

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
Пермский национальный исследовательский политехнический университет
Лысьвенский филиал

Факультет профессионального образования
Направление 09.03.01 Информатика и вычислительная техника
Кафедра Естественнонаучных дисциплин

Зав. кафедрой ЕН
канд. физ.-мат. наук
_____/И.Т. Мухаметьянов
« ____ » _____ 2017 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
на соискание академической степени бакалавра

На тему «Разработка информационной системы “Музей”»

Студент: _____ / Рябов А.А.
(подпись) (Фамилия И.О.)

Состав ВКР:

1. Пояснительная записка на ___ стр.
2. Электронный носитель с материалами ВКР.

Руководитель ВКР:

(подпись, дата) Канд. техн. наук,
доцент кафедры ЕН ЛФ ПНИПУ
Конюхов М.И.

Лысьва, 2017 г.

Содержание

| | |
|--|----|
| ВВЕДЕНИЕ | 8 |
| 1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ..... | 9 |
| 1.1 Информационные системы..... | 9 |
| 1.2 Этапы разработки программного обеспечения ИС..... | 9 |
| 1.2.1 Постановка задачи..... | 9 |
| 1.2.2 Разработка пользовательского интерфейса..... | 10 |
| 1.2.3 Разработка программы..... | 10 |
| 1.2.4 Отладка..... | 10 |
| 1.2.5 Внедрение..... | 10 |
| 1.3 Обзор существующих информационных систем для музеев..... | 11 |
| 1.3.1 «Музей-3»..... | 11 |
| 1.3.2 «НИКА-Музей»..... | 12 |
| 1.3.3 «КАМИС 5»..... | 12 |
| 1.3.4 Стоимость информационных систем..... | 13 |
| 1.3.5 Выводы..... | 13 |
| 2. ВЫБОР ПРОГРАММНЫХ СРЕДСТВ..... | 15 |
| 2.1 Выбор среды разработки программного обеспечения..... | 15 |
| 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ..... | 17 |
| 3.1 Дерево форм..... | 17 |
| 3.2 Форма аутентификации..... | 17 |
| 3.3 Форма выбора области..... | 19 |
| 3.4 Форма Справка..... | 20 |
| 3.5 Форма Музей..... | 20 |
| 3.6 Форма Бухгалтерия, финансы..... | 22 |
| 3.7 Форма Закупки..... | 24 |
| 3.8 Форма Фонды..... | 26 |
| 3.9 Форма Издательская и публикационная деятельность, реклама..... | 28 |

| | |
|--|--|
| 3.10 Форма Методическая работа | 30 |
| 3.11 Форма Научно-образовательная деятельность | 31 |
| 3.12 Форма Информационно-компьютерные технологии | 33 |
| 3.13 Универсальные процедуры для форм областей | 35 |
| ЗАКЛЮЧЕНИЕ..... | 38 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 39 |
| ПРИЛОЖЕНИЕ А | Ошибка! Закладка не определена. |
| ПРИЛОЖЕНИЕ Б | Ошибка! Закладка не определена. |
| ПРИЛОЖЕНИЕ В..... | Ошибка! Закладка не определена. |

ВВЕДЕНИЕ

С появлением доступных персональных компьютеров бухгалтерия практически каждого учреждения стала переходить в цифровой формат, так как это гораздо удобнее и практичнее бумажного. Отпала необходимость в хранении баз данных в бумажном виде, которые занимали невероятно большое количество места, и были попросту неудобными. Сейчас в каждом учреждении, будь то музей, детский сад, школа или что либо - другое существует своя электронная база данных.

Однако навигация по таким базам данных, поиск нужных документов не всегда удобна. При работе с базами данных очень важна удобность, так как хорошая организация может сохранить очень много времени и ресурсов.

Создание информационной системы поможет быстро ориентироваться во всех документах организации, так как таблицы будут содержать все необходимые сведения о них, а также будет возможность открытия документов через информационную систему.

Именно поэтому создание информационной системы «Музей» является темой ВКР.

Целью выпускной квалификационной работы является разработка и реализация программного обеспечения информационной системы «Музей».

Задачи, которые необходимо решить для достижения цели:

- исследовать предметную область;
- определиться с технологией разработки;
- ознакомиться с выбранной технологией разработки;
- разработать информационную систему «Музей»

Объект исследования – МБОУ Чусовской краеведческий музей.

Предмет исследования – информационная система, результатом которой является навигация по документам музея.

1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1 Информационные системы

Благодаря интенсивному развитию информационных технологий происходит расширение области их применения. Если раньше автоматизация бухгалтерского учета была чуть ли не единственной областью, в которой применялись информационные системы, то сейчас наблюдается интеграция информационных технологий в другие области.

Под информационной системой понимается программное обеспечение, предназначенное для сбора, хранения и обработки информации, ориентированное на конечного пользователя, не обладающего высокой квалификацией в области вычислительной техники. Поэтому клиентские приложения информационной системы должны обладать простым, удобным, легко осваиваемым интерфейсом, который предоставляет конечному пользователю все необходимые для работы функции и в то же время не дает ему возможность выполнять какие-то лишние действия.

К информационной системе предъявляются следующие требования:

- Гибкость: способность к адаптации и дальнейшему развитию подразумевают возможность приспособления информационной системы к новым условиям, к введению каких-либо новых областей документов, изменению количества пользователей и т.п.;
- Эффективность: система является эффективной, если с учётом выделенных ей ресурсов она позволяет решать возложенные на неё задачи в минимальные сроки.
- Безопасность: Под безопасностью, прежде всего, подразумевается свойство системы, в силу которого посторонние лица не имеют доступа к информационным ресурсам музея, кроме тех, которые для них предназначены[3].

1.2 Этапы разработки программного обеспечения ИС

1.2.1 Постановка задачи

ПО информационной системы «Музей» создается на основе технического задания (ПРИЛОЖЕНИЕ А)

1.2.2 Разработка пользовательского интерфейса

Важный этап разработки информационной системы - создание пользовательского интерфейса. Интерфейс пользователя – это та часть программы, с которой непосредственно взаимодействует пользователь.

Основной целью данного этапа является создание продуманного, понятного для пользователя интерфейса, взаимодействуя с которым конечный пользователь будет чувствовать себя комфортно.

1.2.3 Разработка программы

Разработка программы - процесс, представляющий собой поэтапное написание программы на выбранном языке программирования.

1.2.4 Отладка

Отладкой называют процесс тестирования программы, поиска и устранения ошибок.

Ошибки в программах бывают трех видов:

- Ошибки компиляции, также называемые ошибками компилятора, препятствуют запуску программы. Если компилятор получает код, который он не может понять, он выдает ошибку компилятора. Большинство ошибок компилятора вызваны ошибками при создании кода. Может быть ошибка в ключевом слове, неверная расстановка или отсутствие знаков препинания ,неверное использование операторов
- Ошибки во время выполнения — это ошибки, возникающие непосредственно во время работы программы. Большинство таких ошибок возникает тогда, когда программа пытается выполнить недопустимую операцию. Пример недопустимой операции – деление на ноль.
- Логические ошибки не позволяют программе выполнять предполагаемые действия. Ваша программа может компилироваться и выполняться без ошибок, но результат выполнения может оказаться неверным. Например: при сложении двух чисел не ввести второе число.

1.2.5 Внедрение

После отладки программы происходит передача программы заказчику, а также проводится обучение пользователей.

1.3 Обзор существующих информационных систем для музеев

Для обзора готовых информационных систем для музея были выбраны следующие программы:

- «Музей-3»
- «НИКА-Музей»
- «КАМИС 5»

1.3.1 «Музей-3»

«Музей-3» - информационная система, предназначенная для автоматизации деятельности музеев любого профиля.

Низкие требования к техническому оснащению объекта автоматизации, возможность установки локальной и сетевой версии, ограничение и профессиональное разделение прав доступа, наличие подробных инструкции и системы подсказок для удобства работы пользователей делают процесс автоматизации доступным для любого музея.

Функции системы:

- Проведение экспоната по всем ступеням учёта музейных предметов;
- Ведение электронного каталога предметов;
- Учет письменных источников по правилам ГАУ;
- Специальный учет музейных предметов, содержащих драгоценные металлы и камни;
- Учёт движения музейных предметов;
- Учет изменения сохранности предмета;
- Учет сведений о реставрационных работах;
- Ведение авторских картотек и других справочников;
- Формирование учетных книг;
- Составление коллекционных описей;
- Печать карточек с описаниями предметов;
- Получение статистической информации по БД;
- Ведение архива научных исследований музея[9].

1.3.2 «НИКА-Музей»

Современная комплексная информационно-справочная система, которая предназначена для решения широкого спектра задач, возникающих в деятельности любого музея.

Особенности и функции:

- учет приема, выдачи и движения музейных предметов;
- создание необходимой учетно-хранительской документации;
- всестороннее описание музейных предметов;
- создание и поддержание видеоохранного банка изображений музейных предметов;
- создание сколь угодно объемной базы данных музейных коллекций, включающую не только описание и изображения музейных предметов, но и описание и хранение связанных с ними исторических событий, исторических лиц, библиографии, географических объектов, видео и аудиоматериалов, выставок и т.п.;
- эффективный поиск предметов и связанных с ними материалов по различным критериям (реквизитам, произвольному описанию) с учетом морфологии русского языка;
- эффективный поиск и отбор предметов для выставок;
- подготовку сопроводительной выставочной документации;
- поддержку современных способов работы с посетителями (посредством Интернет, информационные интерактивные киоски, каталоги и музейные мультимедийные продукты на CD-ROM);
- создание сводных коллекций;
- гибкую настройку под специфику каждого музея[10].

1.3.3 «КАМИС 5»

Функции информационной системы «КАМИС 5»:

- Учет и хранение
- Каталогизация
- Подготовка выставок и публикаций
- Реставрация
- Создание систем для посетителей
- Публикация коллекции в Интернет

Особенности «КАМИС 5»:

- Трехзвенная архитектура: позволяет пользователям подключаться и работать через Интернет
- Работа в локальной сети с любого компьютера через браузер Google Chrome или через интернет
- Современный пользовательский интерфейс:
 - Просмотр всех картотек в виде табличных списков, списков с изображениями или мозаики изображений
 - Возможность настраивать количество отображаемых колонок и их ширину.
 - Проверка орфографии Автоподстановка при вводе значений через справочник
 - Усовершенствованные механизмы поиска: простой с кратким набором атрибутов и расширенный с полным списком атрибутов в алфавитном порядке
 - Выполнение продолжительных по времени задач (массовая загрузка изображений, печать больших документов) в асинхронном режиме, путем постановки в очередь[11].

1.3.4 Стоимость информационных систем

Таблица 1 – Стоимость информационных систем

| Наименование информационной системы | Стоимость лицензии (на год) |
|-------------------------------------|-----------------------------|
| Музей-3 | 12 500 рублей |
| НИКА-Музей | 9 000 рублей |
| КАМИС 5 | 96 000 рублей |

1.3.5 Выводы

Исходя из рассмотренного выше сделаем выводы, почему имеется необходимость в самостоятельном создании информационной системы для Чусовского краеведческого музея:

У Чусовского Краеведческого музея есть свои требования к информационной системе, которые заключаются в том, что информационная система должна решить проблему неудобной навигации в документах, а все рассмотренные выше готовые

информационные системы специализируются на интерактивности, взаимодействии с посетителями музея, и решают эту проблему учреждения лишь частично.

2. ВЫБОР ПРОГРАММНЫХ СРЕДСТВ

2.1 Выбор среды разработки программного обеспечения

Среда разработки программного обеспечения (ПО) — совокупность программных средств, используемая программистами для разработки программного обеспечения. Простая среда разработки включает в себя редактор текста, компилятор и/или интерпретатор, средства автоматизации сборки и отладчик. Когда эти компоненты собраны в единый программный комплекс, говорят об интегрированной среде разработки. Такая среда представлена одной программой, не выходя из которой можно производить весь цикл разработки[6].

Для разработки информационной системы «Музей» были рассмотрены следующие среды разработки ПО:

- Lazarus
- Borland Delphi 7
- Microsoft Visual Studio 15

Microsoft Visual Studio 15 не была выбрана по причине отсутствия опыта работы в ней.

И Lazarus, и Delphi поддерживают код Объектного Паскаля, используют визуальные и не визуальные компоненты, имеют схожий интерфейс. Однако у них имеется несколько различий:

- Lazarus, в отличие от Delphi, бесплатен.
- Lazarus кросс-платформенная IDE, то есть, поддерживает различные операционные системы.
- Lazarus имеет собственную библиотеку компонентов LCL (Lazarus Component Library), а Delphi - VCL (Visual Component Library). Однако VCL и LCL во многом так похожи, что программист при работе с компонентами почти не ощущает разницы. Часто (но не всегда) проекты, написанные на Delphi можно без потерь компилировать на Lazarus.

Для создания программного обеспечения была выбрана среда программирования Lazarus, при выборе данной среды разработки были рассмотрены такие аспекты как:

- низкие требования разработанного приложения к ресурсам компьютера;

- опыт работы в среде Lazarus, полученный в ходе прохождения учебной программы;
- распространение программы по лицензии GNU. GNU - General Public License — лицензия на свободное программное обеспечение, созданная в рамках проекта GNU в 1988 г., по которой автор передаёт программное обеспечение в общественную собственность;
- полностью юникодный (UTF-8) интерфейс и редактор среды Lazarus и поэтому отсутствие проблем с портированием кода, содержащего национальные символы.
- высокая производительность разработанного приложения;
- мощный редактор кода, включающий систему подсказок, гипертекстовую навигацию по исходным текстам, автозавершение кода и рефакторинг[7].

3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ

3.1 Дерево форм

Прежде чем начать этап написания программного кода необходимо составить дерево форма (рис. 1), для того чтобы знать структуру построения программы.

На рисунке 1 показана последовательность перехода между формами. При запуске программы появляется окно аутентификации, затем появляется меню выбора области (по сути – главное меню). С формы выбора областей возможен переход в любую интересующую нас область.

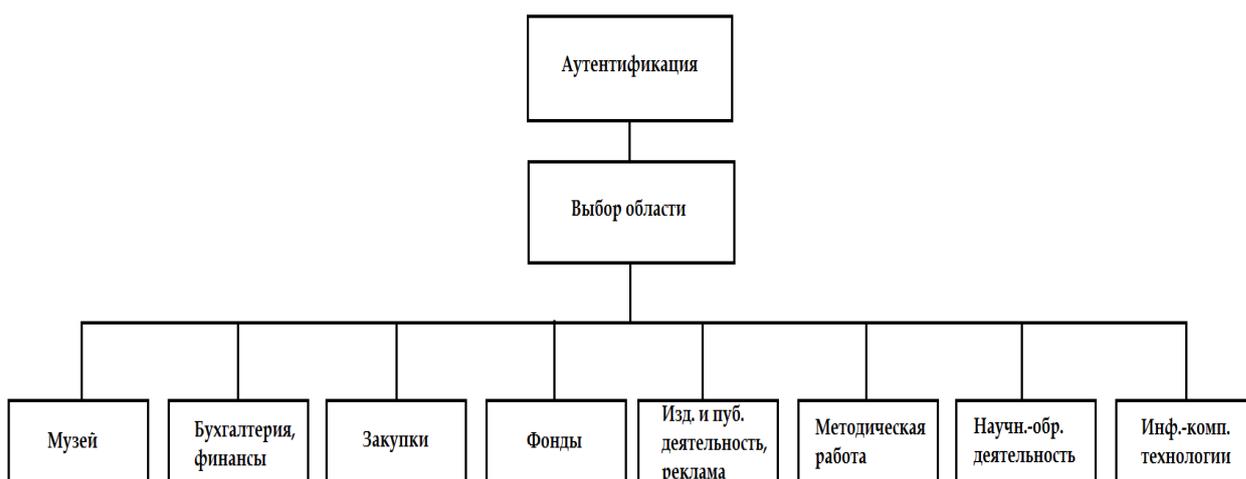


Рисунок 1 – Дерево форм

3.2 Форма аутентификации

Для проверки подлинности пользователей создадим форму аутентификации(рис.2). Доступ к ИС «Музей» будет осуществляться с помощью парольной аутентификации.

Процедура парольной аутентификации обеспечивает защиту данных содержащихся в информационной системе от несанкционированного воздействия. При входе в ИС «Музей» пользователь набирает на клавиатуре свой идентификатор и пароль. Если введенная пара идентификатор/пароль совпала с верной, то аутентификация прошла успешно. В противном случае на экран будет выведено сообщение «'Введен не верный логин или пароль, попробуйте еще раз!»

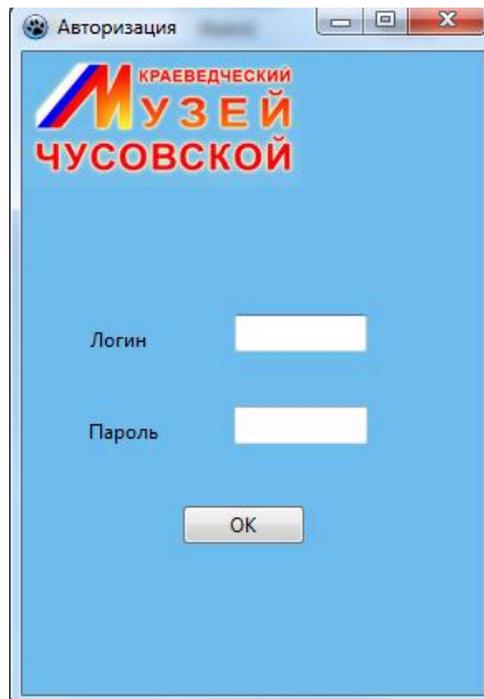


Рисунок 2 – форма аутентификации

Процедура проверки правильности ввода идентификатора и пароля.

```
procedure TForm1.Button1Click(Sender: TObject); // процедура нажатия на кнопку1
var id,pass:string;
begin
id:='Директор';// верная пара идентификатора и пароля
pass:='12345';
if (edit1.Text=id) and (edit2.Text=pass) then
begin
Form2.Visible:=true;
Form1.Visible:=false; // при верном вводе открывается форма выбора области
end
else //в случае неверного ввода
if ((edit1.Text<>id) and (edit2.Text<>pass))
or ((edit1.Text=id) and (edit2.Text<>pass))
or ((edit1.Text<>id) and (edit2.Text=pass)) then
ShowMessage('Введен не верный логин или пароль, попробуйте еще раз!');
end;
```

3.3 Форма выбора области

После успешной аутентификации откроется форма выбора области(по сути главное меню программы)(рис.3). С помощью этой формы можно будет перейти к формам областей, а именно к таким формам, как:

- Музей
- Бухгалтерия, финансы
- Закупки
- Фонды
- Издательская и публикационная деятельность, реклама
- Методическая работа
- Научно-образовательная деятельность
- Информационно-компьютерные технологии

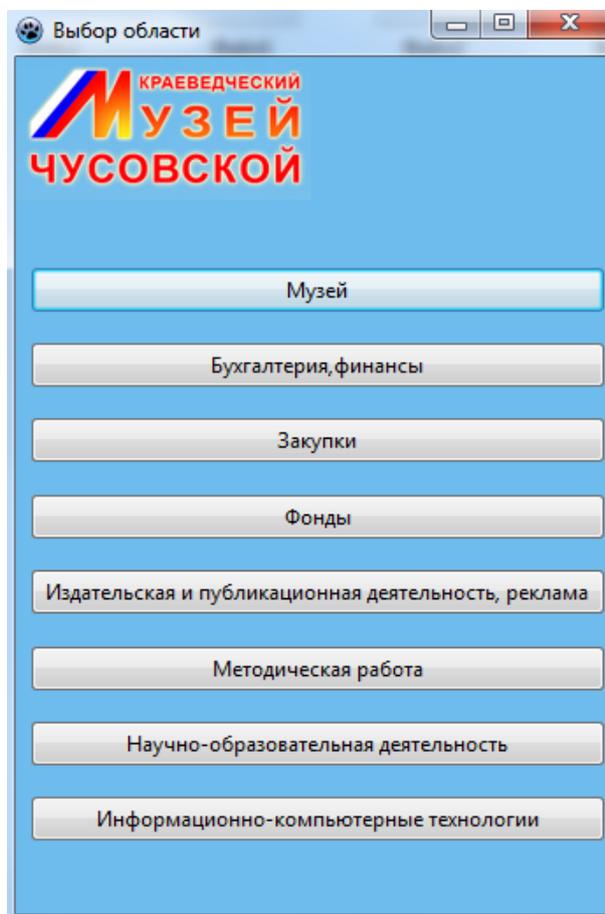


Рисунок 3 – форма выбора областей

Процедура закрытия формы аутентификации
procedure TForm2.FormCreate(Sender: TObject);
begin

```
Form2.Visible:=false; // Если открывается форма выбора областей, форма аутентификации  
закрывается  
end;
```

Процедура перехода на область

```
procedure TForm2.Button1Click(Sender: TObject);  
begin
```

```
Form31.Visible:=true; // при нажатии на кнопку области, открывается форма области  
end;
```

3.4 Форма Справка

Для более комфортного пользования информационной системой создадим форму «Справка»(рис.4).

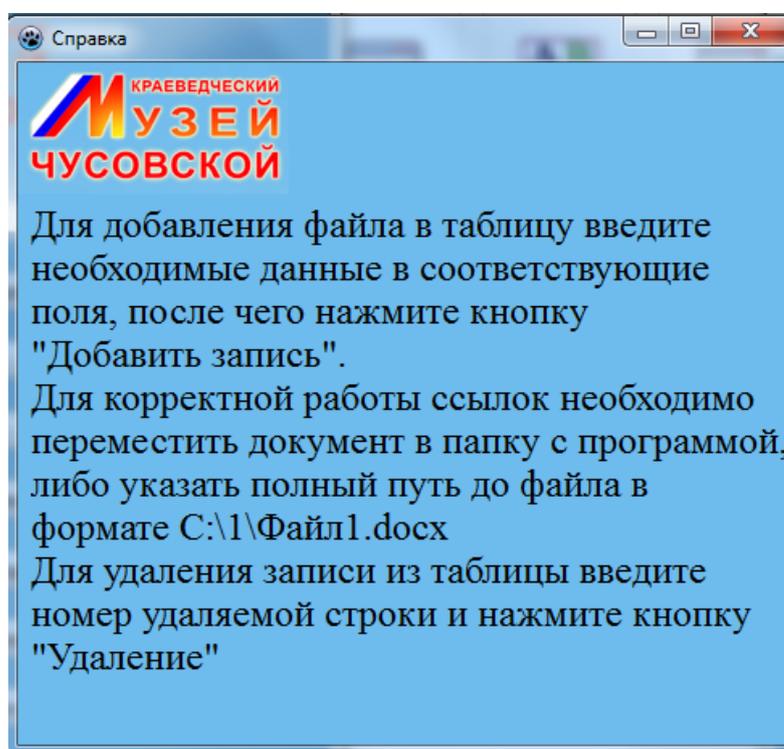


Рисунок 4 – форма Справка

3.5 Форма Музей

На данной форме находятся документы, связанные с областью «Музей», а также инструменты добавления и удаления записей(рис.5). Также на форме присутствует кнопка вызова справки.

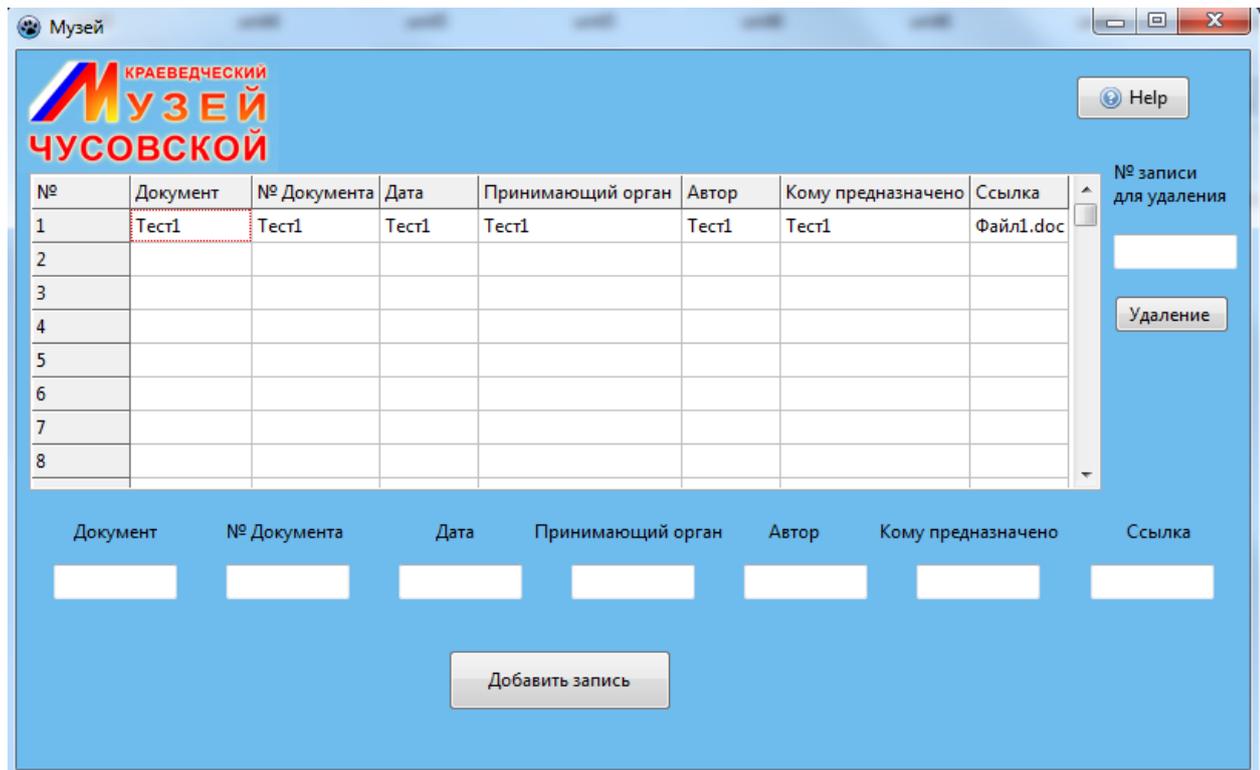


Рисунок 5 – форма Музей

Процедура создания формы

```
TForm31.FormCreate(Sender: TObject);
```

```
var
```

```
  fil:textfile;
```

```
  temp, x, y: integer;
```

```
  tempstr: string;
```

```
begin
```

```
  Form31.Visible:=false; // изначальное условие невидимости формы
```

```
  assignfile(fil,'save1');// открытие файла с данными таблицы
```

```
  reset(fil);
```

```
  readln(fil, temp);
```

```
  stringgrid1.colcount := temp; // занесения количества рядов и строк таблицы во временную
```

```
  readln(fil, temp);           // переменную для корректного заполнения таблицы
```

```
  stringgrid1.rowcount := temp;
```

```
  for X := 0 to stringgrid1.colcount - 1 do //условия цикла заполнения таблицы
```

```
    for y := 0 to stringgrid1.rowcount - 1 do
```

```
      begin // цикл заполнения таблицы
```

```
        readln(fil, tempstr);
```

```
stringgrid1.cells[x, y] := tempstr;  
end;  
closefile(fil); //закрытие файла  
end;
```

Процедура сохранение данных в таблице

```
procedure TForm31.FormClose(Sender: TObject; var CloseAction: TCloseAction);  
var // процедура создана с помощью события закрытия формы  
fil: textfile;  
x, y: integer;  
begin  
assignfile(fil,'save1'); // открытие файла с сохраненными данными  
rewrite(fil); // перезапись файла  
writeln(fil, stringgrid1.colcount);  
writeln(fil, stringgrid1.rowcount);  
for X := 0 to stringgrid1.colcount - 1 do // циклы заполнения таблицы  
  for y := 0 to stringgrid1.rowcount - 1 do  
    writeln(fil, stringgrid1.cells[x, y]); // заполнение таблицы  
closefile(fil); // закрытие файла
```

3.6 Форма Бухгалтерия, финансы

На данной форме находятся документы, связанные с областью «Бухгалтерия, финансы», а также инструменты добавления и удаления записей(рис.6). Также на форме присутствует кнопка вызова справки.

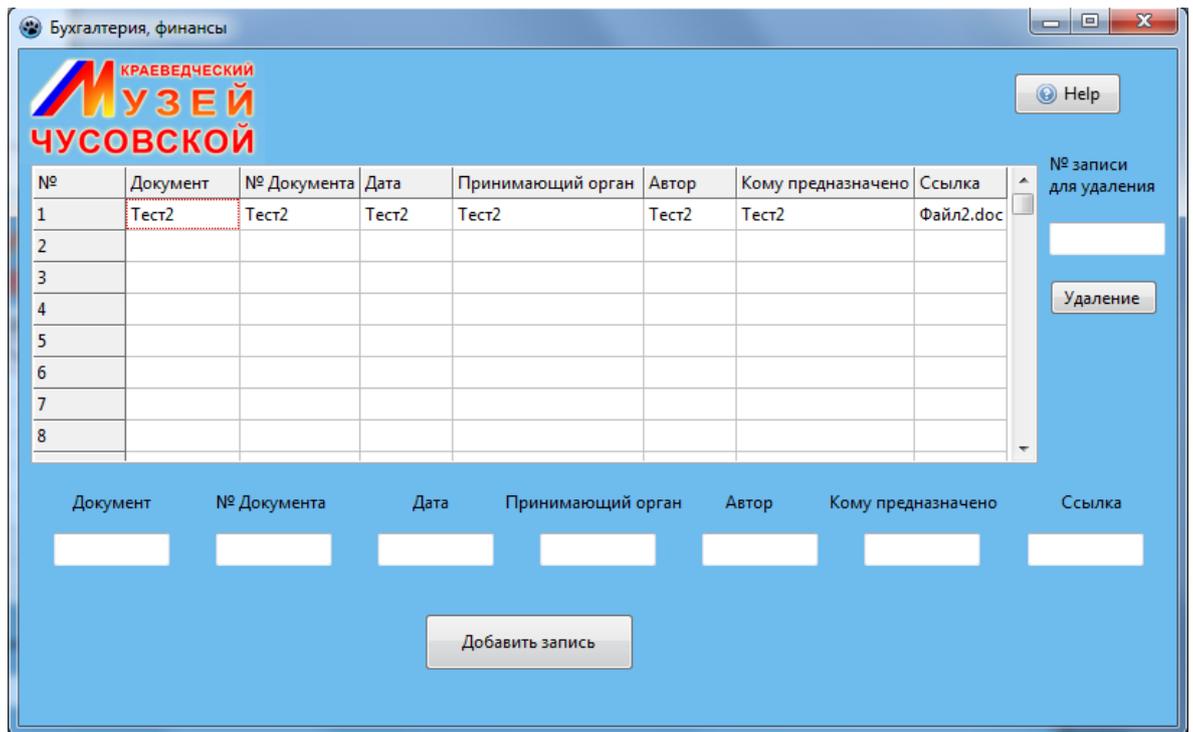


Рисунок 6 – форма Бухгалтерия, финансы

Процедура создания формы

```
TForm32.FormCreate(Sender: TObject);
```

```
var
```

```
  fil:textfile;
```

```
  temp, x, y: integer;
```

```
  tempstr: string;
```

```
begin
```

```
  Form32.Visible:=false; // изначальное условие невидимости формы
```

```
  assignfile(fil,'save2');// открытие файла с данными таблицы
```

```
  reset(fil);
```

```
  readln(fil, temp);
```

```
  stringgrid1.colcount := temp; // занесения количества рядов и строк таблицы во временную
```

```
  readln(fil, temp); // переменную для корректного заполнения таблицы
```

```
  stringgrid1.rowcount := temp;
```

```
  for X := 0 to stringgrid1.colcount - 1 do //условия цикла заполнения таблицы
```

```
    for y := 0 to stringgrid1.rowcount - 1 do
```

```
      begin // цикл заполнения таблицы
```

```
        readln(fil, tempstr);
```

```
        stringgrid1.cells[x, y] := tempstr;
```

```
end;  
closefile(fil); //закрытие файла  
end;
```

Процедура сохранение данных в таблице

```
procedure TForm32.FormClose(Sender: TObject; var CloseAction: TCloseAction);  
var // процедура создана с помощью события закрытия формы  
    fil: textfile;  
    x, y: integer;  
begin  
    assignfile(fil,'saves2'); // открытие файла с сохраненными данными  
    rewrite(fil); // перезапись файла  
    writeln(fil, stringgrid1.colcount);  
    writeln(fil, stringgrid1.rowcount);  
    for X := 0 to stringgrid1.colcount - 1 do // циклы заполнения таблицы  
        for y := 0 to stringgrid1.rowcount - 1 do  
            writeln(fil, stringgrid1.cells[x, y]); // заполнение таблицы  
        closefile(fil); // закрытие файла
```

3.7 Форма Закупки

На данной форме находятся документы, связанные с областью «Закупки», а также инструменты добавления и удаления записей(рис.7). Также на форме присутствует кнопка вызова справки.

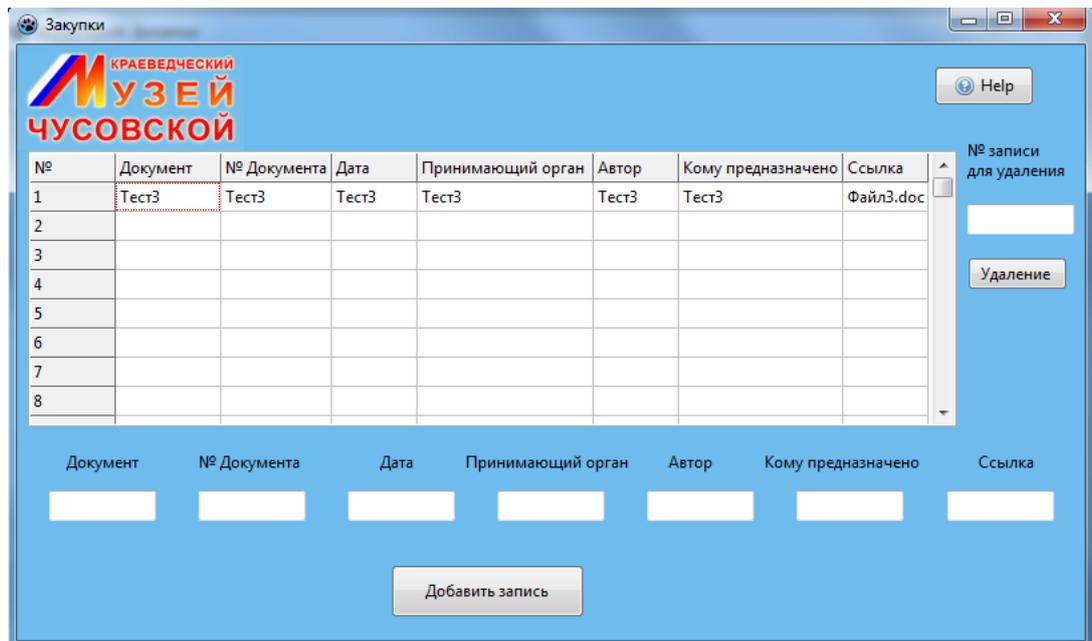


Рисунок 7 – форма Закупки

Процедура создания формы

```
TForm33.FormCreate(Sender: TObject);
```

```
var
```

```
  fil:textfile;
```

```
  temp, x, y: integer;
```

```
  tempstr: string;
```

```
begin
```

```
  Form33.Visible:=false; // изначальное условие невидимости формы
```

```
  assignfile(fil,'save3');// открытие файла с данными таблицы
```

```
  reset(fil);
```

```
  readln(fil, temp);
```

```
  stringgrid1.colcount := temp; // занесения количества рядов и строк таблицы во временную
```

```
  readln(fil, temp);          // переменную для корректного заполнения таблицы
```

```
  stringgrid1.rowcount := temp;
```

```
  for X := 0 to stringgrid1.colcount - 1 do //условия цикла заполнения таблицы
```

```
    for y := 0 to stringgrid1.rowcount - 1 do
```

```
      begin // цикл заполнения таблицы
```

```
        readln(fil, tempstr);
```

```
        stringgrid1.cells[x, y] := tempstr;
```

```
      end;
```

```
closefile(fil); //закрытие файла
end;
```

Процедура сохранение данных в таблице

```
procedure TForm33.FormClose(Sender: TObject; var CloseAction: TCloseAction);
var // процедура создана с помощью события закрытия формы
    fil: textfile;
    x, y: integer;
begin
    assignfile(fil,'save3'); // открытие файла с сохраненными данными
    rewrite(fil); // перезапись файла
    writeln(fil, stringgrid1.colcount);
    writeln(fil, stringgrid1.rowcount);
    for X := 0 to stringgrid1.colcount - 1 do // циклы заполнения таблицы
        for y := 0 to stringgrid1.rowcount - 1 do
            writeln(fil, stringgrid1.cells[x, y]); // заполнение таблицы
        closefile(fil); // закрытие файла
    end;
end;
```

3.8 Форма Фонды

На данной форме находятся документы, связанные с областью «Фонды», а также инструменты добавления и удаления записей(рис.8). Также на форме присутствует кнопка вызова справки.

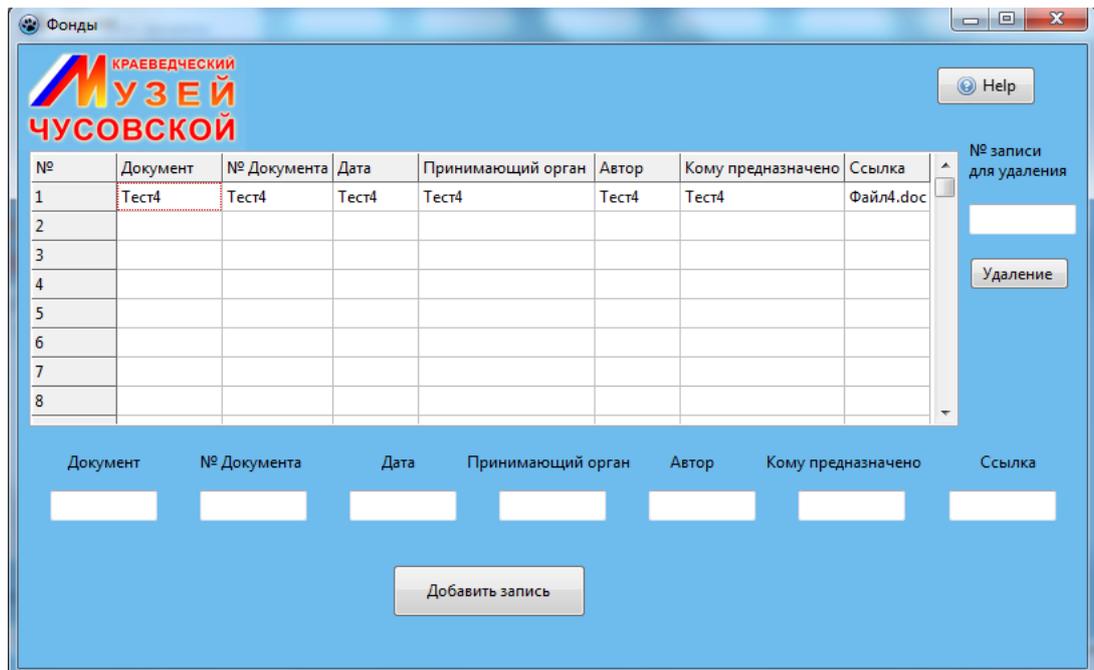


Рисунок 8 – форма Фонды

Процедура создания формы

```
TForm34.FormCreate(Sender: TObject);
```

```
var
```

```
  fil:textfile;
```

```
  temp, x, y: integer;
```

```
  tempstr: string;
```

```
begin
```

```
  Form34.Visible:=false; // изначальное условие невидимости формы
```

```
  assignfile(fil,'save4');// открытие файла с данными таблицы
```

```
  reset(fil);
```

```
  readln(fil, temp);
```

```
  stringgrid1.colcount := temp; // занесения количества рядов и строк таблицы во временную
```

```
  readln(fil, temp); // переменную для корректного заполнения таблицы
```

```
  stringgrid1.rowcount := temp;
```

```
  for X := 0 to stringgrid1.colcount - 1 do //условия цикла заполнения таблицы
```

```
    for y := 0 to stringgrid1.rowcount - 1 do
```

```
      begin // цикл заполнения таблицы
```

```
        readln(fil, tempstr);
```

```
        stringgrid1.cells[x, y] := tempstr;
```

```
      end;
```

```
  closefile(fil); //закрытие файла
```

end;

Процедура сохранение данных в таблице

```
procedure TForm34.FormClose(Sender: TObject; var CloseAction: TCloseAction);
var // процедура создана с помощью события закрытия формы
    fil: textfile;
    x, y: integer;
begin
    assignfile(fil,'save4'); // открытие файла с сохраненными данными
    rewrite(fil); // перезапись файла
    writeln(fil, stringgrid1.colcount);
    writeln(fil, stringgrid1.rowcount);
    for X := 0 to stringgrid1.colcount - 1 do // циклы заполнения таблицы
        for y := 0 to stringgrid1.rowcount - 1 do
            writeln(fil, stringgrid1.cells[x, y]); // заполнение таблицы
    closefile(fil); // закрытие файла
```

3.9 Форма Издательская и публикационная деятельность, реклама

На данной форме находятся документы, связанные с областью «Издательская и публикационная деятельность, реклама», а также инструменты добавления и удаления записей(рис.9). Также на форме присутствует кнопка вызова справки.

| № | Документ | № Документа | Дата | Принимающий орган | Автор | Кому предназначено | Ссылка |
|---|----------|-------------|-------|-------------------|-------|--------------------|-----------|
| 1 | Тест5 | Тест5 | Тест5 | Тест5 | Тест5 | Тест5 | Файл5.doc |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |

Рисунок 9 – форма Издательская и публикационная деятельность, реклама

Процедура создания формы

```
TForm35.FormCreate(Sender: TObject);  
var  
  fil:textfile;  
  temp, x, y: integer;  
  tempstr: string;  
begin  
  Form35.Visible:=false; // изначальное условие невидимости формы  
  assignfile(fil,'save5');// открытие файла с данными таблицы  
  reset(fil);  
  readln(fil, temp);  
  stringgrid1.colcount := temp; // занесения количества рядов и строк таблицы во временную  
  readln(fil, temp);          // переменную для корректного заполнения таблицы  
  stringgrid1.rowcount := temp;  
  for X := 0 to stringgrid1.colcount - 1 do //условия цикла заполнения таблицы  
    for y := 0 to stringgrid1.rowcount - 1 do  
      begin // цикл заполнения таблицы  
        readln(fil, tempstr);  
        stringgrid1.cells[x, y] := tempstr;  
      end;  
    closefile(fil); //закрытие файла  
  end;
```

Процедура сохранение данных в таблице

```
procedure TForm35.FormClose(Sender: TObject; var CloseAction: TCloseAction);  
var // процедура создана с помощью события закрытия формы  
  fil: textfile;  
  x, y: integer;  
begin  
  assignfile(fil,'save5');// открытие файла с сохраненными данными  
  rewrite(fil); // перезапись файла  
  writeln(fil, stringgrid1.colcount);  
  writeln(fil, stringgrid1.rowcount);  
  for X := 0 to stringgrid1.colcount - 1 do // циклы заполнения таблицы
```

```

for y := 0 to stringgrid1.rowcount - 1 do
  writeln(fil, stringgrid1.cells[x, y]); // заполнение таблицы
closefile(fil); // закрытие файла

```

3.10 Форма Методическая работа

На данной форме находятся документы, связанные с областью «Методическая работа», а также инструменты добавления и удаления записей(рис.10). Также на форме присутствует кнопка вызова справки.

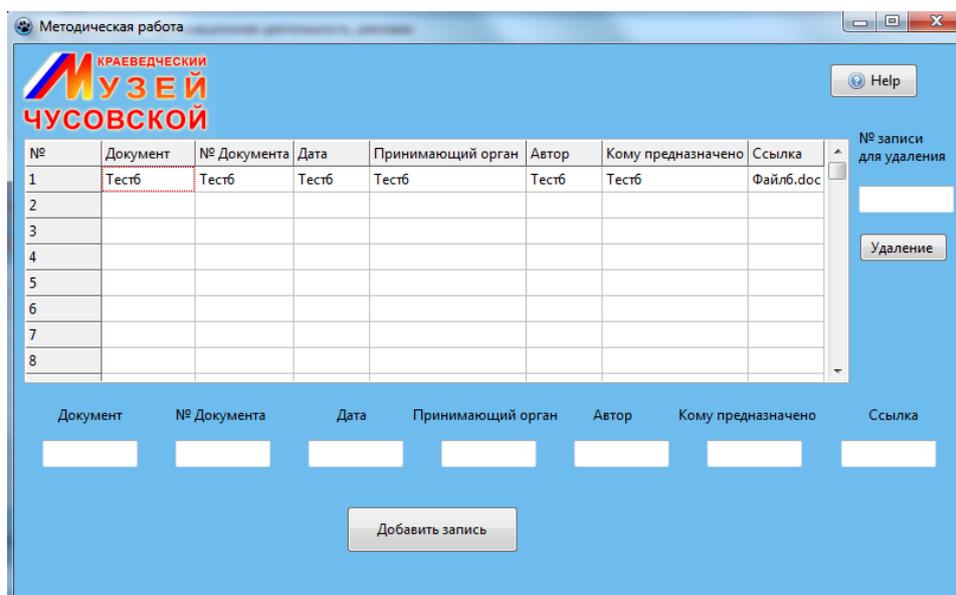


Рисунок 10 – форма Методическая работа

Процедура создания формы

```

 TForm36.FormCreate(Sender: TObject);
var
  fil:textfile;
  temp, x, y: integer;
  tempstr: string;
begin
  Form36.Visible:=false; // изначальное условие невидимости формы
  assignfile(fil,'save6'); // открытие файла с данными таблицы
  reset(fil);
  readln(fil, temp);
  stringgrid1.colcount := temp; // занесения количества рядов и строк таблицы во временную
  readln(fil, temp);           // переменную для корректного заполнения таблицы

```

```

stringgrid1.rowcount := temp;
for X := 0 to stringgrid1.colcount - 1 do //условия цикла заполнения таблицы
  for y := 0 to stringgrid1.rowcount - 1 do
    begin // цикл заполнения таблицы
      readln(fil, tempstr);
      stringgrid1.cells[x, y] := tempstr;
    end;
  closefile(fil); //закрытие файла
end;

```

Процедура сохранение данных в таблице

```

procedure TForm36.FormClose(Sender: TObject; var CloseAction: TCloseAction);
var // процедура создана с помощью события закрытия формы
  fil: textfile;
  x, y: integer;
begin
  assignfile(fil,'save6'); // открытие файла с сохраненными данными
  rewrite(fil); // перезапись файла
  writeln(fil, stringgrid1.colcount);
  writeln(fil, stringgrid1.rowcount);
  for X := 0 to stringgrid1.colcount - 1 do // циклы заполнения таблицы
    for y := 0 to stringgrid1.rowcount - 1 do
      writeln(fil, stringgrid1.cells[x, y]); // заполнение таблицы
  closefile(fil); // закрытие файла

```

3.11 Форма Научно-образовательная деятельность

На данной форме находятся документы, связанные с областью «Научно – образовательная деятельность», а также инструменты добавления и удаления записей(рис.11). Также на форме присутствует кнопка вызова справки.

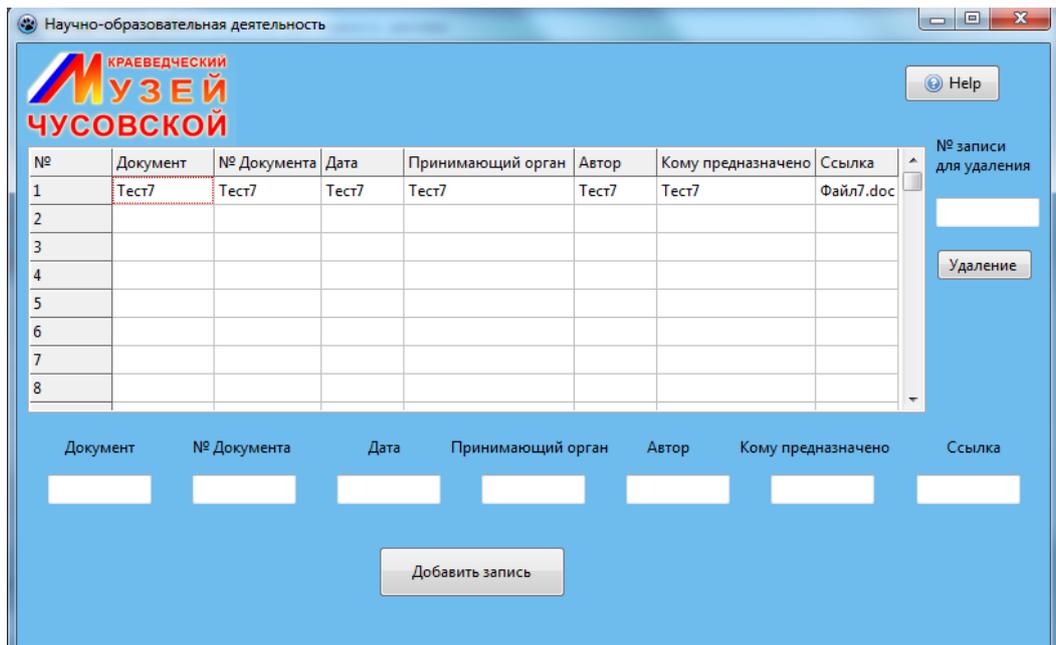


Рисунок 11 – форма Научно-образовательная форма

Процедура создания формы

```
TForm37.FormCreate(Sender: TObject);
```

```
var
```

```
  fil:textfile;
```

```
  temp, x, y: integer;
```

```
  tempstr: string;
```

```
begin
```

```
  Form37.Visible:=false; // изначальное условие невидимости формы
```

```
  assignfile(fil,'save7');// открытие файла с данными таблицы
```

```
  reset(fil);
```

```
  readln(fil, temp);
```

```
  stringgrid1.colcount := temp; // занесения количества рядов и строк таблицы во временную
```

```
  readln(fil, temp); // переменную для корректного заполнения таблицы
```

```
  stringgrid1.rowcount := temp;
```

```
  for X := 0 to stringgrid1.colcount - 1 do //условия цикла заполнения таблицы
```

```
    for y := 0 to stringgrid1.rowcount - 1 do
```

```
      begin // цикл заполнения таблицы
```

```
        readln(fil, tempstr);
```

```
        stringgrid1.cells[x, y] := tempstr;
```

```
      end;
```

```
closefile(fil); //закрытие файла
end;
```

Процедура сохранение данных в таблице

```
procedure TForm37.FormClose(Sender: TObject; var CloseAction: TCloseAction);
var // процедура создана с помощью события закрытия формы
    fil: textfile;
    x, y: integer;
begin
    assignfile(fil,'save7'); // открытие файла с сохраненными данными
    rewrite(fil); // перезапись файла
    writeln(fil, stringgrid1.colcount);
    writeln(fil, stringgrid1.rowcount);
    for X := 0 to stringgrid1.colcount - 1 do // циклы заполнения таблицы
        for y := 0 to stringgrid1.rowcount - 1 do
            writeln(fil, stringgrid1.cells[x, y]); // заполнение таблицы
    closefile(fil); // закрытие файла
```

3.12 Форма Информационно-компьютерные технологии

На данной форме находятся документы, связанные с областью «Информационно-компьютерные технологии», а также инструменты добавления и удаления записей(рис.12). Также на форме присутствует кнопка вызова справки.

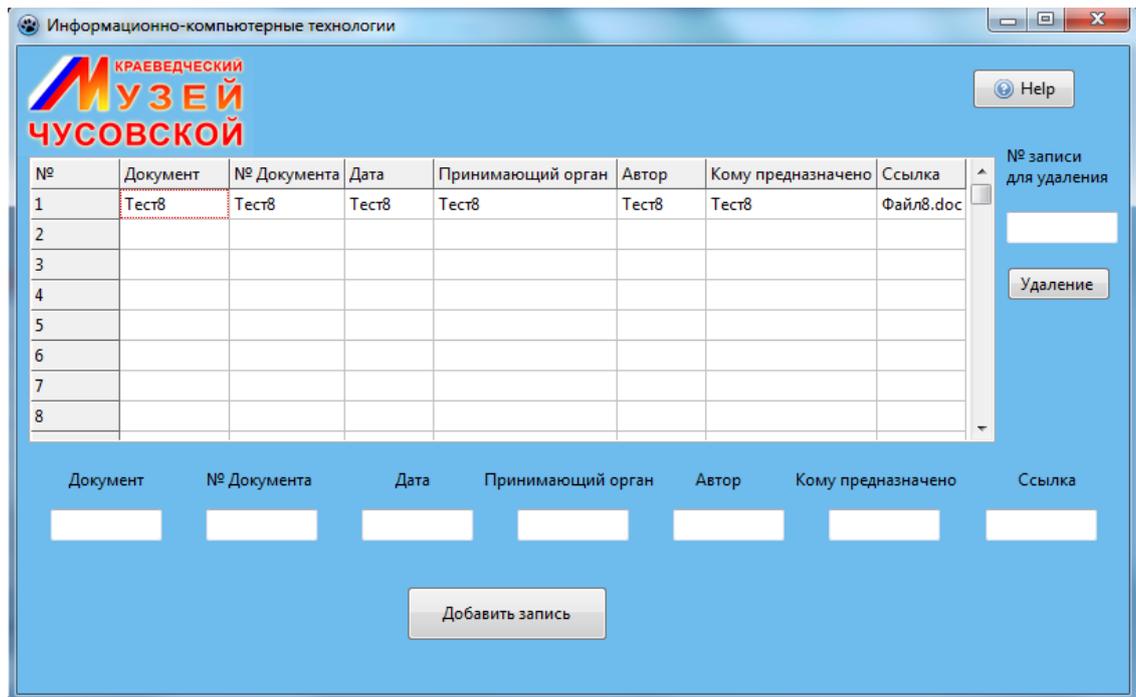


Рисунок 12 – форма Информационно-компьютерные технологии

Процедура создания формы

```
TForm38.FormCreate(Sender: TObject);
```

```
var
```

```
  fil:textfile;
```

```
  temp, x, y: integer;
```

```
  tempstr: string;
```

```
begin
```

```
  Form38.Visible:=false; // изначальное условие невидимости формы
```

```
  assignfile(fil,'save8');// открытие файла с данными таблицы
```

```
  reset(fil);
```

```
  readln(fil, temp);
```

```
  stringgrid1.colcount := temp; // занесения количества рядов и строк таблицы во временную
```

```
  readln(fil, temp); // переменную для корректного заполнения таблицы
```

```
  stringgrid1.rowcount := temp;
```

```
  for X := 0 to stringgrid1.colcount - 1 do //условия цикла заполнения таблицы
```

```
    for y := 0 to stringgrid1.rowcount - 1 do
```

```
      begin // цикл заполнения таблицы
```

```
        readln(fil, tempstr);
```

```
        stringgrid1.cells[x, y] := tempstr;
```

```
      end;
```

```
closefile(fil); //закрытие файла
end;
```

Процедура сохранение данных в таблице

```
procedure TForm38.FormClose(Sender: TObject; var CloseAction: TCloseAction);
var // процедура создана с помощью события закрытия формы
    fil: textfile;
    x, y: integer;
begin
    assignfile(fil,'save8'); // открытие файла с сохраненными данными
    rewrite(fil); // перезапись файла
    writeln(fil, stringgrid1.colcount);
    writeln(fil, stringgrid1.rowcount);
    for X := 0 to stringgrid1.colcount - 1 do // циклы заполнения таблицы
        for y := 0 to stringgrid1.rowcount - 1 do
            writeln(fil, stringgrid1.cells[x, y]); // заполнение таблицы
        closefile(fil); // закрытие файла
```

3.13 Универсальные процедуры для форм областей

Процедура открытия файла

```
procedure TForm31.StringGrid1SelectCell(Sender: TObject; aCol, aRow: Integer;
var CanSelect: Boolean); // процедура организована с помощью события выбора ячейки
var i,n:integer;
f:array[0..200] of string;
begin
begin
    n:=stringgrid1.rowcount; // создание временной переменной с номером ряда
    for i:=1 to n-1 do
begin //цикл, который получает номер столбца выбранной записи
    f[i]:=stringgrid1.cells[7,i];
    f[i]:=UTF8ToSys(f[i]); // перевод кодировки
```

```

if aCol=7 then // если был выбран 7ой столбец( столбец с ссылками), то происходит
begin          // открытие файла
if aRow=i then
ShellExecute(0,nil,PChar('C:\Program Files (x86)\Microsoft
Office\Office12\WINWORD'),PChar(f[i]),nil,1); // непосредственно открытие файла
end;
end;
end;
end;
end;

```

Процедура добавления записи

```

procedure TForm31.Button1Click(Sender: TObject);
var w,e,r:integer; // процедура создана с помощью события нажатия на кнопку
begin
e:=stringgrid1.RowCount; // подсчет количества строк
r:=0; // временная переменная, останавливающая цикл
for w:=1 to e-1 do // начало цикла заполнения строки данными
if stringgrid1.cells[1, w]="" then
begin
if r<>1 then
begin
stringgrid1.cells[1, w]:=edit1.text; // заполнение строки данными из соотв. полей
stringgrid1.cells[2, w]:=edit2.text;
stringgrid1.cells[3, w]:=edit3.text;
stringgrid1.cells[4, w]:=edit4.text;
stringgrid1.cells[5, w]:=edit5.text;
stringgrid1.cells[6, w]:=edit6.text;
stringgrid1.cells[7, w]:=edit7.text;
r:=1; // временная переменная останавливает цикл на добавлении одной строки
end;
end;
end;
end.

```

Процедура вызова формы Помощь

```
procedure TForm31.BitBtn1Click(Sender: TObject);
begin// процедура создана с помощью события нажатия на кнопку
Form3.visible:=true; // открывается форма Справка
end;
```

Процедура удаления записей

```
procedure TForm31.Button2Click(Sender: TObject);
var b:integer; // процедура создана с помощью события нажатия на кнопку
begin
b:=strtoint(edit8.text); // временная переменная с номером удаляемой строки
stringgrid1.Rows[b].clear; //удаление данных в строке
end;
```

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была разработана и реализована информационная система «Музей».

Для достижения поставленной цели был решен ряд задач:

- Проанализирована предметная область МБОУ Чусовской краеведческий музей;
- Определены принципы построения программного продукта;
- Реализовано создание и удаление сведений о документах;
- Реализована возможность открытия файлов непосредственно через программу;
- Реализована помощь пользователям в виде справки;
- Произведена разработка программного обеспечения.

В перспективе возможно добавление в информационную систему нескольких пользователей, других областей документов. Размер папки с информационной системой Музей - 26,6 МБ;

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»
2. «Методические указания по содержанию и оформлению выпускной квалификационной работы бакалавра» // Пермь – 2013 г.
3. Ю. Избачков, В. Петров «Учебное пособие Информационные системы» Второе издание // Санкт-Петербург ИД «Питер» – 2006 г.
4. Мансуров К.Т.. «Основы программирование в среде Lazarus.» Санкт-Петербург ИД «Питер» – 2010г.
5. Е.Р. Алексеев, О.В. Чеснокова, Т.В. Кучер. «Free Pascal и Lazarus: Учебник по программированию»// Москва ИД «ДМК-пресс» - 2010г.
6. Голицына О.Л., Максимов Н.В. «Информационные системы» // ММИЭИФП, 2004г.
7. Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. «Основы построения автоматизированных информационных систем»// ИНФРА-М, 2005
8. Е.В. Бурцева, И.П. Рак, А.В. Селезнев. «Информационные системы»// Издательство ТГТУ, 2009
9. Информационная система Музей-3 [электронный ресурс].
URL: <http://www.givc.ru/activity/software-complex/museum-system/> (дата обращения: 21.05.2017 г.)
10. Информационная система НИКА-Музей [электронный ресурс].
URL:http://mkrf.ru/presstsentr/novosti/region/detail.php?id=15308&sphrase_id=779496&t=sb(дата обращения: 21.05.2017 г.)
11. Информационная система КАМИС-5 [электронный ресурс].
URL: <http://kamis.ru/kamis/kamis-5/>(дата обращения: 21.05.2017 г.)
12. Среды разработки программного обеспечения: История и перспективы [электронный ресурс]
URL: <http://nit.miem.edu.ru/sbornik/2009/plen/008.html>(дата обращения: 21.05.2017 г.)
13. Lazarus about [электронный ресурс]
URL: <https://www.lazarus-ide.org/index.php?page=about> (дата обращения: 21.05.2017 г.)