

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
ДИМИТРОГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИЙ,
УПРАВЛЕНИЯ И ДИЗАЙНА

Курсовая работа по курсу: "ООП"
на тему:
"Проектирование БД"

Выполнил студент гр. ВТ-31: Потеренко А.Г.
Проверил преподаватель: Наскальнюк А.Н.

Димитровград 2005 г.

Содержание .

Задание для курсовой работы.....3

Проектирование таблиц БД.....4

Алгоритм решения поставленной задачи.....6

Тестовая БД.....9

Основные листинги программы.....10

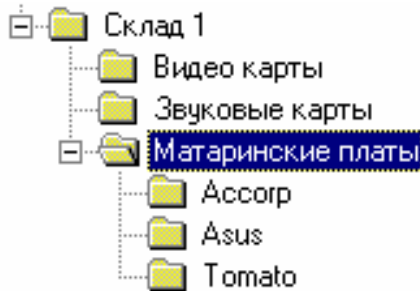
Задание для курсовой работы.

Спроектировать базу данных с помощью языка программирования **C++ Builder**.

В базе данных (БД) необходимо учесть следующие свойства товара:

- a) Наименование товара
- b) Цена товара (закупочная и розничная)
- c) Фирма, в которой куплен товар
- d) Дата закупки;
- e) Номер накладной;
- f) Серийный номер;
- g) Местонахождение (склад);
- h) Дата получения складом;

Вывод списка товаров необходимо сделать в следующем виде:



При выборе конечной группы товаров, всю группу необходимо вывести в отдельном окне.

Необходимо предусмотреть поиск товара по всем складам с заданным серийным номером.

А также серию отчетов позволяющую выводить отчеты о поступлении и продажи как отдельных товаров, так и группу товаров.

Пункт накладные.

Накладные расходные, приходные и накладная перемещения.

Расходная накладная выписывается при продаже товара. Приходная накладная выписывается при приобретении товара. Накладная перемещения при перемещения товара со склада на другой склад

Расходная накладная должна содержать:

Наименование товара, количество, цена, цена с учетом НДС, текущий курс \$; серийный номер, склад отправитель

Приходная накладная должна содержать:

Наименование товара, количество, цена, цена с учетом НДС, текущий курс \$; серийный номер, склад отправитель

Накладная перемещения должна содержать:

Наименование товара, количество, серийный номер, склад получатель, склад отправитель,

Проектирование таблиц БД.

Наша БД будет состоять из 8 таблиц формата "Paradox 7" (расширение файлов соответственно .DB). Пять таблиц будут хранить основные данные: *STRTAB1.DB*, *STRTAB2.DB*, *STRTAB3.DB*, *STRTAB4.DB*, *STRTAB5.DB* – данные хранятся с учетом целостности хранимых в них данных. Еще три: *NAKLADNAAPERМ.DB*, *NAKLADNAAPRIH.DB*, *NAKLADNAARASH.DB* – соответственно файл накладных перемещения, накладных прихода и накладных расхода. Если пользователь захочет иметь несколько таких БД, то ему придется хранить файлы (8 файлов) в разных каталогах, и настроить псевдонимы этих БД, лежащих в разных каталогах. Все файлы можно получить вместе с программой. Данные файлы создавались в *Database Desktop*.

Формат STRTAB1.DB:

Кодировка: **dBASE RUS cp866**

Поле №1: **NUM** "I" (тип Long Integer) (Key *) (Серийный номер товара)
Поле №2: **COD_TOVAR2** "I" (тип Long Integer) (Указатель на "категорию товара")
Поле №3: **COD_TOVAR3** "I" (тип Long Integer) (Указатель на "фирму, у которой купили товар")
Поле №4: **COD_TOVAR4** "I" (тип Long Integer) (Указатель на "склад, в котором хранится товар")
Поле №5: **COD_TOVAR5** "I" (тип Long Integer) (Указатель на "маркировка категории")
Поле №6: **ROZN** "N" (тип Number) (Розничная цена товара)
Поле №7: **ZAK** "N" (тип Number) (Закупочная цена товара)
Поле №8: **DATA** "D" (тип Date) (Дата закупки)
Поле №9: **DATA_SKLAD** "D" (тип Date) (Дата закупки)
Поле №10: **NAKL** "I" (тип Long Integer) (Номер накладной)

В данной таблице по полю "NUM" построен индекс (Unique, Maintained). Поэтому при попытке занести в таблицу два одинаковых значения возникнет исключение. Наша программа обрабатывает это событие проверкой на вновь создаваемую запись.

Данная таблица может как добавлять записи, так и удалять их.

Формат STRTAB2.DB, STRTAB3.DB, STRTAB4.DB, STRTAB5.DB:

Кодировка: **dBASE RUS cp866**

Поле №1: **COD** "+" (тип Autoincrement) (Key *) (Сюда ссылаются основные поля *STRTAB1.DB*)
Поле №2: **NAME** "A" (Size 255)

В данной таблице по полю "COD" построен индекс (Unique, Maintained). Поэтому при попытке занести в таблицу два одинаковых значения возникнет исключение. Наша программа обрабатывает это событие проверкой на вновь создаваемую запись.

Следует отметить то, что пользователь может только вносить данные в эти таблицы. Ручное редактирование таблиц возможно только с помощью альтернативной программы. Наша программа не предусматривает этой возможности. Пользователь может явно вносить записи в таблицы *STRTAB2.DB* и *STRTAB4.DB*. Неявно записи добавляются в таблицы *STRTAB3.DB* и *STRTAB5.DB*. При добавлении нового товара в БД – программа просматривает файл и узнает – есть ли уже данное поле с таким названием или нет. Соответственно, если оно есть – возвращается код, если нет – код новой записи и запись вносится в БД.

Формат NAKLADNAAPERМ.DB:

Кодировка: **dBASE RUS cp866**

Поле №1: **ID** "+" (Key *) (Построен индекс)
Поле №2: **NUM** "I" (тип Long Integer) (Серийный номер товара)
Поле №3: **VID** "A" (Size 255) (Маркировка товара)
Поле №4: **NAIMENOVANIE** "A" (Size 255) (Категория товара)
Поле №5: **SKLAD_POL** "A" (Size 255) (Склад получатель)
Поле №6: **SKLAD_OTPR** "A" (Size 255) (Склад отправитель)
Данные сюда заносятся, если товар перемещают по складам.

Формат NAKLADNAAPRIH.DB:

Кодировка: **dBASE RUS cp866**

Поле №1: **ID** "+" (Key *) (Построен индекс)
Поле №2: **NUM** "I" (тип Long Integer) (Серийный номер товара)
Поле №3: **VID** "A" (Size 255) (Маркировка товара)
Поле №4: **ZAK** "N" (тип Number) (Закупочная цена товара)
Поле №5: **NDS** "I" (тип Long Integer)
Поле №6: **DOLLAR** "N" (тип Number) (Курс доллара)
Поле №7: **NAIMENOVANIE** "A" (Size 255) (Категория товара)

Поле №8: **SKLAD_POLUCHATEL** "A" (Size 255) (Склад получатель)

Формат NAKLADNAARASH.DB:

Кодировка: **dBASE RUS cp866**

Поле №1: **ID** "+" (Key *) (Построен индекс)

Поле №2: **NUM** "I" (тип Long Integer) (Серийный номер товара)

Поле №3: **VID** "A" (Size 255) (Маркировка товара)

Поле №4: **ROZN** "N" (тип Number) (Розничная цена товара)

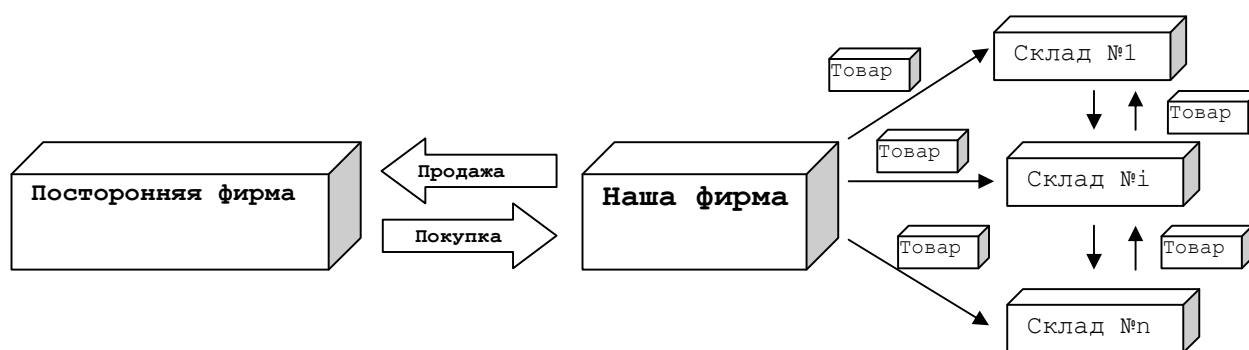
Поле №5: **NDS** "I" (тип Long Integer)

Поле №6: **DOLLAR** "N" (тип Number) (Курс доллара)

Поле №7: **NAIMENOVANIE** "A" (Size 255) (Категория товара)

Поле №8: **SKALD_OTPRAVITEL** "A" (Size 255) (Склад отправитель)

Логическое проектирование.



Алгоритм решения поставленной задачи.

Основная работа с БД будет проводиться с помощью технологии ADO. Поэтому будем использовать два компонента *TADOConnection* и *TADOTable*. В программе они создаются динамически и описываются в основном классе так:

```
TADOConnection *ADOC;  
TADOTable *ADOT1;  
TADOTable *ADOT2;  
TADOTable *ADOT3;  
TADOTable *ADOT4;  
TADOTable *ADOT5;  
TADOTable *RASH;  
TADOTable *PRIN;  
TADOTable *PERM;
```

Основной класс нашей программы: **TABL_CLASS**. В нем описываются все методы для работы с БД. Также мы будем использовать структуру:

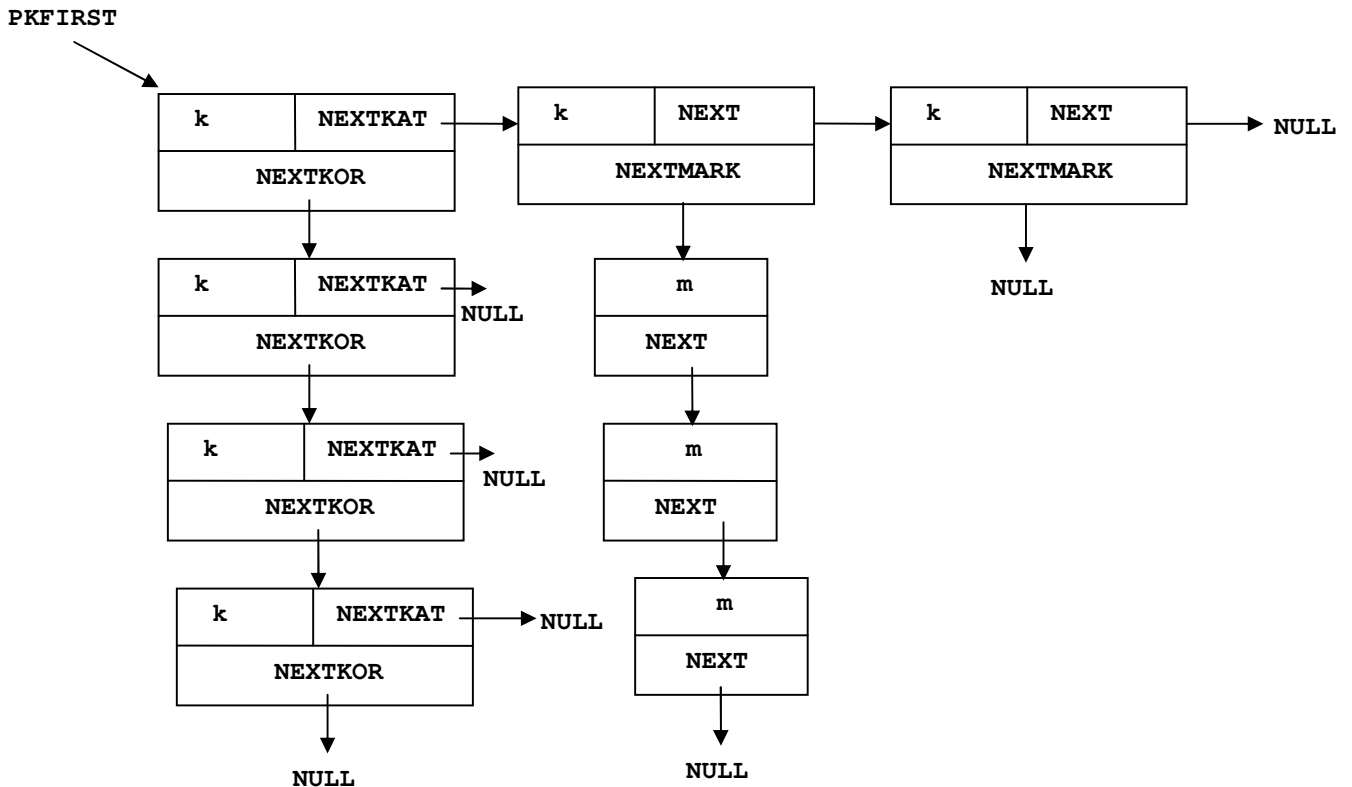
```
struct OKN_SP  
{  
    HWND HAND; //Идентификатор окна  
    OKN_SP * NEXT;  
};
```

Так как по заданию нам необходимо вывести по желанию пользователя в окне структуру любого склада, то нам необходимо хранить указатели на открытые окна. При необходимости пользователь может закрыть любое окно, а при выходе из программы один из методов закрывает все окна, уничтожая выделенную под них память.

Для вывода дерева структуры склада необходимо иметь следующие структуры, описываемые в главном классе так:

```
struct MARK  
{  
    int m;  
    MARK * NEXT;  
};  
struct KATEGOR  
{  
    int k;  
    KATEGOR * NEXT;  
    MARK * NEXTMARK;  
};  
struct KOREN  
{  
    int k;  
    KOREN * NEXTKOR;  
    KATEGOR * NEXTKAT;  
};
```

Структура дерева в памяти может выглядеть следующим образом:



Опишем основные методы класса программы:

void NOTCONNECT()	Отсоединяет от БД
boolean CONNECT(AnsiString Z)	Метод соединения с БД (Z - псевдоним)
boolean TABLE_ACTIV()	Метод создает ADOTable для работы с таблицами
void PROC_KATEGOR_ADD(TListBox * LIST, TADOTable * ADOPROC)	Добавляет в файл новые категории
void PROC_COMBO_READ(TComboBox *CB, TADOTable *ADOTAB)	Читаем из файлов в "Приобретение товара"
boolean WRITE_STRTAB1(int CT2, int CT3, int CT4, int CT5, int ROZN, int ZAK, int NAKL, int NUM, AnsiString DATA, AnsiString DATA_SKLAD);	Добавляет новый товар в БД при покупке
int FUNC_STRING_IN_INT(AnsiString S, TADOTable * ADO)	Возвращение кода по имени
void PROC_DEREVO(TADOTable * ADOTAB, TADOTable * ADOTAB1)	Строит дерево из таблицы №1 - дерево цифр (указателей)
void PROC_DEREVO_DELETE()	Удаляет дерево после использования
AnsiString FUNC_INT_IN_STRING(int i, TADOTable * ADO)	По имени - код
void TREE_VIEW(TTreeView * TV)	Просмотр товаров во всех складах
int SET_TREE_WINDWOW(TTreeView *TV, TTreeNode * TN1)	Выводит товары конкретного склада
void ADDINDEX(int j)	Добавляет в список чисел для отображения их в окне
void DELETEINDEX()	Удаляет список чисел для отображения
AnsiString GETSKLAD(int j)	Возвращение названия склада
TTreeNode * GETTREENODE()	Возвращает указатель на TN1
void SETTREENODE()	Необходимо для TN1=NULL при повторном открытии
int GETABSOLUTINDEX(TTreeNode * TNODE)	Возвращает абсолютный текущий индекс в дереве
void SETINDEXTREE(TTreeView *TV)	Устанавливает список всех выбранных узлов

```

void FIND(TEdit *S1,TEdit *S2,TEdit *S3,TEdit *S4,
          TEdit *S5,TEdit *S6,TEdit *S7,TEdit *S8,
          TEdit *S9,TEdit *S10)      Поиск товара
void CHANGESKLAD(TEdit *S1,TEdit *S2,TEdit *S3,TComboBox *CB)
                                   Перемещение товара на другой склад
void DELETETOVAR(TEdit * ED1)      Продажа товара
void NAKLPERM(int NUM, AnsiString S1) Выписывает накладную перемещения
void NAKLPRIH(int NUM, AnsiString S1, AnsiString S2, AnsiString S3,
              float DOLLAR, int NDS, float ZAK) Выписывает накладную прихода
void NAKLRASH(int NUM, float DOLLAR, int NDS) Выписывает накладную расхода

```

Следующие методы работают для оформления отчета в формате HTML

```

void OTCHET_PRODAGA(AnsiString S, AnsiString KAT) Создает отчет по продаже
void OTCHET_PRODAGA_ONE(AnsiString S, AnsiString S1) Создает отчет по продаже одного товара
void FILESS(FILE * files, AnsiString BD, AnsiString BUFER)
void LINIA(FILE * files)
void MESSAGES(FILE * files, AnsiString MES)
void OTCHET_KATEGOR(AnsiString S, AnsiString KAT)
void OTCHET_ZAGOLOVOK(FILE * files, AnsiString S)
void OTCHET_PERM(AnsiString S, int NUM)
void OTCHET_POKUPKA_ONE(AnsiString S, AnsiString C)
void OTCHET_POKUPKA(AnsiString S, AnsiString C)
void OTCHET_POK_KATEGOR(AnsiString S, AnsiString C)
float PROC_DOLLAR() Возвращает курс доллара
int PROC_NDS() Возвращает текущий НДС
boolean SETNDSOLLAR(AnsiString DOL, AnsiString ND) Устанавливает НДС и курс доллара
void IMPORT(TEdit *Edi) Импортирование БД из ненормализованной таблицы PARADOX

```

Экземпляр класса описывается следующим образом:

```
TABL_CLASS TBL
```

Тестовая БД.

Основную работу программы можно проверить с помощью тестовой базы данных. Она прилагается к основной программе. Поэтому описывать весь тест не имеет смысла – пользователь сможет в самом начале наглядно увидеть работоспособность программы.

Основные листинги программы.

Unit1.cpp

```

////////////////////////////////////+OK+////////////////////////////////////
#include <vcl.h>
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit4.h"
#include "Unit5.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit8.h"
#include "Unit9.h"
#include <stdio.h>
#include <dstring.h>;
#pragma hdrstop
#pragma package (smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
__fastcall TForm1::TForm1(TComponent* Owner): TForm(Owner){}
////////////////////////////////////+Реакция на событие изменения формы+////////////////////////////////////
void __fastcall TForm1::CHANGE(TObject *Sender, int &NewWidth,
    int &NewHeight, bool &Resize)
{
    TreeView1->Height=Form1->Height-255;
    GroupBox1->Height=Form1->Height-124;
}
////////////////////////////////////+Отсоединяемся от БД+////////////////////////////////////
void __fastcall TForm1::BitBtn4Click(TObject *Sender)
{
    TBL.NOTCONNECT();
    Form1->StatusBar1->Panels->Items[1]->Text= "Нет соединения с БД";
    Form1->BitBtn2->Enabled=false;
    Form1->BitBtn3->Enabled=false;
    Form1->BitBtn4->Enabled=false;
    Form1->BitBtn5->Enabled=false;
    Form1->BitBtn6->Enabled=false;
    Form1->BitBtn7->Enabled=false;
    Form1->BitBtn8->Enabled=false;
    Form1->BitBtn1->Enabled=true;
    GroupBox1->Visible=false;
    Form1->TreeView1->Items->Clear();
    //////////////////////////////////////Необходимо все убрать из памяти перед закрытием////////////////////////////////////
    CLEARWINDOW(); //Перед следующим просмотром необходимо все окна удалить
    TBL.SETTREENODE(); //Обнуляем указатель на NODE
    TBL.DELETEINDEX(); //Удаляем список чисел для выделения номера склада
}
////////////////////////////////////+Инициализация+////////////////////////////////////
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    Form1->StatusBar1->Panels->Items[1]->Text= "Нет соединения с БД";
    Form1->BitBtn2->Enabled=false;
    Form1->BitBtn3->Enabled=false;
    Form1->BitBtn4->Enabled=false;
    Form1->BitBtn5->Enabled=false;
    Form1->BitBtn6->Enabled=false;
    Form1->BitBtn7->Enabled=false;
    Form1->BitBtn8->Enabled=false;
    GroupBox1->Visible=false;
    Form1->Height=581;
    Form1->Width=905;
    Form1->Top=93;
    Form1->Left=86;
    Form1->FormStyle=fsMDIForm;
}
////////////////////////////////////+Работа с узлами+////////////////////////////////////
void __fastcall TForm1::TreeView1Collapsing(TObject *Sender,
    TTreeNode *Node, bool &AllowCollapse)
{
    Node->ImageIndex=0; //Свертывание
    Node->SelectedIndex=0;
}
void __fastcall TForm1::TreeView1Expanding(TObject *Sender,
    TTreeNode *Node, bool &AllowExpansion)
{
    Node->ImageIndex=1; //Развертывание
    Node->SelectedIndex=1;
}
////////////////////////////////////+Открываем соединение с базой данных+////////////////////////////////////
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
    Form5->Top=311;
    Form5->Left=307;
}

```

```

Form5->Edit1->Text="";
Form5->ShowModal();
if (TBL.CONNECT(Form5->Edit1->Text) && TBL.TABLE_ACTIV()) //Form5->Edit1->Text=Псевдоним
{
    Form1->StatusBar1->Panels->Items[1]->Text= "Вы работаете с БД: "+Form5->Edit1->Text;
    Form1->BitBtn2->Enabled=true;
    Form1->BitBtn3->Enabled=true;
    Form1->BitBtn4->Enabled=true;
    Form1->BitBtn5->Enabled=true;
    Form1->BitBtn6->Enabled=true;
    Form1->BitBtn7->Enabled=true;
    Form1->BitBtn8->Enabled=true;
    Form1->BitBtn1->Enabled=false;
    GroupBox1->Visible=true;
    Form1->TreeView1->Items->Clear();
    ShowMessage("Вам необходимо заполнить денежные параметры - курс $ и НДС!!!");
}
else
{
    ShowMessage("Не удалось установить соединение!");
};
}
//+Метод класса+
void TForm1::TABL_CLASS::NOTCONNECT()
{
    ADOC->Connected=false;
    ADOT1->Active=false;
    ADOT2->Active=false;
    ADOT3->Active=false;
    ADOT4->Active=false;
    ADOT5->Active=false;
    RASH->Active=false;
    PRIH->Active=false;
    PERM->Active=false;
    ADOC->Destroying();
    ADOT1->Destroying();
    ADOT2->Destroying();
    ADOT3->Destroying();
    ADOT4->Destroying();
    ADOT5->Destroying();
    RASH->Destroying();
    PRIH->Destroying();
    PERM->Destroying();
}
//+Метод класса+
boolean TForm1::TABL_CLASS::CONNECT(AnsiString Z)
{
    try
    {
        AnsiString S;
        S="Provider=MSDASQL.1;Persist Security Info=False;Data Source=";
        S=S+Z;
        ADOC=new TADOConnection(ADOC);
        ADOC->ConnectionString=S;
        ADOC->Provider="MSDASQL.1";
        ADOC->LoginPrompt=false;
        ADOC->Connected=true;
        if (ADOC->Connected)
        {
            return (true);
        }
        else
        {
            return (false);
        }
    }
    catch (...)
    {
        return (false);
    };
};
//+Метод класса+
boolean TForm1::TABL_CLASS::TABLE_ACTIV()
{
    try
    {
        //Таблица №1 - главная
        ADOT1=new TADOTable(ADOT1);
        ADOT1->TableName="STRTAB1";
        ADOT1->Connection=ADOC;
        ADOT1->Active=true;
        //Таблица №2
        ADOT2=new TADOTable(ADOT2);
        ADOT2->TableName="STRTAB2";
        ADOT2->Connection=ADOC;
        ADOT2->Active=true;
        //Таблица №3
        ADOT3=new TADOTable(ADOT3);
        ADOT3->TableName="STRTAB3";
        ADOT3->Connection=ADOC;
        ADOT3->Active=true;
        //Таблица №4
        ADOT4=new TADOTable(ADOT4);
        ADOT4->TableName="STRTAB4";
        ADOT4->Connection=ADOC;
        ADOT4->Active=true;
        //Таблица №5
    }
}

```

```

ADOT5=new TADOTable(ADOT5);
ADOT5->TableName="STRTAB5";
ADOT5->Connection=ADOC;
ADOT5->Active=true;
//////////////////////////////////Накладные//////////////////////////////////
RASH=new TADOTable(RASH);
RASH->TableName="NAKLADNARASH";
RASH->Connection=ADOC;
RASH->Active=true;
PRIH=new TADOTable(PRIH);
PRIH->TableName="NAKLADNAPRIH";
PRIH->Connection=ADOC;
PRIH->Active=true;
PERM=new TADOTable(PERM);
PERM->TableName="NAKLADNAPERM";
PERM->Connection=ADOC;
PERM->Active=true;
//////////////////////////////////
if (ADOT1->Active==true && ADOT2->Active==true && ADOT3->Active==true
    && ADOT4->Active==true && ADOT5->Active==true && RASH->Active==true &&
    PRIH->Active==true && PERM->Active==true)
{
    return (true);
}
else
{
    return (false);
}
}
catch (...)
{
    return (false);
};
};
//////////////////////////////////
//////////////////////////////////+Управление БД//////////////////////////////////
//////////////////////////////////
void __fastcall TForm1::BitBtn8Click(TObject *Sender)
{
    Form4->Edit5->Text="";
    Form4->Edit3->Text="";
    Form4->Edit4->Text="";
    Form4->Label1->Visible=false;
    Form4->Button4->Enabled=true;
    Form4->Button8->Enabled=true;
    Form4->Top=180;
    Form4->Left=319;
    Form4->ListBox1->Clear();
    Form4->ListBox2->Clear();
    Form4->Tag=0;
    Form4->ShowModal();
    switch (Form4->Tag)
    {
        case 1: TBL.PROC_KATEGOR_ADD(Form4->ListBox1,TBL.ADOT2); break;
        case 2: TBL.PROC_KATEGOR_ADD(Form4->ListBox2,TBL.ADOT4); break;
    }
}
//////////////////////////////////+Метод класса//////////////////////////////////
void TForm1::TABL_CLASS::PROC_KATEGOR_ADD(TListBox * LIST, TADOTable * ADOPROC)
{
    ADOPROC->Last(); //Переходим к последней записи
    for (int i=0;i<LIST->Count;i++)
    {
        Variant TRES;
        Variant KeyValues=LIST->Items->Strings[i];
        AnsiString KeyFields="NAME";
        AnsiString Result="NAME";
        TRES=ADOPROC->Lookup(KeyFields,KeyValues,Result);
        if (VarIsNull(TRES))
        {
            ADOPROC->Insert(); //Дополнили одной записью
            ADOPROC->FieldByName("NAME")->AsString=LIST->Items->Strings[i];
            ADOPROC->Post();
            ADOPROC->Last();
        }
        else
        {
            ShowMessage("Одно из вносимых данных уже есть в БД - они не будут заменены повторно!");
        }
    };
    ADOPROC->Active=false;
    ADOPROC->Active=true;
};
//////////////////////////////////
//////////////////////////////////+Приобретение товара//////////////////////////////////
//////////////////////////////////
void __fastcall TForm1::BitBtn5Click(TObject *Sender)
{
    Form2->Edit10->Text="";
    Form2->Edit2->Text="";
    Form2->Edit3->Text="";
    Form2->Edit4->Text="";
    Form2->Edit5->Text="";
    Form2->Edit6->Text="";
    Form2->Edit7->Text="";
    Form2->Edit9->Text="";
    Form2->Tag=0; //Не дописывать в файл
    Form2->Top=232;
    Form2->Left=339;
}

```

```

Form2->ComboBox1->Clear();
Form2->ComboBox2->Clear();
TBL.PROC_COMBO_READ (Form2->ComboBox1, TBL.ADOT2);
TBL.PROC_COMBO_READ (Form2->ComboBox2, TBL.ADOT4);
Form2->ShowModal();
if (Form2->Tag==1)
{
    if (Form2->ComboBox1->Text!=" " && Form2->ComboBox2->Text!=" ")
    {
        int CT2=TBL.FUNC_STRING_IN_INT (Form2->ComboBox1->Text, TBL.ADOT2);
        int CT3=TBL.FUNC_STRING_IN_INT (Form2->Edit4->Text, TBL.ADOT3);
        int CT4=TBL.FUNC_STRING_IN_INT (Form2->ComboBox2->Text, TBL.ADOT4);
        int CT5=TBL.FUNC_STRING_IN_INT (Form2->Edit10->Text, TBL.ADOT5);
        int ROZN=StrToInt (Form2->Edit3->Text);
        int ZAK=StrToInt (Form2->Edit2->Text);
        int NAKL=StrToInt (Form2->Edit6->Text);
        int NUM=StrToInt (Form2->Edit7->Text);
        AnsiString DATA=Form2->Edit5->Text;
        AnsiString DATA_SKLAD=Form2->Edit9->Text;
        if (TBL.WRITE_STRTAB1 (CT2, CT3, CT4, CT5, ROZN, ZAK, NAKL, NUM, DATA, DATA_SKLAD)==true)
        {
            TBL.NAKLPRIH (NUM, Form2->ComboBox1->Text, Form2->Edit10->Text,
                Form2->ComboBox2->Text, TBL.PROC_DOLLAR(), TBL.PROC_NDS(), ZAK);
        }
        else
        {
            ShowMessage ("При дополнении сведений в БД произошла ошибка - возможно серийный номер уже есть в БД!");
        };
    }
    else ShowMessage ("ВЫ НЕ ЗАПОЛНИЛИ ФАЙЛ КАТЕГОРИЙ ИЛИ СКЛАДОВ!");
};
}
}
///////////////////////////////////////////////////Для денег//////////////////////////////////////
float TForm1::TABL_CLASS::PROC_DOLLAR()
{
    return DOLLAR;
};
int TForm1::TABL_CLASS::PROC_NDS()
{
    return NDS;
};
boolean TForm1::TABL_CLASS::SETNDS DOLLAR (AnsiString DOL, AnsiString ND)
{
    try
    {
        StrToFloat (DOL);
        StrToInt (ND);
        if (StrToInt (ND)>=0 && StrToInt (ND)<101)
        {
            NDS=StrToInt (ND);
            DOLLAR=StrToFloat (DOL);
        }
        else
            return false;
    }
    catch (...)
    {return false;};
};
///////////////////////////////////////////////////Импортирование БД//////////////////////////////////////
void TForm1::TABL_CLASS::IMPORT (TEdit *Edi)
{
    AnsiString S;
    S="Provider=MSDASQL.1;Persist Security Info=False;Data Source=";
    S=S+Edi->Text;
    ADOCI=new TADOConnection (ADOI);
    ADOCI->ConnectionString=S;
    ADOCI->Provider="MSDASQL.1";
    ADOCI->LoginPrompt=false;
    ADOCI->Connected=true;
    if (ADOCI->Connected)
    {
        ADOI=new TADOTable (ADOI);
        ADOI->TableName="BAZA";
        ADOI->Connection=ADOI;
        ADOI->Active=true;
        ///////////////////////////////////Сам импорт//////////////////////////////////////
        ADOI->First ();
        while (ADOI->Eof==NULL)
        {
            int CT2=FUNC_STRING_IN_INT (ADOI->FieldByName ("Категория товара")->AsString, ADOT2);
            int CT3=FUNC_STRING_IN_INT (ADOI->FieldByName ("Фирма")->AsString, ADOT3);
            int CT4=FUNC_STRING_IN_INT (ADOI->FieldByName ("Склад")->AsString, ADOT4);
            int CT5=FUNC_STRING_IN_INT (ADOI->FieldByName ("Маркировка товара")->AsString, ADOT5);
            float ROZN=ADOI->FieldByName ("Розничная цена")->AsFloat;
            float ZAK=ADOI->FieldByName ("Закупочная цена")->AsFloat;
            int NAKL=ADOI->FieldByName ("Номер накладной")->AsInteger;
            int NUM=ADOI->FieldByName ("Серийный номер товара")->AsInteger;
            AnsiString DATA=DateToStr (ADOI->FieldByName ("Дата закупки")->AsDateTime);
            AnsiString DATA_SKLAD=DateToStr (ADOI->FieldByName ("Дата получения складом")->AsDateTime);
            if (WRITE_STRTAB1 (CT2, CT3, CT4, CT5, ROZN, ZAK, NAKL, NUM, DATA, DATA_SKLAD)==true)
            {
                NAKLPRIH (NUM, ADOI->FieldByName ("Категория товара")->AsString,
                    ADOI->FieldByName ("Маркировка товара")->AsString,
                    ADOI->FieldByName ("Склад")->AsString,
                    PROC_DOLLAR(), PROC_NDS(), ZAK);
            }
        }
        ADOI->Next ();
    }
}

```

```

    };
    ShowMessage("Импорт произведен удачно!");
}
else
{
    ShowMessage("Не удалось установить соединение!");
    return;
}
}

//////////+Метод класса+//////////
void TForm1::TABL_CLASS::NAKLPRIH(int NUM, AnsiString S1, AnsiString S2,
    AnsiString S3, float DOLLAR, int NDS, float ZAK)
{
    PRIH->Insert();
    PRIH->FieldByName("NUM")->AsInteger=NUM;
    float CENA=ZAK;
    float CENA_=CENA*(NDS*0.01)+1;
    PRIH->FieldByName("NDS")->AsFloat=CENA_;
    PRIH->FieldByName("ZAK")->AsFloat=CENA;
    PRIH->FieldByName("DOLLAR")->AsFloat=DOLLAR;
    PRIH->FieldByName("NAIMENOVANIE")->AsString=S1;
    PRIH->FieldByName("VID")->AsString=S2;
    PRIH->FieldByName("SKLAD_POLUCHATEL")->AsString=S3;
    PRIH->Post();
    PRIH->Active=false;
    PRIH->Active=true;
};

//////////+Метод класса+//////////
void TForm1::TABL_CLASS::PROC_COMBO_READ(TComboBox *CB, TADOTable *ADOTAB)
{
    ADOTAB->First();
    while (ADOTAB->Eof==NULL)
    {
        CB->Items->Add(ADOTAB->FieldByName("NAME")->AsString);
        ADOTAB->Next();
    };
};

//////////+Метод класса+//////////
boolean TForm1::TABL_CLASS::WRITE_STRTAB1(int CT2, int CT3, int CT4, int CT5,
    int ROZN, int ZAK, int NAKL, int NUM,
    AnsiString DATA, AnsiString DATA_SKLAD)
{
    Variant TRES;
    Variant KeyValues=NUM;
    AnsiString KeyFields="NUM";
    AnsiString Result="NUM";
    TRES=ADOT1->Lookup(KeyFields,KeyValues,Result);
    ////////////Проверяем результат поиска//////////
    if (VarIsNull(TRES))
    {
        ////////////Такой записи нет - добавляем в файл//////////
        ADOT1->Last();
        ADOT1->Insert(); //Дополнили одной записью
        ADOT1->FieldByName("COD_TOVAR2")->AsInteger=CT2;
        ADOT1->FieldByName("COD_TOVAR3")->AsInteger=CT3;
        ADOT1->FieldByName("COD_TOVAR4")->AsInteger=CT4;
        ADOT1->FieldByName("COD_TOVAR5")->AsInteger=CT5;
        ADOT1->FieldByName("ROZN")->AsInteger=ROZN;
        ADOT1->FieldByName("ZAK")->AsInteger=ZAK;
        ADOT1->FieldByName("NAKL")->AsInteger=NAKL;
        ADOT1->FieldByName("NUM")->AsInteger=NUM;
        ADOT1->FieldByName("DATA")->AsString=DATA;
        ADOT1->FieldByName("DATA_SKLAD")->AsString=DATA_SKLAD;
        ADOT1->Post();
        ////////////Обновляем данные//////////
        ADOT1->Active=false;
        ADOT1->Active=true;
        return (true);
    }
    else
    {
        return (false);
    };
};

//////////+Метод класса+//////////
int TForm1::TABL_CLASS::FUNC_STRING_IN_INT(AnsiString S, TADOTable * ADO)
{
    Variant TRES;
    Variant KeyValues=S;
    AnsiString KeyFields="NAME";
    AnsiString Result="COD";
    ////////////Ищем максимум COD//////////
    int k=0;
    ADO->First();
    while (ADO->Eof==NULL)
    {
        if (ADO->FieldByName("COD")->AsInteger>=k)
            k=ADO->FieldByName("COD")->AsInteger;
        ADO->Next();
    };
    ////////////
    TRES=ADO->Lookup(KeyFields,KeyValues,Result);
    ////////////Проверяем результат поиска//////////
    if (VarIsNull(TRES))
    {
        ////////////Такой записи нет - добавляем в файл//////////
        ADO->Insert(); //Дополнили одной записью
        ADO->FieldByName("NAME")->AsString=S;
        ADO->Post(); //Закрепляем

```

```

ADO->Active=false;
ADO->Active=true;
return (k+1);
}
else
{
return (TRES);
};
};
//////////+Метод класса+//////////
AnsiString TForm1::TABL_CLASS::FUNC_INT_IN_STRING(int i, TADOTable * ADO)
{
Variant TRES;
Variant KeyValues=i;
AnsiString KeyFields="COD";
AnsiString Result="NAME";
TRES=ADO->Lookup(KeyFields,KeyValues,Result);
return (TRES);
};
//////////+Метод класса+//////////
void TForm1::TABL_CLASS::PROC_DEREVO_DELETE()
{
//////////Удаляем подподрезья//////////
PK=PKFIRST;
BUFER1=NULL;
while (PK!=NULL)
{
PKO=PK->NEXTKAT;
while (PKO!=NULL)
{
//////////
PM=PKO->NEXTMARK;
while (PM!=NULL)
{
BUFER1=PM->NEXT;
delete PM;
PM=BUFER1;
};
PKO->NEXTMARK=NULL;
//////////
PKO=PKO->NEXT;
};
PK=PK->NEXTKOR;
};
//////////Удаляем подрезья//////////
PK=PKFIRST;
BUFER2=NULL;
while (PK!=NULL)
{
//////////
PKO=PK->NEXTKAT;
while (PKO!=NULL)
{
BUFER2=PKO->NEXT;
delete PKO;
PKO=BUFER2;
};
PK->NEXTKAT=NULL;
//////////
PK=PK->NEXTKOR;
};
//////////Удаляем корни//////////
PK=PKFIRST;
BUFER3=NULL;
while (PK!=NULL)
{
BUFER3=PK->NEXTKOR;
delete PK;
PK=BUFER3;
};
delete PK;
delete PKFIRST;
};
//////////+Метод класса+//////////
void TForm1::TABL_CLASS::PROC_DEREVO(TADOTable * ADOTAB, TADOTable * ADOTAB1)
{
//////////Строим корни//////////
ADOTAB->First();
while (ADOTAB->Eof==NULL)
{
if (PK==NULL)
{
PK=new (KOREN);
PKFIRST=PK;
PK->k=ADOTAB->FieldByName("COD")->AsInteger;
PK->NEXTKOR=NULL;
PK->NEXTKAT=NULL;
ADOTAB->Next();
}
else
{
PK->NEXTKOR = new (KOREN);
PK=PK->NEXTKOR;
PK->k=ADOTAB->FieldByName("COD")->AsInteger;
PK->NEXTKOR=NULL;
PK->NEXTKAT=NULL;
ADOTAB->Next();
}
}
}

```

```

    };
    ///////////////////////////////////////////////////Строим листья - категории//////////////////////////////////////
    ADOTAB1->First();
    while (ADOTAB1->Eof==NULL)
    {
        int SKL=ADOTAB1->FieldByName("COD_TOVAR4")->AsInteger; //SKL = номер склада в таблице №1
        PK=PKFIRST;
        //Идем по списку корней
        while (PK!=NULL)
        {
            if (SKL==PK->k)
            {
                //////////////////////////////////Нашли ветвь с нужным складом//////////////////////////////////////
                if (PK->NEXTKAT==NULL)
                {
                    //////////////////////////////////Если в данном складе нет ни одной категории - то создаем//////////////////////////////////////
                    PK->NEXTKAT=new (KATEGOR);
                    PKO=PK->NEXTKAT;
                    PKO->k=ADOTAB1->FieldByName("COD_TOVAR2")->AsInteger;
                    PKO->NEXT=NULL;
                    PKO->NEXTMARK=new (MARK);
                    PM=PKO->NEXTMARK;
                    PM->m=ADOTAB1->FieldByName("COD_TOVAR5")->AsInteger;
                    PM->NEXT=NULL;
                }
            }
            else
            {
                //////////////////////////////////Категория уже есть//////////////////////////////////////
                boolean flag=false;
                PKO=PK->NEXTKAT;
                KATEGOR * buf; //Предыдущий
                while (PKO!=NULL)
                {
                    if (PKO->k==ADOTAB1->FieldByName("COD_TOVAR2")->AsInteger)
                    {
                        flag=true;
                        //////////////////////////////////Категория есть - а марка?//////////////////////////////////////
                        //////////////////////////////////Здесь уже есть хотя бы одна марка//////////////////////////////////////
                        boolean flag_=false;
                        PM=PKO->NEXTMARK;
                        MARK * buf_;
                        while (PM!=NULL)
                        {
                            if (PM->m==ADOTAB1->FieldByName("COD_TOVAR5")->AsInteger)
                            {
                                //////////////////////////////////Данная марка уже есть//////////////////////////////////////
                                flag_=true;
                            }
                            buf_=PM;
                            PM=PM->NEXT;
                        }
                    }
                    if (flag==false)
                    {
                        //////////////////////////////////Данной марки еще нет//////////////////////////////////////
                        PM=buf_;
                        PM->NEXT=new (MARK);
                        PM=PM->NEXT;
                        PM->m=ADOTAB1->FieldByName("COD_TOVAR5")->AsInteger;
                        PM->NEXT=NULL;
                    }
                    //////////////////////////////////END MARK//////////////////////////////////////
                }
            }
            buf=PKO;
            PKO=PKO->NEXT;
        }
        if (flag==false)
        {
            //////////////////////////////////Не нашли в списке категорию//////////////////////////////////////
            PKO=buf;
            PKO->NEXT=new (KATEGOR);
            PKO=PKO->NEXT;
            PKO->k=ADOTAB1->FieldByName("COD_TOVAR2")->AsInteger;
            PKO->NEXT=NULL;
            //////////////////////////////////Т.К. категории еще нет, то и марка будет первой//////////////////////////////////////
            PKO->NEXTMARK=new (MARK);
            PM=PKO->NEXTMARK;
            PM->m=ADOTAB1->FieldByName("COD_TOVAR5")->AsInteger;
            PM->NEXT=NULL;
        }
    };
    PK=PK->NEXTKOR;
    ADOTAB1->Next();
};

/////////////////////////////////////////////////Просмотр БД//////////////////////////////////////
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    CLEARWINDOW(); //Перед следующим просмотром необходимо все окна удалить
    TBL.SETREENODE(); //Обнуляем указатель на NODE
    TBL.DELETEINDEX(); //Удаляем список чисел для выделения номера склада
    TBL.PROC_DEREVO_DELETE(); //Разрушаем дерево, по которому выводили на экран
    //////////////////////////////////Просмотр заново//////////////////////////////////////
    TBL.PROC_DEREVO(TBL.ADOT4, TBL.ADOT1);
    TBL.TREE_VIEW(Form1->TreeView1);
};

/////////////////////////////////////////////////Метод класса//////////////////////////////////////
void TForm1::TABL_CLASS::TREE_VIEW(TTreeView * TV)
{

```



```

ADDINDEX(0);
////////////////////////////////////
TTreeNode * Node1;
int SKL=0; //Счетчик складов
int KAT=0;
int MAR=0;
int index=0;
int BUF_KAT=0;
TV->Items->Clear();
PK=PKFIRST;
while (PK!=NULL)
{
    AnsiString SS=FUNC_INT_IN_STRING(PK->k,ADOT4);
    TV->Items->Add(NULL,SS);
    ////////////////////////////////////Добавляем поддеревья////////////////////////////////////
    PKO=PK->NEXTKAT;
    while (PKO!=NULL)
    {
        AnsiString SS=FUNC_INT_IN_STRING(PKO->k,ADOT2);
        Node1 = TV->Items->Item[BUF_KAT];
        TV->Items->AddChild(Node1,SS);
        ////////////////////////////////////Добавляем подподдеревья////////////////////////////////////
        PM=PKO->NEXTMARK;
        int PAR=MAR;
        while (PM!=NULL)
        {
            AnsiString SS=FUNC_INT_IN_STRING(PM->m,ADOT5);
            Node1 = TV->Items->Item[KAT+PAR+SKL+1];
            TV->Items->AddChild(Node1,SS);
            PM=PM->NEXT;
            MAR++;
        }
        ////////////////////////////////////
        PKO=PKO->NEXT;
        KAT++;
    };
    ////////////////////////////////////
    PK=PK->NEXTKOR;
    SKL++;
    ////////////////////////////////////
    BUF_KAT=SKL+KAT+MAR;
    ////////////////////////////////////Нужно для открытия в отдельном окне////////////////////////////////////
    ADDINDEX(BUF_KAT);
};
};
////////////////////////////////////+Метод класса////////////////////////////////////
void TForm1::TABL_CLASS::ADDINDEX(int j)
{
    if (PIFIRST==NULL)
    {
        PIFIRST=new (INDEXTREE);
        PI=PIFIRST;
        PI->m=j;
        PI->NEXT=NULL;
    }
    else
    {
        PI->NEXT=new (INDEXTREE);
        PI=PI->NEXT;
        PI->m=j;
        PI->NEXT=NULL;
    };
};
////////////////////////////////////+Метод класса////////////////////////////////////
void TForm1::TABL_CLASS::DELETEINDEX()
{
    PI=PIFIRST;
    while (PI!=NULL)
    {
        BUF=PI->NEXT;
        delete PI;
        PI=BUF;
    };
    PI=NULL;
    BUF=NULL;
    PIFIRST=NULL;
};
////////////////////////////////////+Метод класса////////////////////////////////////
AnsiString TForm1::TABL_CLASS::GETSKLAD(int j)
{
    int COUNT=1; //У нас же в файле нумерация 1,2,3,..
    PI=PIFIRST;
    while (PI!=NULL)
    {
        BUF=PI->NEXT;
        if (j>PI->m && j<BUF->m)
        {
            return (FUNC_INT_IN_STRING(COUNT, ADOT4));
        };
        PI=PI->NEXT;
        COUNT++;
    };
};
////////////////////////////////////
////////////////////////////////////+Просмотр в отдельном окне////////////////////////////////////
////////////////////////////////////
////////////////////////////////////+Главный метод, создающий окна////////////////////////////////////

```

```

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    if (Form1->TBL.GETTREENODE() != NULL)
    {
        //////////////////////////////////////////////////
        FORMES= new TForm(Form1);
        FORMES->FormStyle=fsMDIChild;
        FORMES->BorderStyle=bsSizeToolWin;
        if (FIRST==NULL)
        {
            FIRST = new (OKN_SP);
            worker = FIRST;
            worker->HAND = FORMES->Handle;
            worker->NEXT=NULL;
            //////////////////////////////////////////////////Создаем дерево просмотра////////////////////////////////////
            TTreeView * TV = new TTreeView(Form1);
            TV->Parent=FORMES;
            TV->Align=alClient;
            TV->Color=cl3DLight;
            TBL.SET_TREE_WINDWOW(TV, TBL.GETTREENODE());
            FORMES->Caption=TBL.GETSKLAD(TBL.GETABSOLUTINDEX(TBL.GETTREENODE()));
        }
    }
    else
    {
        worker->NEXT = new (OKN_SP);
        worker=worker->NEXT;
        worker->HAND= FORMES->Handle;
        worker->NEXT=NULL;
        //////////////////////////////////////////////////Создаем дерево просмотра////////////////////////////////////
        TTreeView * TV = new TTreeView(Form1);
        TV->Parent=FORMES;
        TV->Align=alClient;
        TV->Color=cl3DLight;
        TBL.SET_TREE_WINDWOW(TV, TBL.GETTREENODE());
        FORMES->Caption=TBL.GETSKLAD(TBL.GETABSOLUTINDEX(TBL.GETTREENODE()));
    };
    //////////////////////////////////////////////////
}
}
//////////////////////////////////////////////////+Метод класса+////////////////////////////////////
void TForm1::TABL_CLASS::SETINDEXTREE(TTreeView *TV)
{
    TN1=TV->Selected;
}
//////////////////////////////////////////////////+Метод класса+////////////////////////////////////
int TForm1::TABL_CLASS::GETABSOLUTINDEX(TTreeNode * TNODE)
{
    return TNODE->AbsoluteIndex;
}
//////////////////////////////////////////////////+Метод класса+////////////////////////////////////
TTreeNode * TForm1::TABL_CLASS::GETTREENODE()
{
    return TN1;
}
//////////////////////////////////////////////////+Метод класса+////////////////////////////////////
void TForm1::TABL_CLASS::SETTREENODE()
{
    TN1=NULL;
}
//////////////////////////////////////////////////+Метод класса+////////////////////////////////////
int TForm1::TABL_CLASS::SET_TREE_WINDWOW(TTreeView *TV, TTreeNode * TN1)
{
    TTreeNode * Nodel;
    int SKL=0; //Счетчик складов
    int KAT=0;
    int MAR=0;
    int index=0;
    int BUF_KAT=0;
    TV->Items->Clear();
    PK=PKFIRST;
    while (PK!=NULL)
    {
        if (FUNC_STRING_IN_INT(GETSKLAD(TN1->AbsoluteIndex), ADOT4)==PK->k)
        {
            AnsiString SS=FUNC_INT_IN_STRING(PK->k, ADOT4);
            TV->Items->Add(NULL, SS);
            //////////////////////////////////////////////////Добавляем поддерева////////////////////////////////////
            PKO=PK->NEXTKAT;
            while (PKO!=NULL)
            {
                AnsiString SS=FUNC_INT_IN_STRING(PKO->k, ADOT2);
                Nodel = TV->Items->Item[BUF_KAT];
                TV->Items->AddChild(Nodel, SS);
                //////////////////////////////////////////////////Добавляем подподдерева////////////////////////////////////
                PM=PKO->NEXTMARK;
                int PAR=MAR;
                while (PM!=NULL)
                {
                    AnsiString SS=FUNC_INT_IN_STRING(PM->m, ADOT5);
                    Nodel = TV->Items->Item[KAT+PAR+SKL+1];
                    TV->Items->AddChild(Nodel, SS);
                    PM=PM->NEXT;
                    MAR++;
                };
            };
            PKO=PKO->NEXT;
            KAT++;
        }
    }
}

```

```

    };
    //////////////////////////////////////
    SKL++;
    //////////////////////////////////
    BUF_KAT=SKL*KAT+MAR;
    }
    PK=PK->NEXTKOR;
    };
};
////////////////////////////////////+Расположение окон+////////////////////////////////////
void __fastcall TForm1::Button6Click(TObject *Sender)
{
    Form1->Cascade();
}
void __fastcall TForm1::Button9Click(TObject *Sender)
{
    Form1->Tile();
}
////////////////////////////////////+Переход к следующему окну+////////////////////////////////////
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Form1->Next();
}
////////////////////////////////////+Переход к предыдущему окну+////////////////////////////////////
void __fastcall TForm1::Button4Click(TObject *Sender)
{
    Form1->Previous();
}
////////////////////////////////////+Закрытие окна+////////////////////////////////////
void __fastcall TForm1::Button5Click(TObject *Sender)
{
    //////////////////////////////////Здесь Form1->MDIChildren[0] - MDI окно, имеющее фокус ввода////////////////////////////////
    if (FIRST!=NULL) //Если есть список
    {
        if (FIRST->NEXT!=NULL) //Если число элементов списка >1
        {
            worker=FIRST;
            buf=NULL;
            boolean flag=false;
            //////////////////////////////////Ищем нужный элемент списка////////////////////////////////
            while (worker!=NULL && flag!=true)
            {
                if (Form1->MDIChildren[0]->Handle == worker->HAND)
                {
                    flag=true;
                }
                else
                {
                    buf = worker;
                    worker=worker->NEXT;
                }
            };
            //////////////////////////////////worker = ОКНО, которое пользователь хочет закрыть////////////////////////////////
            if (flag==true)
            {
                if (buf != NULL) //Если удаляемый элемент не первый
                {
                    buf->NEXT=worker->NEXT; //Переставляем указатели
                    delete worker; //Освобождаем память из под узла
                }
                else //Если удаляемый элемент первый
                {
                    FIRST=worker->NEXT;
                    delete worker;
                }
            };
            delete(Form1->MDIChildren[0]);
        };
    };
    //////////////////////////////////
}
else //Если в списке всего один элемент
{
    worker=FIRST;
    delete worker;
    delete(Form1->MDIChildren[0]);
    FIRST=NULL;
};
};
}
////////////////////////////////////Событие, возникающее при выходе из программы////////////////////////////////////
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
    CLEARWINDOW();
}
////////////////////////////////////Процедура закрытия всех окон////////////////////////////////////
void TForm1::CLEARWINDOW()
{
    worker=FIRST;
    buf=NULL;
    while (worker!=NULL)
    {
        buf = worker->NEXT;
        delete worker;
        delete(Form1->MDIChildren[0]);
        worker=buf;
    };
    FIRST=NULL;
    buf=NULL;
}
}
////////////////////////////////////Событие, возникающее при выборе склада для отображения его в отдельном окне//

```

```

void __fastcall TForm1::TREEMOUSE(TObject *Sender, TMouseButton Button,
TShiftState Shift, int X, int Y)
{
    TBL.SETINDEXTREE(Form1->TreeView1);
}
void __fastcall TForm1::TREEKEY(TObject *Sender, WORD &Key,
TShiftState Shift)
{
    TBL.SETINDEXTREE(Form1->TreeView1);
}
////////////////////////////////////
////////////////////////////////////+Поиск товара по серийному номеру+////////////////////////////////////
////////////////////////////////////
void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
    Form6->Edit1->Text="";
    Form6->Edit2->Text="";
    Form6->Edit3->Text="";
    Form6->Edit4->Text="";
    Form6->Edit5->Text="";
    Form6->Edit6->Text="";
    Form6->Edit7->Text="";
    Form6->Edit8->Text="";
    Form6->Edit9->Text="";
    Form6->Edit10->Text="";
    Form6->Left=290;
    Form6->Top=241;
    Form6->ShowModal();
}
////////////////////////////////////+Метод класса+////////////////////////////////////
void TForm1::TABL_CLASS::FIND(TEdit *S1,TEdit *S2,TEdit *S3,TEdit *S4,
TEdit *S5,TEdit *S6,TEdit *S7,TEdit *S8,
TEdit *S9,TEdit *S10)
{
    ADOT1->First();
    while (ADOT1->Eof==NULL)
    {
        //////////////////////////////////////Идем по таблице №1////////////////////////////////////
        if (ADOT1->FieldByName("NUM")->AsInteger==StrToInt(S1->Text))
        {
            //////////////////////////////////////Нашли запись////////////////////////////////////
            S2->Text=FUNC_INT_IN_STRING(ADOT1->FieldByName("COD_TOVAR2")->AsInteger,ADOT2);
            S3->Text=FUNC_INT_IN_STRING(ADOT1->FieldByName("COD_TOVAR5")->AsInteger,ADOT5);
            S4->Text=FUNC_INT_IN_STRING(ADOT1->FieldByName("COD_TOVAR3")->AsInteger,ADOT3);
            S5->Text=FUNC_INT_IN_STRING(ADOT1->FieldByName("COD_TOVAR4")->AsInteger,ADOT4);
            S6->Text=ADOT1->FieldByName("NAKL")->AsInteger;
            S7->Text=ADOT1->FieldByName("DATA")->AsString;
            S8->Text=ADOT1->FieldByName("DATA_SKLAD")->AsString;
            S9->Text=ADOT1->FieldByName("ZAK")->AsInteger;
            S10->Text=ADOT1->FieldByName("ROZN")->AsInteger;
        };
        ADOT1->Next();
    }
};
////////////////////////////////////
////////////////////////////////////+HELP+////////////////////////////////////
////////////////////////////////////
void __fastcall TForm1::BitBtn9Click(TObject *Sender)
{
    Form7->Left=247;
    Form7->Top=141;
    Form7->ShowModal();
}
////////////////////////////////////
////////////////////////////////////+Перемещение товара+////////////////////////////////////
////////////////////////////////////
void __fastcall TForm1::BitBtn7Click(TObject *Sender)
{
    Form8->Edit1->Text="";
    Form8->Edit2->Text="";
    Form8->Edit3->Text="";
    Form8->Left=294;
    Form8->Top=268;
    Form8->ComboBox1->Clear();
    TBL.PROC_COMBO_READ(Form8->ComboBox1,TBL.ADOT4);
    Form8->ShowModal();
    try
    {
        TDateTime dtDate = StrToDate(Form8->Edit3->Text);
        if ((StrToInt(Form8->Edit1->Text)) && (Form8->Edit1->Text!="") &&
            (StrToInt(Form8->Edit2->Text)) && Form8->ComboBox1->Text!="")
        {
            //////////////////////////////////////Пользователь ввел верные данные////////////////////////////////////
            TBL.NAKLPERM(StrToInt(Form8->Edit1->Text), Form8->ComboBox1->Text); //Мы должны учесть старый и новый склад
            TBL.CHANGESKLAD(Form8->Edit1,Form8->Edit2,Form8->Edit3,Form8->ComboBox1);
        }
    }
    catch (...)
    {
        ShowMessage("Вы некоректно ввели значения!");
    }
}
////////////////////////////////////+Метод класса+////////////////////////////////////
void TForm1::TABL_CLASS::NAKLPERM(int NUM, AnsiString S1)
{
    ADOT1->First();
    while (ADOT1->Eof==NULL)
    {
        //////////////////////////////////////Поиск по серийному номеру////////////////////////////////////
        if (ADOT1->FieldByName("NUM")->AsInteger==NUM)
        {
            PERM->Insert();
            PERM->FieldByName("NUM")->AsInteger=NUM;

```

```

    PERM->FieldByName("NAIMENOVANIE")->AsString=
    FUNC_INT_IN_STRING(ADOT1->FieldByName("COD_TOVAR2")->AsInteger,ADOT2);
    PERM->FieldByName("VID")->AsString=
    FUNC_INT_IN_STRING(ADOT1->FieldByName("COD_TOVAR5")->AsInteger,ADOT5);
    PERM->FieldByName("SKLAD_POL")->AsString=S1;
    PERM->FieldByName("SKLAD_OTPR")->AsString=
    FUNC_INT_IN_STRING(ADOT1->FieldByName("COD_TOVAR4")->AsInteger,ADOT4);
    PERM->Post();
    PERM->Active=false;
    PERM->Active=true;
    return; //Исправлено 23.12.2005
};
ADOT1->Next();
};
};
//+Метод класса+
void TForm1::TABL_CLASS::CHANGESKLAD(TEdit *S1,TEdit *S2,TEdit *S3,TComboBox *CB)
{
    ADOT1->First();
    while (ADOT1->Eof==NULL)
    {
        //Идем по таблице №1
        if (ADOT1->FieldByName("NUM")->AsInteger==StrToInt(S1->Text))
        {
            //Нашли запись
            ADOT1->Edit();
            ADOT1->FieldByName("COD_TOVAR4")->AsInteger=FUNC_STRING_IN_INT(CB->Text, ADOT4);
            ADOT1->FieldByName("NAKL")->AsInteger=StrToInt(S2->Text);
            ADOT1->FieldByName("DATA_SKLAD")->AsString=S3->Text;
            ADOT1->Post();
            ADOT1->Active=false;
            ADOT1->Active=true;
            return; //Исправлено 23.12.2005
        }
    };
    ADOT1->Next();
}
}
//+Продажа товара+
void __fastcall TForm1::BitBtn6Click(TObject *Sender)
{
    Form9->Edit1->Text="";
    Form9->Left=327;
    Form9->Top=326;
    Form9->ShowModal();
    try
    {
        StrToInt(Form9->Edit1->Text);
        TBL.NAKLRASH(StrToInt(Form9->Edit1->Text),TBL.PROC_DOLLAR(),TBL.PROC_NDS()); //Перед тем как удалять - создадим накладную
        TBL.DELETETOVAR(Form9->Edit1);
    }
    catch (...)
    {
        ShowMessage("При работе возникла ошибка!");
    }
}
//+Метод класса+
void TForm1::TABL_CLASS::NAKLASH(int NUM, float DOLLAR, int NDS)
{
    boolean FLAG=false;
    ADOT1->First();
    while (ADOT1->Eof==NULL)
    {
        //Поиск по серийному номеру
        if (ADOT1->FieldByName("NUM")->AsInteger==NUM)
        {
            RASH->Insert();
            RASH->FieldByName("NUM")->AsInteger=NUM;
            float CENA=ADOT1->FieldByName("ROZN")->AsFloat;
            float CENA_=CENA*((NDS*0.01)+1);
            RASH->FieldByName("NDS")->AsFloat=CENA_;
            RASH->FieldByName("ROZN")->AsFloat=CENA;
            RASH->FieldByName("DOLLAR")->AsFloat=DOLLAR;
            RASH->FieldByName("NAIMENOVANIE")->AsString=
            FUNC_INT_IN_STRING(ADOT1->FieldByName("COD_TOVAR2")->AsInteger,ADOT2);
            RASH->FieldByName("VID")->AsString=
            FUNC_INT_IN_STRING(ADOT1->FieldByName("COD_TOVAR5")->AsInteger,ADOT5);
            RASH->FieldByName("SKLAD_OTPRVITEL")->AsString=
            FUNC_INT_IN_STRING(ADOT1->FieldByName("COD_TOVAR4")->AsInteger,ADOT4);
            RASH->Post();
            RASH->Active=false;
            RASH->Active=true;
            FLAG=true;
            return; //Исправлено 23.12.2005
        }
    };
    ADOT1->Next();
};
if (FLAG==false)
{
    ShowMessage("Товара с данным серийным номером нет в ВД!");
};
//+Метод класса+
void TForm1::TABL_CLASS::DELETETOVAR(TEdit * ED1)
{
    ADOT1->First();
    while (ADOT1->Eof==NULL)
    {
        //Идем по таблице №1
        if (ADOT1->FieldByName("NUM")->AsInteger==StrToInt(ED1->Text))
        {
            ADOT1->Delete();
            ADOT1->Active=false;
            ADOT1->Active=true;
        }
    }
}

```

```

        return;
    };
    ADOT1->Next ();
}

}
////////////////////////////////////
////////////////////////////////////+Управление отчетами+////////////////////////////////////
////////////////////////////////////
void __fastcall TForm1::BitBtn3Click(TObject *Sender)
{
    Form3->Edit1->Text="";
    Form3->ComboBox5->Text="";
    Form3->Edit3->Text="";
    Form3->ComboBox6->Text="";
    Form3->Edit4->Text="";
    Form3->Edit2->Text="";
    Form3->RadioButton1->Checked=true;
    Form3->RadioButton4->Checked=true;
    Form3->RadioButton6->Checked=true;
    Form3->Label13->Visible=true;
    Form3->Edit1->Visible=true;
    Form3->RadioButton8->Checked=true;
    Form3->Label15->Visible=true;
    Form3->Edit3->Visible=true;
    Form3->Edit4->Visible=true;
    Form3->Top=143;
    Form3->Left=211;
    Form3->ComboBox5->Clear ();
    Form3->ComboBox6->Clear ();
    TBL.PROC_COMBO_READ (Form3->ComboBox5, TBL.ADOT2);
    TBL.PROC_COMBO_READ (Form3->ComboBox6, TBL.ADOT2);
    Form3->Show ();
}

////////////////////////////////////+Метод класса+////////////////////////////////////
void TForm1::TABL_CLASS::FILESS(FILE * files, AnsiString BD, AnsiString BUFER)
{
    char HTML[1000];
    AnsiString BUF;
    BUF="<h3>";
    BUF=BUF+BUFER;
    BUF=BUF+BD;
    StrCopy (HTML, BUF.c_str());
    int LEN=strlen(HTML);
    fseek(files, 0, SEEK_END);
    fwrite(&HTML, LEN, 1, files);
    //////////////////////////////////
    BUF="</h3>";
    StrCopy (HTML, BUF.c_str());
    LEN=strlen(HTML);
    fseek(files, 0, SEEK_END);
    fwrite(&HTML, LEN, 1, files);
};

////////////////////////////////////+Метод класса+////////////////////////////////////
void TForm1::TABL_CLASS::LINIA(FILE * files)
{
    char HTML[1000];
    AnsiString BUF="<hr size=4>";
    StrCopy (HTML, BUF.c_str());
    int LEN=strlen(HTML);
    fseek(files, 0, SEEK_END);
    fwrite(&HTML, LEN, 1, files);
}

////////////////////////////////////+Метод класса+////////////////////////////////////
void TForm1::TABL_CLASS::MESSAGES(FILE * files, AnsiString MES)
{
    char HTML[1000];
    AnsiString BUF;
    BUF="<h3>";
    BUF=BUF+MES;
    StrCopy (HTML, BUF.c_str());
    int LEN=strlen(HTML);
    fseek(files, 0, SEEK_END);
    fwrite(&HTML, LEN, 1, files);
    //////////////////////////////////
    BUF="</h3>";
    StrCopy (HTML, BUF.c_str());
    LEN=strlen(HTML);
    fseek(files, 0, SEEK_END);
    fwrite(&HTML, LEN, 1, files);
}

////////////////////////////////////+Метод класса+////////////////////////////////////
void TForm1::TABL_CLASS::OTCHET_ZAGOLOVOK(FILE * files, AnsiString S)
{
    char HTML[1000];
    StrCopy (HTML, "<html>");
    int LEN=strlen(HTML);
    fseek(files, 0, SEEK_END);
    fwrite(&HTML, LEN, 1, files);
    //////////////////////////////////
    StrCopy (HTML, "<body bgcolor=20B2AA>");
    LEN=strlen(HTML);
    fseek(files, 0, SEEK_END);
    fwrite(&HTML, LEN, 1, files);
    //////////////////////////////////
    AnsiString SS="<h1 align=center>";
    SS=SS+S;
    SS=SS+"</h1>";
}

```

```

StrCopy(HTML,SS.c_str());
LEN=strlen(HTML);
fseek(files, 0, SEEK_END);
fwrite(&HTML, LEN, 1, files);
}
//////////+Метод класса+//////////
void TForm1::TABL_CLASS::OTCHET_PRODAGA(AnsiString S, AnsiString KAT)
{
    FILE * files;
    AnsiString F=S;
    F=F+".html";
    if ((files=fopen(F.c_str(), "w"))==NULL)
    {
        ShowMessage("Невозможно создать файл!");
        return;
    };
    //Создаем файл с отчетом
    OTCHET_ZAGOLOVOK(files,"Отчет по продаже товаров по категории и виду");
    RASH->First();
    while (RASH->Eof==NULL)
    {
        if (RASH->FieldByName("NAIMENOVANIE")->AsString==KAT)
        {
            FILESS(files,RASH->FieldByName("NUM")->AsString,"Серийный номер товара: ");
            FILESS(files,RASH->FieldByName("NAIMENOVANIE")->AsString,"Категория товара: ");
            FILESS(files,RASH->FieldByName("VID")->AsString,"Вид товара: ");
            FILESS(files,RASH->FieldByName("ROZN")->AsString,"Розничная цена товара: ");
            FILESS(files,RASH->FieldByName("NDS")->AsString,"Розничная цена товара с учетом НДС: ");
            FILESS(files,RASH->FieldByName("DOLLAR")->AsString,"Курс $: ");
            FILESS(files,RASH->FieldByName("SKLAD_OTPRAVITEL")->AsString,"Склад отправитель: ");
            LINIA(files);
        }
        RASH->Next();
    };
    fclose(files);
    ShellExecute(NULL,"open",F.c_str(),NULL,NULL,SW_MAXIMIZE); //Открываем файл с отчетом
}
//////////+Метод класса+//////////
void TForm1::TABL_CLASS::OTCHET_PRODAGA_ONE(AnsiString S, AnsiString S1)
{
    FILE * files;AnsiString F=S;F=F+".html";
    if ((files = fopen(F.c_str(), "w"))==NULL)
    {
        ShowMessage("Невозможно создать файл!");
        return;
    };
    OTCHET_ZAGOLOVOK(files,"Отчет по продаже товара по серийному номеру");
    RASH->First();
    while (RASH->Eof==NULL)
    {
        if (RASH->FieldByName("NUM")->AsString==StrToInt(S1))
        {
            FILESS(files,RASH->FieldByName("NUM")->AsString,"Серийный номер товара: ");
            FILESS(files,RASH->FieldByName("NAIMENOVANIE")->AsString,"Категория товара: ");
            FILESS(files,RASH->FieldByName("VID")->AsString,"Вид товара: ");
            FILESS(files,RASH->FieldByName("ROZN")->AsString,"Розничная цена товара: ");
            FILESS(files,RASH->FieldByName("NDS")->AsString,"Розничная цена товара с учетом НДС: ");
            FILESS(files,RASH->FieldByName("DOLLAR")->AsString,"Курс $: ");
            FILESS(files,RASH->FieldByName("SKLAD_OTPRAVITEL")->AsString,"Склад отправитель: ");
            LINIA(files);
        }
        RASH->Next();
    };
    fclose(files);
    ShellExecute(NULL,"open",F.c_str(),NULL,NULL,SW_MAXIMIZE); //Открываем файл с отчетом
}
//////////+Метод класса+//////////
void TForm1::TABL_CLASS::OTCHET_KATEGOR(AnsiString S, AnsiString KAT)
{
    FILE * files;AnsiString F=S;F=F+".html";
    if ((files = fopen(F.c_str(), "w"))==NULL)
    {
        ShowMessage("Невозможно создать файл!");
        return;
    };
    OTCHET_ZAGOLOVOK(files,"Отчет по продаже товаров по категории");
    FILESS(files,KAT,"Категория товара: ");
    RASH->First();
    int j=0;
    while (RASH->Eof==NULL)
    {
        if (RASH->FieldByName("NAIMENOVANIE")->AsString==KAT)
        {
            FILESS(files,RASH->FieldByName("VID")->AsString,"Вид товара: ");
            j++;
        }
        RASH->Next();
    };
    //Количество проданного товара
    AnsiString I;

```

```

I="Было продано всего: ";
I=I+VarToStr(j);
I=I+" шт.";
MESSAGES(files,I);
////////////////////////////////////
fclose(files);
ShellExecute(NULL,"open",F.c_str(),NULL,NULL,SW_MAXIMIZE); //Открываем файл с отчетом
}
////////////////////////////////////+Метод класса////////////////////////////////////
void TForm1::TABL_CLASS::OTCHET_PERM(AnsiString S, int NUM)
{
    FILE * files;AnsiString F=S;F=F+".html";
    if ((files = fopen(F.c_str(), "w"))==NULL)
    {
        ShowMessage("Невозможно создать файл!");
        return;
    };
    OTCHET_ZAGOLOVOK(files,"Отчет по перемещению товара по складам");
    ////////////////////////////////////Выводим наименование товара////////////////////////////////////
    PERM->First();
    while (PERM->Eof==NULL)
    {
        if (PERM->FieldByName("NUM")->AsInteger==NUM)
        {
            FILESS(files,PERM->FieldByName("NAIMENOVANIE")->AsString,"Категория товара: ");
            FILESS(files,PERM->FieldByName("VID")->AsString,"Вид товара: ");
            PERM->Last();
        }
        PERM->Next();
    };
    LINIA(files);
    ////////////////////////////////////Выводим путь по складам////////////////////////////////////
    AnsiString PUT;
    int p=0;
    PERM->First();
    while (PERM->Eof==NULL)
    {
        if (PERM->FieldByName("NUM")->AsInteger==NUM)
        {
            p++;
            PUT="Перемещение №";
            PUT=PUT+VarToStr(p);
            MESSAGES(files,PUT);
            FILESS(files,PERM->FieldByName("SKLAD_OTPR")->AsString,"Находился на складе: ");
            FILESS(files,PERM->FieldByName("SKLAD_POL")->AsString,"Переместили на склад: ");
        }
        PERM->Next();
    };
    fclose(files);
    ShellExecute(NULL,"open",F.c_str(),NULL,NULL,SW_MAXIMIZE); //Открываем файл с отчетом
}
////////////////////////////////////+Метод класса////////////////////////////////////
void TForm1::TABL_CLASS::OTCHET_POKUPKA_ONE(AnsiString S, AnsiString C)
{
    FILE * files;AnsiString F=S;F=F+".html";
    if ((files = fopen(F.c_str(), "w"))==NULL)
    {
        ShowMessage("Невозможно создать файл!");
        return;
    };
    ////////////////////////////////////
    OTCHET_ZAGOLOVOK(files,"Отчет по покупке товара по серийному номеру");
    ////////////////////////////////////
    PRIH->First();
    while (PRIH->Eof==NULL)
    {
        if (PRIH->FieldByName("NUM")->AsString==StrToInt(C))
        {
            FILESS(files,PRIH->FieldByName("NUM")->AsString,"Серийный номер товара: ");
            FILESS(files,PRIH->FieldByName("NAIMENOVANIE")->AsString,"Категория товара: ");
            FILESS(files,PRIH->FieldByName("VID")->AsString,"Вид товара: ");
            FILESS(files,PRIH->FieldByName("ZAK")->AsString,"Закупочная цена товара: ");
            FILESS(files,PRIH->FieldByName("NDS")->AsString,"Закупочная цена товара с учетом НДС: ");
            FILESS(files,PRIH->FieldByName("DOLLAR")->AsString,"Курс $: ");
            FILESS(files,PRIH->FieldByName("SKLAD_POLUCHATEL")->AsString,"Склад получатель: ");
            LINIA(files);
        }
        PRIH->Next();
    };
    ////////////////////////////////////
    fclose(files);
    ShellExecute(NULL,"open",F.c_str(),NULL,NULL,SW_MAXIMIZE); //Открываем файл с отчетом
}
////////////////////////////////////+Метод класса////////////////////////////////////
void TForm1::TABL_CLASS::OTCHET_POKUPKA(AnsiString S, AnsiString C)
{
    FILE * files;
    AnsiString F=S;
    F=F+".html";
    if ((files = fopen(F.c_str(), "w"))==NULL)
    {
        ShowMessage("Невозможно создать файл!");
        return;
    };
    OTCHET_ZAGOLOVOK(files,"Отчет по покупке товаров по категории и виду");
    PRIH->First();
    while (PRIH->Eof==NULL)

```



```

{
    if (PRIH->FieldByName("NAIMENOVANIE")->AsString==C)
    {
        FILESS(files,PRIH->FieldByName("NUM")->AsString,"Серийный номер товара: ");
        FILESS(files,PRIH->FieldByName("NAIMENOVANIE")->AsString,"Категория товара: ");
        FILESS(files,PRIH->FieldByName("VID")->AsString,"Вид товара: ");
        FILESS(files,PRIH->FieldByName("ZAK")->AsString,"Закупочная цена товара: ");
        FILESS(files,PRIH->FieldByName("NDS")->AsString,"Закупочная цена товара с учетом НДС: ");
        FILESS(files,PRIH->FieldByName("DOLLAR")->AsString,"Курс $: ");
        FILESS(files,PRIH->FieldByName("SKLAD_POLUCHATEL")->AsString,"Склад получатель: ");
        LINIA(files);
    }
    PRIH->Next();
};
fclose(files);
ShellExecute(NULL,"open",F.c_str(),NULL,NULL,SW_MAXIMIZE); //Открываем файл с отчетом
}
//////////+Метод класса+//////////
void TForm1::TABL_CLASS::OTCHET_POK_KATEGOR(AnsiString S, AnsiString C)
{
    FILE * files;AnsiString F=S;F=F+".html";
    if ((files = fopen(F.c_str(), "w"))==NULL)
    {
        ShowMessage("Невозможно создать файл!");
        return;
    };
    OTCHET_ZAGOLOVOK(files,"Отчет по покупке товаров по категории");
    FILESS(files,C,"Категория товара: ");
    ////////////
    PRIH->First();
    int j=0;
    while (PRIH->Eof==NULL)
    {
        if (PRIH->FieldByName("NAIMENOVANIE")->AsString==C)
        {
            FILESS(files,PRIH->FieldByName("VID")->AsString,"Вид товара: ");
            j++;
        }
        PRIH->Next();
    };
    ////////////Количество проданного товара//////////
    AnsiString I;
    I="Было куплено всего: ";
    I=I+VarToStr(j);
    I=I+" шт.";
    MESSAGES(files,I);
    ////////////
    fclose(files);
    ShellExecute(NULL,"open",F.c_str(),NULL,NULL,SW_MAXIMIZE); //Открываем файл с отчетом
}

```