



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

КОНТРОЛЬНАЯ РАБОТА

**по дисциплине «Реализация бизнес-процессов в
Информационных системах»**

на тему:

**«Разработка простейшего приложения с базой данных»
1 вариант**

Выполнила:

студент группы ИМО-МП-20

Барилко И.А.

Проверил:

Гусятинер Л.Б.

Оглавление

Введение, постановка задачи	2
Работа с таблицей.....	2
Клиентская часть.....	5
Заключение	7

Введение, постановка задачи

Предлагаемый вариант предполагает выделение ряда бизнес-процессов с последующим представлением. В данной контрольной работе будет продемонстрирован результат работы с одним из процессов.

В ходе работы над проектом по задаче ОАО "Экскаваторный завод "Ковровец"", согласно варианту, был разобран бизнес-процесс – работа с деталями и раскройка листов. Некоторые заготовки завода изготавливаются в заготовительном цехе на раскройном оборудовании с ЧПУ. Задача звучит следующим образом: «Отдел главного конструктора создает чертежи деталей и сохраняет их в формате dxf. После этого чертежи передаются по сети в отдел главного металлурга, там формируют размещение набора деталей на листе (раскрой), вычисляют коэффициент использования металла (КИМ, то есть отношение массы детали к массе заготовки). После этого по сети чертежи попадают в бюро ЧПУ заготовительного цеха, где разрабатываются программы для раскроя. По сети программа передается оператору станка.»

Нужно реализовать добавление строк в таблицу, изменение и удаление. Должна быть представлено создано оконное приложение.

Работа с таблицей

Было создано три таблицы – таблица листов, чертежей и таблица связей чертежей с листами. Работа велась с таблицей листов.

```
CREATE TABLE LISTS (
    ID_LIST    INTEGER NOT NULL,
    LIST_MASS  FLOAT NOT NULL,
    LIST_NAME  CHAR(20)
);
```

Листинг кода 1 – создание таблицы листов.

У таблицы есть такие поля, как id листа – ID_LIST, его масса – LIST_MASS, его название – LIST_NAME.

Для автоматической генерации id нужно создать генератор. Нужно помнить, что нумерация будет идти только вперед, даже если удалить какой-либо элемент.

Чтобы с таблицей можно было производить действия, обязательно нужно иметь возможность добавлять данные, изменять и удалять уже имеющиеся, а так же выполнять поиск по названию.

```
procedure TfLists.btNewClick(Sender: TObject);
// Вызов формы добавления новой строки
var
    Id_List: integer;
begin
    if fListsEd.ShowModal = mrOK then begin
        try
            with sp do begin
                if Active then Close;
                StoredProcName := 'LISTS_INS';
                with Params do begin
                    Clear;
                    CreateParam( ftFloat, 'List_Mass', ptInput ).AsFloat := StrToFloat(fListsEd.txtList_Mass.Text);
                    CreateParam( ftString, 'List_Name', ptInput ).AsString := fListsEd.txtList_Name.Text;
                    CreateParam( ftInteger, 'Id_List', ptOutput ).AsInteger;
                end; // with Params

                Prepare;
                ExecProc;
                dm.trans.Commit;
                Id_List := sp.ParamByName( 'Id_List' ).AsInteger;
            end; // with sp

            dm.trans.StartTransaction;
            //ds.DataSet.Close;
            ds.DataSet.Open;
            ds.DataSet.Locate('Id_List', Id_List, [] );
        except
            ShowMessage('Ошибка добавления!');
        end;
    end;
end;
```

Листинг кода 2 – добавление строки.

```

procedure TfLists.btEditClick(Sender: TObject);
// Вызов формы редактирования и корректировка строки
var
  Id_List: integer;
begin
  Id_List := ds.DataSet.FieldByName( 'Id_List' ).AsInteger;
  fListsEd.txtList_Mass.Text := ds.DataSet.FieldByName( 'List_Mass' ).AsString;
  fListsEd.txtList_Name.Text := ds.DataSet.FieldByName( 'List_Name' ).AsString;

  if fListsEd.ShowModal = mrOK then begin
    try
      with sp do begin
        if Active then Close;
        StoredProcName := 'LISTS_UPD';
        with Params do begin
          Clear;
          CreateParam( ftInteger, 'Id_List', ptInput ).AsInteger := ds.DataSet['Id_List'];
          CreateParam( ftFloat, 'List_Mass', ptInput ).AsFloat :=
            StrToFloat(fListsEd.txtList_Mass.Text);
          CreateParam( ftString, 'List_Name', ptInput ).AsString :=
            fListsEd.txtList_Name.Text;

          end; // with Params

          Prepare;
          ExecProc;
          dm.trans.Commit;
        end; // with sp

        dm.trans.StartTransaction;
        ds.DataSet.Open;
        ds.DataSet.Locate('Id_List', Id_List, [] );
      except
        ShowMessage('Ошибка редактирования!');
      end;
    end;
  end;
end;

```

Листинг кода 3 – редактирование и корректировка строки.

```

procedure TfLists.btDeleteClick(Sender: TObject);
// Удаление строки
begin
  if MessageDlg('Удалить', mtConfirmation, mbYesNoCancel, 0) <> mrYes then
    Abort
  else
    with sp do begin
      if Active then Close;
      StoredProcName:= 'LISTS_DEL';

      with Params do begin
        Clear;
        CreateParam(ftInteger, 'Id_List', ptInput).AsInteger := ds.DataSet['Id_List'];
      end; // with Params

      try
        Prepare;
        ExecProc;

        ds.DataSet.Close;
        ds.DataSet.Open;
      except
        ShowMessage('Удаление запрещено!');
      end; // with sp
    end; // if
  end;
end;

```

Листинг кода 4 – удаление строки.

```

procedure TForm5.Button1Click(Sender: TObject);
// Поиск по названию
begin
  if ( MaskEdit1.Text <> '' ) then
  begin
    if not ds.DataSet.Locate('List_Name', MaskEdit1.Text, [loPartialKey]) then
      ShowMessage('Не найдено');
    end;
  end;
end;

```

Листинг кода 5 – поиск по названию.

Клиентская часть

Клиентская часть реализовывается с использованием Lazarus.

Было создано оконное приложение windows forms.

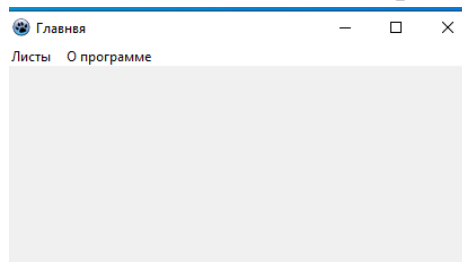


Рисунок 6 – оконное приложение. Главная.

При переходе в закладку листы – попадаем в таблицу.

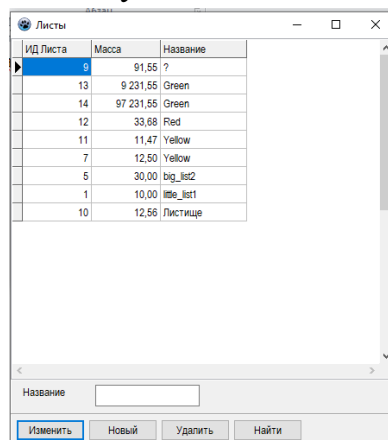


Рисунок 7 – таблица листов.

На рисунке 7 можно увидеть кнопки для добавления, изменения и удаления строк, а так же поле для поиска и кнопку.

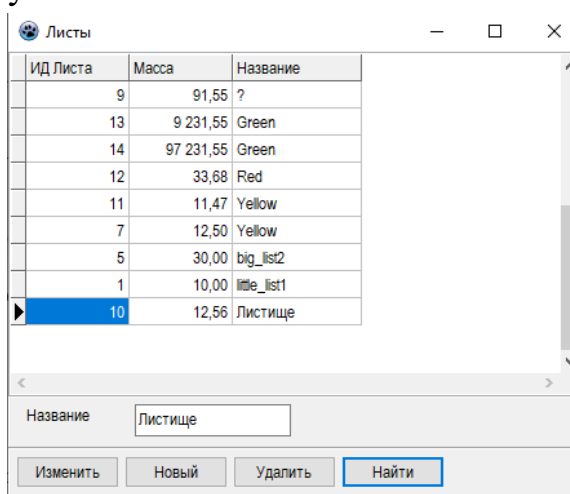


Рисунок 8 – пример работы поиска.

При вводе названия в поле для поиска и нажатии кнопки «найти», указатель встает на нужную строку. В случае, если нет строки, где используется введенное название, программа выдает ошибку.

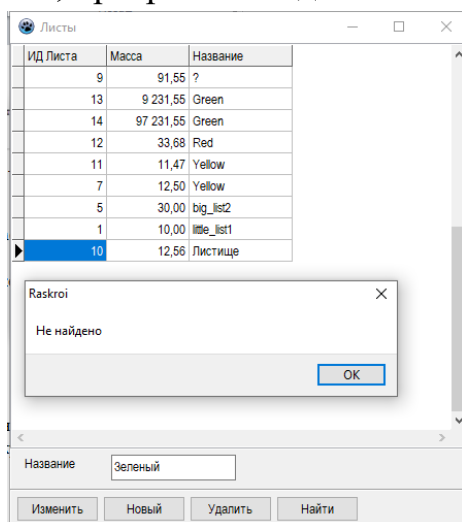


Рисунок 9 – пример поиска несуществующего наименования.

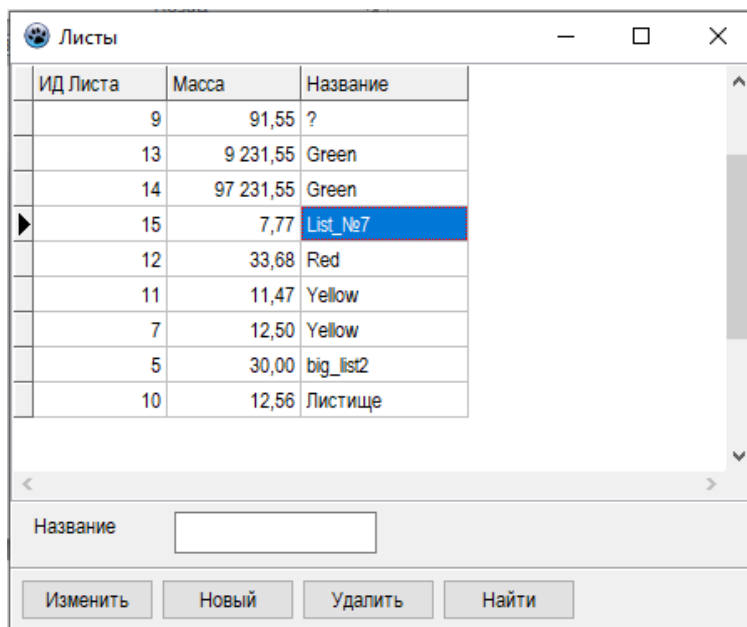


Рисунок 10. Удаление листа little_list1 и добавление листа List_№7 с массой 7,77.

Заключение

В ходе данной контрольной работы был исследован бизнес-процесс в предлагаемых рамках.

Были добавлены функции добавления, изменения и удаления строки, поиску по наименованию.

Сформировано клиентское приложение, позволяющее работать с таблицей Листов.

Продемонстрирована его работа на рисунках. Добавлены листинги кода.