:ил. - (Серия "Классика computer science"). – Текст : непосредственный.
5. Конноли, Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика / Т. Конноли, К. Бегг, А. Страчан. , 2-е изд.; пер. с англ. - М.: Издат. дом «Вильямс», 2000. - 1120 с.: ил. – Текст : непосредственный.
6. Кузнецов С.Д. Основы баз данных.: учебное пособие /С.Д. Кузнецов -2-ое изд. испр. – Москва: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний,2007.-448с.:ил. — Текст : непосредственный.
7. Малыхина, М.П. Базы данных: основы, проектирование, использование / М.П. Малыхина. - Спб.: БХВ-Петербург, 2004. - 512 с.: ил. – Текст : непосредственный.
8. Мак-Манус, Джеффри, П., Голдштейн, Джеки, Прайс, Кевин, Т.Обработка баз данных на Visual Basic .NET, 3-е издание.: Пер. с англ. - М.:Изд.дом Вильямс, 2003. - 416 с.: ил. — Текст : непосредственный.

9. Национальный Открытый Университет "ИНТУИТ": сайт. – URL: <http://www.intuit.ru/> (дата обращения 11.02.2020). – Текст: электронный.
10. Работа с базами данных на языке C#: Технология ADO.NET: Учеб. пособие / сост. О. Н. Евсеева, А. Б. Шамшев. - Ульяновск: УлГТУ, 2009. - 170 с. – Текст непосредственный.
11. Фролов А.В., Фролов Г.В. - Визуальное проектирование приложений C#.- М.:КУДИЦ-ОБРАЗ, 2003 .- 512с. – Текст непосредственный.
12. Шилдт, Герберт , C# 4.0. полное руководство. Пер. с англ. - М.: ООО "И.Д. Вильямс", 2011. -1056с. – Текст непосредственный.

Приложение

Приложение 1 «Заставка программы»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace Slovar
{
    public partial class Fon : Form
    {
        Timer tmr = new Timer();
    public Fon()
    {
        InitializeComponent();
    }
    void tmr_Tick(object sender, EventArgs e)
    {
        Form_goal f2 = new Form_goal();
        tmr.Stop();
        this.Hide();
        f2.Show();
    }
    private void Fon_Load(object sender, EventArgs e)
    {
        tmr.Interval = 2150;
        tmr.Start();
        tmr.Tick += new EventHandler(tmr_Tick);
    }
}
}
```

Приложение 2 «О программе»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Linq;
using System.Reflection;
using System.Windows.Forms;
namespace Slovar
{
    partial class aboutProgr : Form
    {
        public aboutProgr()
        {
            InitializeComponent();
            this.Text = String.Format("О {0}", AssemblyTitle);
        }
    }
}
```

```

        this.labelProductName.Text = "Название продукта: Лексика ППО" + AssemblyProduct;
        this.labelCopyright.Text = " Авторские права: Кафедра информатики " +
AssemblyCopyright;
        this.labelCompanyName.Text = " Название организации: ТувГУ " +
AssemblyCompany;
        this.richTextBox1.Text = richTextBox1.Text + AssemblyDescription;
    }

    #region Методы доступа к атрибутам сборки

    public string AssemblyTitle
    {
        get
        {
            object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyTitleAttribute), false);
            if (attributes.Length > 0)
            {
                AssemblyTitleAttribute titleAttribute = (AssemblyTitleAttribute)attributes[0];
                if (titleAttribute.Title != "")
                {
                    return titleAttribute.Title;
                }
            }
            return
System.IO.Path.GetFileNameWithoutExtension(Assembly.GetExecutingAssembly().CodeBase);
        }
    }

    public string AssemblyVersion
    {
        get
        {
            return Assembly.GetExecutingAssembly().GetName().Version.ToString();
        }
    }

    public string AssemblyDescription
    {
        get
        {
            object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyDescriptionAttribute),
false);
            if (attributes.Length == 0)
            {
                return "";
            }
            return ((AssemblyDescriptionAttribute)attributes[0]).Description;
        }
    }

```

```

    public string AssemblyProduct
    {
        get
        {
            object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyProductAttribute),
false);
            if (attributes.Length == 0)
            {
                return "";
            }
            return ((AssemblyProductAttribute)attributes[0]).Product;
        }
    }
    public string AssemblyCopyright
    {
        get
        {
            object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyCopyrightAttribute),
false);
            if (attributes.Length == 0)
            {
                return "";
            }
            return ((AssemblyCopyrightAttribute)attributes[0]).Copyright;
        }
    }

    public string AssemblyCompany
    {
        get
        {
            object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyCompanyAttribute),
false);
            if (attributes.Length == 0)
            {
                return "";
            }
            return ((AssemblyCompanyAttribute)attributes[0]).Company;
        }
    }
    #endregion
}

```

Приложение 3 «Помощь»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;

```

```

using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Slovar
{
    public partial class Form7 : Form
    {
        public Form7()
        {
            InitializeComponent();
            this.richTextBox1.Text = richTextBox1.Text;
        }
    }
}

```

Приложение 4 «Основная часть программы»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Windows.Forms;
using System.Data.OleDb;
using System.IO;

namespace Slovar
{
    public partial class Form_goal : Form
    {
        bool red = false; //для инициализации редактировать\просмотр
        bool listb = false;
        string tb_s;
        OleDbDataAdapter adapter;
        OleDbCommandBuilder builder;
        OleDbCommand Tyva = new OleDbCommand();
        OleDbConnection conn = new
        OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=PPO.accdb");
        Encoding cp1251 = Encoding.Unicode;

        public Form_goal()
        {
            InitializeComponent();
            this.KeyPreview = true;
            this.MinimumSize = new Size(960, 583);
        }

        private void red_t()
        {

```

```

if (red == true)
{
    l_prim.Text = "Редактирование данных";
    l_adm.Visible = false;
    tb_search.Width = 227;
    tb_search.Location = new Point(93, 21);
    gb_red.Visible = true;
    l_search.Visible = true;
    bNavigator.Visible = true;
    dataGrid.ReadOnly = false;

}
else
{
    l_prim.Text = "Просмотреть лексемы";
    tb_search.Width = 230 + l_search.Width;
    gb_changepas.Visible = false;
    tb_search.Location = new Point(10, 21);
    gb_red.Visible = false;
    l_search.Visible = false;
    bNavigator.Visible = true;
    bNavigator.BindingSource = binSource;
    dataGrid.ReadOnly = true;
}
}

public void select_lb(int l, TextBox t, ComboBox cBo)
{
    tb_s = null;
    tb_s = t.Text;
    switch (l)
    {
        case 0:
        {
            tb_s = tb_s + cBo.SelectedItem.ToString();
            break;
        }
        case 1:
        {
            tb_s = tb_s + cBo.SelectedItem.ToString();
            break;
        }
        case 2:
        {
            tb_s = tb_s + cBo.SelectedItem.ToString();
            break;
        }
    }
}
}

```

```

private void Form_goal_Load(object sender, EventArgs e)
{
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "ppoDataSet1.PPO_pas". При необходимости она может быть перемещена или удалена.
    this.pPO_pasTableAdapter.Fill(this.ppoDataSet1.PPO_pas);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
    "ppoDataSet1.Main_PPO". При необходимости она может быть перемещена или удалена.
    this.main_PPOTableAdapter.Fill(this.ppoDataSet1.Main_PPO);

    this.p_gl.Dock = DockStyle.Fill;
    StreamReader sr = new StreamReader(new MemoryStream(cp1251.GetBytes("ё")),
cp1251);
    char[] chars = new char[1];
    sr.Read(chars, 0, 1);
    cBox.Items.Add(new String(chars));
    sr = null;
    sr = new StreamReader(new MemoryStream(cp1251.GetBytes("н")), cp1251);
    char[] chars1 = new char[1];
    sr.Read(chars1, 0, 1);
    cBox.Items.Add(new String(chars1));
    sr = null;
    sr = new StreamReader(new MemoryStream(cp1251.GetBytes("ҫ")), cp1251);
    char[] chars2 = new char[1];
    sr.Read(chars2, 0, 1);
    cBox.Items.Add(new String(chars2));
    listb = false;
    bNavigator.BindingSource = binSource;
    main_PPOTableAdapter.Fill(this.ppoDataSet1.Main_PPO);
}

private void Form_goal_FormClosing(object sender, FormClosingEventArgs e)
{
    Application.Exit();
}

private void nпосмомpToolStripMenuItem_Click(object sender, EventArgs e)
{
    red = false;
    red_t();
    this.p_gl.Visible = false;
    this.p_prosm.Visible = true;
    this.p_prosm.Dock = DockStyle.Fill;
    this.tb_search.Focus();
}

private void tb_search_TextChanged(object sender, EventArgs e)
{
    bNavigator.Visible = false;

    if (red == false)

```

```

{
    dataGrid.Visible = true;
    string s2, s3, s4, s5, s6, s7, s8;
    string f;
    if (String.IsNullOrEmpty(tb_search.Text))
        f = "Leksema like ";
    else
        f = "Leksema like '%" + tb_search.Text + "%'";
    dataGrid.DataSource = null;
    DataRow[] R = ppoDataSet1.Main_PPO.Select(f);
    dataGrid.Rows.Clear();
    for (int i = 0; i < R.Length; i++)
    {
        s2 = Convert.ToString(R[i]["L_S_G"]);
        s3 = Convert.ToString(R[i]["L_S_PG"]);
        s4 = Convert.ToString(R[i]["Leksema"]);
        s5 = Convert.ToString(R[i]["Perevod"]);
        s6 = Convert.ToString(R[i]["Bukvperevod"]);
        s7 = Convert.ToString(R[i]["Dialekt"]);
        s8 = Convert.ToString(R[i]["Istochnik"]);

        dataGrid.Rows.Add(new String[] { s2, s3, s4, s5, s6, s7, s8 });
    }
}

private void tb_search_Click(object sender, EventArgs e)
{
    cBox.Visible = true;
    if (listb != true) { tb_search.Text = ""; }
}

private void m_red_Click(object sender, EventArgs e)
{
    red = true;
    red_t();
    gb_changepas.Visible = true;
    l_adm.Visible = false;
    this.p_gl.Visible = false;
    this.p_prosm.Visible = true;
    this.p_prosm.Dock = DockStyle.Fill;
}

private void l_spis_Click(object sender, EventArgs e)
{
    bNavigator.Visible = true;
    dataGrid.DataSource = binSource;
    listb = false;
}

private void l_search_Click(object sender, EventArgs e)

```

```

{
    listb = false;
    bNavigator.Visible = false;
    dataGrid.DataSource = null;
    string s2, s3, s4, s5, s6, s7, s8;
    string f;
    f = "Leksema like %" + tb_search.Text + "%";
    DataRow[] R = ppoDataSet1.Main_PPO.Select(f);
    dataGrid.Rows.Clear();
    for (int i = 0; i < R.Length; i++)
    {
        s2 = Convert.ToString(R[i]["L_S_G"]);
        s3 = Convert.ToString(R[i]["L_S_PG"]);
        s4 = Convert.ToString(R[i]["Leksema"]);
        s5 = Convert.ToString(R[i]["Perevod"]);
        s6 = Convert.ToString(R[i]["Bukvperevod"]);
        s7 = Convert.ToString(R[i]["Dialekt"]);
        s8 = Convert.ToString(R[i]["Istochnik"]);

        dataGrid.Rows.Add(new String[] { s2, s3, s4, s5, s6, s7, s8 });
    }
}

public void comm()
{
    main_PPOTableAdapter.Connection.Close();
    adapter = new OleDbDataAdapter("SELECT * FROM Main_PPO", conn);
    builder = new OleDbCommandBuilder(adapter);
    try
    {
        dataGrid.EndEdit();
        conn.Open();
        adapter.Fill(ppoDataSet1);
        dataGrid.DataSource = ppoDataSet1.Tables[0];
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
    finally
    {
        {
        }
    }
}

private void l_save_Click(object sender, EventArgs e)
{
    listb = false;
    comm();
    adapter.Update(ppoDataSet1.Tables[0]);
    conn.Close();
    main_PPOTableAdapter.Connection.Open();
}

```

```

private void l_delete_Click(object sender, EventArgs e)
{
    listb = false;
    int rw;
    rw = dataGrid.CurrentRow.RowIndex;
    MessageBox.Show("Вы удаляете строку № " + rw);
    ppoDataSet1.Main_PPO.Rows[rw].Delete();
}

private void l_update_Click(object sender, EventArgs e)
{
    dataGrid.DataSource = binSource;
    main_PPOTableAdapter.Fill(ppoDataSet1.Main_PPO);
}

private void l_spis_MouseUp(object sender, MouseEventArgs e)
{
    l_spis.BorderStyle = BorderStyle.FixedSingle;
}

private void l_spis_MouseDown(object sender, MouseEventArgs e)
{
    l_spis.BorderStyle = BorderStyle.Fixed3D;
}

private void l_save_MouseUp(object sender, MouseEventArgs e)
{
    l_save.BorderStyle = BorderStyle.FixedSingle;
}

private void l_save_MouseDown(object sender, MouseEventArgs e)
{
    l_save.BorderStyle = BorderStyle.Fixed3D;
}

private void l_delete_MouseUp(object sender, MouseEventArgs e)
{
    l_delete.BorderStyle = BorderStyle.FixedSingle;
}

private void l_delete_MouseDown(object sender, MouseEventArgs e)
{
    l_delete.BorderStyle = BorderStyle.Fixed3D;
}

private void l_search_MouseUp(object sender, MouseEventArgs e)
{
    l_search.BorderStyle = BorderStyle.FixedSingle;
}

```

```

private void l_search_MouseDown(object sender, MouseEventArgs e)
{
    l_search.BorderStyle = BorderStyle.Fixed3D;
}

private void m_new_Click(object sender, EventArgs e)
{
    Form_new_rec f3 = new Form_new_rec();
    f3.ShowDialog();
}

private void m_p_Click(object sender, EventArgs e)
{
    Form_pass f4 = new Form_pass();
    string f = "Kod=1";
    DataRow[] fselr = ppoDataSet1.PPO_pas.Select(f);
    l_password.Text = Convert.ToString(fselr[0]["pass"]);
    l_adm.Text = "";
    f4.Owner = this;
    f4.ShowDialog();
}

private void таблицаTunaLSGToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form_z f5 = new Form_z();
    f5.ShowDialog();
}

private void таблицаTunaLSPGToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form6 f6 = new Form6();
    f6.ShowDialog();
}

private void помощьToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form7 f7 = new Form7();
    f7.ShowDialog();
}

private void оПрограммеToolStripMenuItem_Click(object sender, EventArgs e)
{
    aboutProgr w = new aboutProgr();
    w.ShowDialog();
}

private void cBox_SelectedIndexChanged(object sender, EventArgs e)
{
    if (String.IsNullOrEmpty(tb_search.Text)) tb_search.Clear();
    select_lb(cBox.SelectedIndex, tb_search, cBox);
    listb = true;
    tb_search.Text = tb_s;
}

```

```

        tb_search.SelectionStart = tb_search.Text.Length;
        tb_search.Focus();
    }

    public string[] finding(string kk)
    {
        string s2;
        string f;
        f = "Leksema like '" + kk.ToString() + "%'";
        DataRow[] R = ppoDataSet1.Main_PPO.Select(f);
        string[] t = new string[R.Length];
        for (int i = 0; i < R.Length; i++)
        {
            s2 = Convert.ToString(R[i]["Leksema"]);
            t[i] = s2.ToString();
        }
        return t;
    }

    static string delStr(string s, char t)
    {
        string s1 = t.ToString();
        int l = s.IndexOf(t);
        if (l > -1)
        {
            s = s.Remove(l, s1.Length);
            return delStr(s, t);
        }
        return s;
    }

    private void b_change_Click(object sender, EventArgs e)
    {
        try
        {
            Tyva.Connection = conn;
            conn.Open();
            string s = "UPDATE PPO_pas SET [pass]=''" + textBox1.Text + "' WHERE Kod=1";
            Tyva.CommandText = s;
            Tyva.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
        finally
        {
            conn.Close();
            textBox1.Clear();
            MessageBox.Show("Пароль успешно изменен");
        }
    }

```

```

    }
}

private void toolStripButton1_Click(object sender, EventArgs e)
{
    string tooltext = toolStripButton1.Text;
    if (tooltext == "Сортировать по возрастанию")
    {
        dataGrid.Sort(dataGrid.Columns[0], ListSortDirection.Ascending);
        tooltext = "Отменить сортировку";
    }
    else
    {
        if (tooltext == "Отменить сортировку")
        {
            dataGrid.DataSource = null;
            binSource.DataMember = null;
            binSource.DataMember = "Main_PPO";
            dataGrid.DataSource = binSource;
            tooltext = "Сортировать по возрастанию";
        }
        toolStripButton1.Text = tooltext;
    }
}

private void выходToolStripMenuItem_Click(object sender, EventArgs e)
{
    string caption = "Выход из программы";
    string message = "Вы уверены, что хотите выйти из программы?";
    DialogResult result = MessageBox.Show(message, caption, MessageBoxButtons.YesNo);
    if (result == DialogResult.No)
    {
        return;
    }
    else
    {
        {
            this.Close();
        }
    }
}
}
}

```

Приложение 5 «Добавление новой записи»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

```

```

namespace Slovar
{
    public partial class Form_new_rec : Form
    {
        public Form_new_rec()
        {
            InitializeComponent();

            private void button1_Click(object sender, EventArgs e)
            {
                OleDbCommand Tyva = new OleDbCommand();
                OleDbConnection conn = new
                OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=PPO.accdb"); ;
                Tyva.Connection = conn;
                try
                {
                    conn.Open();
                    string s = "INSERT INTO Main_PPO
(L_S_G,L_S_PG,Leksema,Perevod,Bukvperevod,Dialekt,Istochnik) VALUES ("";
                    s = s + (num1.Text.ToString()) + ", ";
                    s = s + (num2.Text.ToString()) + ", ";
                    s = s + (textBox1.Text.ToString()) + ", ";
                    s = s + (textBox2.Text.ToString()) + ", ";
                    s = s + (textBox3.Text.ToString()) + ", ";
                    s = s + (textBox4.Text.ToString()) + ", ";
                    s = s + (textBox5.Text.ToString()) + ")";
                    Tyva.CommandText = s;
                    Tyva.ExecuteNonQuery();
                }

                catch (Exception ex)
                {
                    MessageBox.Show(ex.ToString());
                }
                finally
                {
                    conn.Close();
                    MessageBox.Show("Запись добавлена, нажмите просмотреть весь список,
чтобы обновить данные");
                }
            }

            private void button2_Click(object sender, EventArgs e)
            {
                this.Close();
            }
        }
    }
}

```

Приложение 6 «Ввод пароля»

```

using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Slovar
{
    public partial class Form_pass : Form
    {
        public Form_pass()
        {
            InitializeComponent();

            private void enter_cl()
            {
                Form_goal main = this.Owner as Form_goal;
                try
                {
                    string a;
                    a = textBox1.Text.ToString();
                    if (a == main.l_password.Text)
                    {
                        main.l_adm.Text = "1";
                    }
                    else
                    {
                        MessageBox.Show("Вы ввели неправильный пароль", "Ошибка");
                        main.l_adm.Text = "0";
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
            finally
            {
                this.Close();
            }
        }

        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (e.KeyChar == (char)Keys.Enter)
            {
                enter_cl();
            }
        }
    }
}

```

```

    }

    private void checkBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBox1.Checked == true)
            textBox1.UseSystemPasswordChar = false;
        else
            textBox1.UseSystemPasswordChar = true;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        enter_cl();
    }

    private void Form_pass_FormClosed(object sender, FormClosedEventArgs e)
    {
        Form_goal main = this.Owner as Form_goal;

        if (main.l_adm.Text == "1")
        {
            main.m_red.Visible = true;
            main.m_new.Visible = true;
            main.m_p.Visible = false;
            main.l_adm.Text = "Вы вошли в режим администратора";
            main.l_adm.Visible = false;
        }
    }
}
}
}

```