

Зміст

Текст основного модуля.....	1
Текст модуля обліку приходу товарів.....	4
Текст модуля переліку товарів.....	5
Текст модуля переліку контрагентів.....	9
Текст модуля обліку товарів.....	11
Текст модуля обліку груп товарів.....	13
Текст модуля обліку реалізації товарів.....	14
Текст модуля обліку статистики закуівлі товарів.....	16
Текст модуля обліку статистики продажу товарів.....	18
Текст модуля обліку товару на складі (наявність товару).....	20

Текст основного модуля

Ім'я файлу — main.pas

```
unit main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, Menus, DB, DBTables, jpeg, ExtCtrls;

type
  TMainForm = class(TForm)
    butt_Store: TBitBtn;
    butt_Buy: TBitBtn;
    butt_Sell: TBitBtn;
    butt_StatisticsBuy: TBitBtn;
    butt_StatisticsSell: TBitBtn;
    butt_Exit: TBitBtn;
    MainMenu1: TMainMenu;
    mm_Catalogs: TMenuItem;
    cat_Organisation: TMenuItem;
    cat_Contragents: TMenuItem;
    cat_GroupsOfNomenclature: TMenuItem;
    cat_Nomenclature: TMenuItem;
    cat_Contracts: TMenuItem;
    mm_Goods: TMenuItem;
    g_Store: TMenuItem;
    g_StatBuy: TMenuItem;
    g_StatSell: TMenuItem;
    mm_Operations: TMenuItem;
    op_Buy: TMenuItem;
    op_Sell: TMenuItem;
    mm_Reps: TMenuItem;
    rep_Price: TMenuItem;
    rep_AvailGoods: TMenuItem;
    rep_Buy: TMenuItem;
    rep_Sell: TMenuItem;
    mm_Exit: TMenuItem;
    mm_About: TMenuItem;
    About1: TMenuItem;
    Label1: TLabel;
    Label2: TLabel;
    Image1: TImage;
    procedure mm_ExitClick(Sender: TObject);
    procedure About1Click(Sender: TObject);
    procedure butt_ExitClick(Sender: TObject);
    procedure butt_StoreClick(Sender: TObject);
    procedure butt_BuyClick(Sender: TObject);
    procedure butt_SellClick(Sender: TObject);
    procedure butt_StatisticsBuyClick(Sender: TObject);
    procedure butt_StatisticsSellClick(Sender: TObject);
    procedure g_StoreClick(Sender: TObject);
    procedure op_BuyClick(Sender: TObject);
    procedure op_SellClick(Sender: TObject);
    procedure g_StatBuyClick(Sender: TObject);
    procedure g_StatSellClick(Sender: TObject);
    procedure rep_BuyClick(Sender: TObject);
    procedure rep_SellClick(Sender: TObject);
```

```

    procedure cat_ContragentsClick(Sender: TObject);
    procedure cat_OrganisationClick(Sender: TObject);
    procedure cat_GroupsOfNomenclatureClick(Sender: TObject);
    procedure cat_ContractsClick(Sender: TObject);
    procedure cat_NomenclatureClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    MainForm: TMainForm;

implementation

uses about, store, buy, sell, statbuy, statsell, dates, contragents,
    organisation, groups, contracts, goods;

{$R *.dfm}

procedure TMainForm.mm_ExitClick(Sender: TObject);
begin
    close
end;

procedure TMainForm.About1Click(Sender: TObject);
begin
    f_about.show
end;

procedure TMainForm.butt_ExitClick(Sender: TObject);
begin
    close
end;

procedure TMainForm.butt_StoreClick(Sender: TObject);
begin
    f_Store.show
end;

procedure TMainForm.butt_BuyClick(Sender: TObject);
begin
    f_buy.show
end;

procedure TMainForm.butt_SellClick(Sender: TObject);
begin
    f_sell.show
end;

procedure TMainForm.butt_StatisticsBuyClick(Sender: TObject);
begin
    f_statbuy.show
end;

procedure TMainForm.butt_StatisticsSellClick(Sender: TObject);
begin
    f_statsell.show
end;

```

```

procedure TMainForm.g_StoreClick(Sender: TObject);
begin
    butt_StoreClick(mainform)
end;

procedure TMainForm.op_BuyClick(Sender: TObject);
begin
    butt_BuyClick(mainform)
end;

procedure TMainForm.op_SellClick(Sender: TObject);
begin
    butt_SellClick(mainform)
end;

procedure TMainForm.g_StatBuyClick(Sender: TObject);
begin
    butt_StatisticsBuyClick(mainform)
end;

procedure TMainForm.g_StatSellClick(Sender: TObject);
begin
    butt_StatisticsSellClick(mainform)
end;

procedure TMainForm.rep_BuyClick(Sender: TObject);
begin
    f_dates.show
end;

procedure TMainForm.rep_SellClick(Sender: TObject);
begin
    //f_dates.showmodal
end;

procedure TMainForm.cat_ContragentsClick(Sender: TObject);
begin
    f_Contragents.Show
end;

procedure TMainForm.cat_OrganisationClick(Sender: TObject);
begin
    f_Organisation.Show
end;

procedure TMainForm.cat_GroupsOfNomenclatureClick(Sender: TObject);
begin
    f_Groups.Show
end;

procedure TMainForm.cat_ContractsClick(Sender: TObject);
begin
    f_Contracts.Show
end;

procedure TMainForm.cat_NomenclatureClick(Sender: TObject);
begin
    f_Goods.Show
end;

end.

```

Текст модуля обліку приходу товарів

Ім'я файлу — buy.pas

```
unit buy;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ComCtrls, ExtCtrls, DBCtrls, Grids, DBGrids,
  DB, DBTables, Mask;

type
  Tf_Buy = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    DateTimePicker1: TDateTimePicker;
    butt_Goods: TBitBtn;
    butt_Exit: TBitBtn;
    DataSource1: TDataSource;
    Query1: TQuery;
    DataSource2: TDataSource;
    Query2: TQuery;
    DBEdit1: TDBEdit;
    CheckBox1: TCheckBox;
    ComboBox1: TComboBox;
    DBEdit2: TDBEdit;
    DataSource3: TDataSource;
    Query3: TQuery;
    procedure butt_ExitClick(Sender: TObject);
    procedure butt_GoodsClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_Buy: Tf_Buy;

implementation

uses goods;
{$R *.dfm}

procedure Tf_Buy.butt_ExitClick(Sender: TObject);
begin
  close
end;

procedure Tf_Buy.butt_GoodsClick(Sender: TObject);
begin
  f_goods.Show
end;
```

```

procedure Tf_Buy.FormCreate(Sender: TObject);
begin
  Query2.Close;
  Query2.SQL.Text:='SELECT Name FROM Contragent WHERE ContrType="Постач"';
  Query2.Open;
  with Query2 do
  begin
    open;
    while not EOF do
    begin
      ComboBox1.items.add(FieldByName('Name').AsString);
      next;
    end;
  end;
end;

end;

end.

```

Текст модуля переліку товарів

Ім'я файлу — contracts.pas

```

unit contracts;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, Grids, DBGrids, Buttons, ExtCtrls, DBCtrls,
  DB, DBTables;

type
  Tf_Contracts = class(TForm)
    DBGrid1: TDBGrid;
    GroupBox1: TGroupBox;
    cb_From1: TCheckBox;
    cb_To1: TCheckBox;
    cb_From2: TCheckBox;
    cb_To2: TCheckBox;
    DateTimePicker1: TDateTimePicker;
    DateTimePicker2: TDateTimePicker;
    DateTimePicker3: TDateTimePicker;
    DateTimePicker4: TDateTimePicker;
    Label1: TLabel;
    Label2: TLabel;
    DBNavigator1: TDBNavigator;
    butt_Contragents: TBitBtn;
    butt_Exit: TBitBtn;
    DataSource1: TDataSource;
    Query1: TQuery;
    procedure butt_ExitClick(Sender: TObject);
    procedure butt_ContragentsClick(Sender: TObject);
    procedure cb_From1Click(Sender: TObject);
    procedure cb_To1Click(Sender: TObject);
    procedure cb_From2Click(Sender: TObject);

```

```

    procedure cb_To2Click(Sender: TObject);
    procedure DateTimePicker1CloseUp(Sender: TObject);
    procedure DateTimePicker1UserInput(Sender: TObject;
        const UserString: String; var DateAndTime: TDateTime;
        var AllowChange: Boolean);
    procedure DateTimePicker2CloseUp(Sender: TObject);
    procedure DateTimePicker2UserInput(Sender: TObject;
        const UserString: String; var DateAndTime: TDateTime;
        var AllowChange: Boolean);
    procedure DateTimePicker3CloseUp(Sender: TObject);
    procedure DateTimePicker3UserInput(Sender: TObject;
        const UserString: String; var DateAndTime: TDateTime;
        var AllowChange: Boolean);
    procedure DateTimePicker4CloseUp(Sender: TObject);
    procedure DateTimePicker4UserInput(Sender: TObject;
        const UserString: String; var DateAndTime: TDateTime;
        var AllowChange: Boolean);
private
    { Private declarations }
public
    { Public declarations }
end;
procedure selection();
procedure changedate();
var
    f_Contracts: Tf_Contracts;

implementation

uses contragents;

{$R *.dfm}

procedure Tf_Contracts.butt_ExitClick(Sender: TObject);
begin
    close
end;

procedure Tf_Contracts.butt_ContragentsClick(Sender: TObject);
begin
    f_Contracts.Show
end;

procedure Tf_Contracts.cb_From1Click(Sender: TObject);
begin
    selection()
end;

procedure Tf_Contracts.cb_To1Click(Sender: TObject);
begin
    selection()
end;

procedure Tf_Contracts.cb_From2Click(Sender: TObject);
begin
    selection()
end;

procedure Tf_Contracts.cb_To2Click(Sender: TObject);
begin
    selection()
end;

```

```

end;

procedure changedate();
var datel, date2: TDate;
begin
    Datel := 0;
    Date2 := 0;

    if (f_Contracts.cb_From1.Checked AND f_Contracts.cb_To1.Checked) Then
        if f_Contracts.DateTimePicker1.Date > f_Contracts.DateTimePicker2.Date Then
            Datel := f_Contracts.DateTimePicker1.Date
        else
            Datel := f_Contracts.DateTimePicker2.Date
    else if f_Contracts.cb_From1.Checked Then
        Datel := f_Contracts.DateTimePicker1.Date
    else if f_Contracts.cb_To1.Checked Then
        Datel := f_Contracts.DateTimePicker2.Date;

    if (f_Contracts.cb_From2.Checked AND f_Contracts.cb_To2.Checked) Then
        if f_Contracts.DateTimePicker3.Date < f_Contracts.DateTimePicker4.Date Then
            Date2 := f_Contracts.DateTimePicker3.Date
        else
            Date2 := f_Contracts.DateTimePicker4.Date
    else if f_Contracts.cb_From2.Checked Then
        Date2 := f_Contracts.DateTimePicker3.Date
    else if f_Contracts.cb_To2.Checked Then
        Date2 := f_Contracts.DateTimePicker4.Date;

    if (Datel <> 0) AND (Date2 <> 0) then
        if Datel > Date2 Then
            begin
                MessageBox(0, 'Дата початку дії не повинна бути пізнішою за дату
закінчення дії договору!', 'Магазин-склад', MB_OK+MB_ICONWARNING+MB_TASKMODAL);
                f_contracts.DateTimePicker1.Date := now();
                f_contracts.DateTimePicker2.Date := now();
                f_contracts.DateTimePicker3.Date := now();
                f_contracts.DateTimePicker4.Date := now();
            end;
        end;

procedure selection();
var from1, to1, from2, to2 :string;
begin
    changedate();

    from1:=datetostr(f_Contracts.DateTimePicker1.Date);
    to1:=datetostr(f_Contracts.DateTimePicker2.Date);
    from2:=datetostr(f_Contracts.DateTimePicker3.Date);
    to2:=datetostr(f_Contracts.DateTimePicker4.Date);

    f_Contracts.Query1.Close;
    f_Contracts.Query1.SQL.Text:='SELECT Contract.NContr, Contragent.Name,
Contract.Data1, Contract.Data2, Contract.Info FROM Contragent, Contract WHERE
Contract.ContrID=Contragent.ID';
    if f_Contracts.cb_From1.Checked then
        f_Contracts.Query1.SQL.Text := f_Contracts.Query1.SQL.Text + ' AND
Contract.Data1>="'+from1+'"';
    if f_Contracts.cb_To1.Checked then
        f_Contracts.Query1.SQL.Text := f_Contracts.Query1.SQL.Text + ' AND

```

```

Contract.Data1<="'+to1+'";
    if f_Contracts.cb_From2.Checked then
        f_Contracts.Query1.SQL.Text := f_Contracts.Query1.SQL.Text + ' AND
Contract.Data2>="'+from2+'";
    if f_Contracts.cb_To2.Checked then
        f_Contracts.Query1.SQL.Text := f_Contracts.Query1.SQL.Text + ' AND
Contract.Data2<="'+to2+'";
    f_Contracts.Query1.Open;
    f_Contracts.Query1.Refresh;
end;

procedure Tf_Contracts.DateTimePicker1CloseUp(Sender: TObject);
begin
    selection()
end;

procedure Tf_Contracts.DateTimePicker1UserInput(Sender: TObject;
    const UserString: String; var DateAndTime: TDateTime;
    var AllowChange: Boolean);
begin
    selection()
end;

procedure Tf_Contracts.DateTimePicker2CloseUp(Sender: TObject);
begin
    selection()
end;

procedure Tf_Contracts.DateTimePicker2UserInput(Sender: TObject;
    const UserString: String; var DateAndTime: TDateTime;
    var AllowChange: Boolean);
begin
    selection()
end;

procedure Tf_Contracts.DateTimePicker3CloseUp(Sender: TObject);
begin
    selection()
end;

procedure Tf_Contracts.DateTimePicker3UserInput(Sender: TObject;
    const UserString: String; var DateAndTime: TDateTime;
    var AllowChange: Boolean);
begin
    selection()
end;

procedure Tf_Contracts.DateTimePicker4CloseUp(Sender: TObject);
begin
    selection()
end;

procedure Tf_Contracts.DateTimePicker4UserInput(Sender: TObject;
    const UserString: String; var DateAndTime: TDateTime;
    var AllowChange: Boolean);
begin
    selection()
end;

end.

```

Текст модуля переліку контрагентів

Ім'я файлу — contragents.pas

```
unit contragents;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, DBCtrls, StdCtrls, Grids, DBGrids, Buttons, DB,
  DBTables;

type
  Tf_Contragents = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    butt_Exit: TBitBtn;
    RadioGroup1: TRadioGroup;
    RadioGroup2: TRadioGroup;
    DataSource1: TDataSource;
    Query1: TQuery;
    procedure butt_ExitClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure RadioGroup1Click(Sender: TObject);
    procedure RadioGroup2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
  procedure selection();
var
  f_Contragents: Tf_Contragents;

implementation

uses goods;

{$R *.dfm}

procedure Tf_Contragents.butt_ExitClick(Sender: TObject);
begin
  close
end;

procedure Tf_Contragents.RadioGroup1Click(Sender: TObject);
begin
  selection();
end;

procedure Tf_Contragents.RadioGroup2Click(Sender: TObject);
begin
  selection();
end;
```

```

procedure selection();
begin

    if (f_Contragents.RadioGroup1.Buttons[0].Checked) and
(f_Contragents.RadioGroup2.Buttons[0].Checked) then
    begin
        f_Contragents.query1.close;
        f_Contragents.query1.SQL.Text:='SELECT Name, ContrType, Organiz, Adress,
Phone, Bank, INN, EDRPOU FROM Contragent';
        f_Contragents.query1.Open
    end;

    if f_Contragents.RadioGroup1.Buttons[0].Checked and
f_Contragents.RadioGroup2.Buttons[1].Checked then
    begin
        f_Contragents.query1.close;
        f_Contragents.query1.SQL.Text:='SELECT Name, ContrType, Organiz, Adress,
Phone, Bank, INN, EDRPOU FROM Contragent WHERE Organiz="юп."';
        f_Contragents.query1.Open
    end;

    if f_Contragents.RadioGroup1.Buttons[0].Checked and
f_Contragents.RadioGroup2.Buttons[2].Checked then
    begin
        f_Contragents.query1.close;
        f_Contragents.query1.SQL.Text:='SELECT Name, ContrType, Organiz, Adress,
Phone, Bank, INN, EDRPOU FROM Contragent WHERE Organiz="физ."';
        f_Contragents.query1.Open
    end;

    if f_Contragents.RadioGroup1.Buttons[1].Checked and
f_Contragents.RadioGroup2.Buttons[0].Checked then
    begin
        f_Contragents.query1.close;
        f_Contragents.query1.SQL.Text:='SELECT Name, ContrType, Organiz, Adress,
Phone, Bank, INN, EDRPOU FROM Contragent WHERE ContrType="Поставч"';
        f_Contragents.query1.Open
    end;

    if f_Contragents.RadioGroup1.Buttons[1].Checked and
f_Contragents.RadioGroup2.Buttons[1].Checked then
    begin
        f_Contragents.query1.close;
        f_Contragents.query1.SQL.Text:='SELECT Name, ContrType, Organiz, Adress,
Phone, Bank, INN, EDRPOU FROM Contragent WHERE ContrType="Поставч" AND
Organiz="юп."';
        f_Contragents.query1.Open
    end;

    if f_Contragents.RadioGroup1.Buttons[1].Checked and
f_Contragents.RadioGroup2.Buttons[2].Checked then
    begin
        f_Contragents.query1.close;
        f_Contragents.query1.SQL.Text:='SELECT Name, ContrType, Organiz, Adress,
Phone, Bank, INN, EDRPOU FROM Contragent WHERE ContrType="Поставч" AND
Organiz="физ."';
        f_Contragents.query1.Open
    end;
end;

```

```

    if f_Contragents.RadioGroup1.Buttons[2].Checked and
f_Contragents.RadioGroup2.Buttons[0].Checked then
    begin
        f_Contragents.query1.close;
        f_Contragents.query1.SQL.Text:='SELECT Name, ContrType, Organiz, Adress,
Phone, Bank, INN, EDRPOU FROM Contragent WHERE ContrType="Клієнт"';
        f_Contragents.query1.Open
    end;

    if f_Contragents.RadioGroup1.Buttons[2].Checked and
f_Contragents.RadioGroup2.Buttons[1].Checked then
    begin
        f_Contragents.query1.close;
        f_Contragents.query1.SQL.Text:='SELECT Name, ContrType, Organiz, Adress,
Phone, Bank, INN, EDRPOU FROM Contragent WHERE ContrType="Клієнт" AND
Organiz="юр."';
        f_Contragents.query1.Open
    end;

    if f_Contragents.RadioGroup1.Buttons[2].Checked and
f_Contragents.RadioGroup2.Buttons[2].Checked then
    begin
        f_Contragents.query1.close;
        f_Contragents.query1.SQL.Text:='SELECT Name, ContrType, Organiz, Adress,
Phone, Bank, INN, EDRPOU FROM Contragent WHERE ContrType="Клієнт" AND
Organiz="фіз."';
        f_Contragents.query1.Open
    end;

end;

end.

```

Текст модуля обліку товарів

Ім'я файлу — goods.pas

```

unit goods;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, DBCtrls, StdCtrls, Buttons, Grids, DBGrids, DB,
    DBTables;

type
    Tf_Goods = class(TForm)
        DBGrid1: TDBGrid;
        DBNavigator1: TDBNavigator;
        gb_Selection: TGroupBox;
        cb_Group: TCheckBox;
        cb_Name: TCheckBox;
        Edit1: TEdit;
        butt_Groups: TBitBtn;
        butt_AvailGoods: TBitBtn;
    end;

```

```

    butt_Exit: TBitBtn;
    DataSource1: TDataSource;
    Query1: TQuery;
    Query2: TQuery;
    DataSource2: TDataSource;
    ComboBox1: TComboBox;
    procedure FormCreate(Sender: TObject);
    procedure butt_ExitClick(Sender: TObject);
    procedure butt_AvailGoodsClick(Sender: TObject);
    procedure butt_GroupsClick(Sender: TObject);
    procedure ComboBox1Select(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure cb_GroupClick(Sender: TObject);
    procedure cb_NameClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

procedure selection();

var
    f_Goods: Tf_Goods;

implementation

uses groups, store;

{$R *.dfm}

procedure Tf_Goods.FormCreate(Sender: TObject);
begin
    Query2.Close;
    Query2.SQL.Text:='SELECT DISTINCT GoodGroups.TypeName FROM GoodGroups';
    Query2.Open;
    with Query2 do
    begin
        open;
        while not EOF do
        begin
            ComboBox1.items.add(FieldByName('TypeName').AsString);
            next;
        end;
    end;
end;

procedure Tf_Goods.butt_ExitClick(Sender: TObject);
begin
    close
end;

procedure Tf_Goods.butt_AvailGoodsClick(Sender: TObject);
begin
    f_store.Show
end;

procedure Tf_Goods.butt_GroupsClick(Sender: TObject);
begin
    f_Groups.Show
end;

```

```

procedure Tf_Goods.ComboBox1Select(Sender: TObject);
begin
    selection()
end;

procedure Tf_Goods.Edit1Change(Sender: TObject);
begin
    selection()
end;

procedure selection();
begin
    f_goods.Query1.Close;
    f_goods.Query1.SQL.Clear;
    f_goods.Query1.SQL.Text:='SELECT T.ID, G.TypeName, T.Name, T.Unit,
T.PricePurch, T.PriceRealiz FROM Goods T, GoodGroups G WHERE T.GroupID = G.ID';
    if f_goods.cb_Group.Checked then
    begin
        f_goods.Query1.SQL.Text := f_goods.Query1.SQL.Text + ' AND G.TypeName =
:ChooseGroup';
        f_goods.Query1.ParamByName('ChooseGroup').AsString :=
f_goods.ComboBox1.Text;
    end;
    if f_goods.cb_Name.Checked then
    begin
        f_goods.Query1.SQL.Text := f_goods.Query1.SQL.Text + ' AND lower(T.Name)
LIKE lower(:ChooseName)';
        f_goods.Query1.ParamByName('ChooseName').AsString := '%' +
f_goods.Edit1.Text + '%';
    end;
    f_goods.Query1.Open;
    f_goods.Query1.Refresh;
end;

procedure Tf_Goods.cb_GroupClick(Sender: TObject);
begin
    selection()
end;

procedure Tf_Goods.cb_NameClick(Sender: TObject);
begin
    selection()
end;

end.

```

Текст модуля обліку груп товарів

Ім'я файлу — groups.pas

```

unit groups;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, DB, DBTables;

```

```

type
  Tf_Groups = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Button1: TButton;
    Query1: TQuery;
    DataSource1: TDataSource;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_Groups: Tf_Groups;
implementation

{$R *.dfm}

procedure Tf_Groups.Button1Click(Sender: TObject);
begin
  f_groups.Hide;
end;

end.

```

Текст модуля обліку реалізації товарів

Ім'я файлу — sell.pas

```

unit Sell;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ComCtrls, ExtCtrls, DBCtrls, Grids, DBGrids,
  DB, DBTables, Mask;

type
  Tf_Sell = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    ComboBox1: TComboBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    DateTimePicker1: TDateTimePicker;
    butt_Store: TBitBtn;
    butt_Exit: TBitBtn;
    DataSource1: TDataSource;
    Query1: TQuery;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    Query2: TQuery;
    CheckBox1: TCheckBox;
    DataSource2: TDataSource;
  end;

```

```

    DataSource3: TDataSource;
    Query3: TQuery;
    procedure butt_ExitClick(Sender: TObject);
    procedure butt_StoreClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure ComboBox1Select(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
    procedure selection();
var
    f_Sell: Tf_Sell;

implementation

uses store;

{$R *.dfm}

procedure Tf_Sell.butt_ExitClick(Sender: TObject);
begin
    close
end;

procedure Tf_Sell.butt_StoreClick(Sender: TObject);
begin
    f_store.Show;
end;

procedure Tf_Sell.FormCreate(Sender: TObject);
begin
    Query3.Close;
    Query3.SQL.Text:='SELECT Name FROM Contragent WHERE ContrType="Клієнт"';
    Query3.Open;
    with Query3 do
    begin
        open;
        while not EOF do
        begin
            ComboBox1.items.add(FieldByName('Name').AsString);
            next;
        end;
    end;
end;

end;

procedure Tf_Sell.CheckBox1Click(Sender: TObject);
begin
    selection();
end;

procedure Tf_Sell.ComboBox1Select(Sender: TObject);
begin
    selection();
end;
end;
end.

```

Текст модуля обліку статистики закуівлі товарів

Ім'я файлу — statbuy.pas

```
unit statbuy;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, Buttons, ExtCtrls, DBCtrls, Grids, DBGrids,
  DB, DBTables;

type
  Tf_StatBuy = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    GroupBox1: TGroupBox;
    cb_From: TCheckBox;
    cb_To: TCheckBox;
    DateTimePicker1: TDateTimePicker;
    DateTimePicker2: TDateTimePicker;
    butt_Buy: TBitBtn;
    butt_Exit: TBitBtn;
    DataSource1: TDataSource;
    Query1: TQuery;
    procedure butt_ExitClick(Sender: TObject);
    procedure butt_BuyClick(Sender: TObject);
    procedure butt_RepBuyClick(Sender: TObject);
    procedure DBGrid1CellClick(Column: TColumn);
    procedure cb_FromClick(Sender: TObject);
    procedure cb_ToClick(Sender: TObject);
    procedure DateTimePicker1CloseUp(Sender: TObject);
    procedure DateTimePicker2CloseUp(Sender: TObject);
    procedure DateTimePicker1UserInput(Sender: TObject;
      const UserString: String; var DateAndTime: TDateTime;
      var AllowChange: Boolean);
    procedure DateTimePicker2UserInput(Sender: TObject;
      const UserString: String; var DateAndTime: TDateTime;
      var AllowChange: Boolean);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
  procedure selection();
var
  f_StatBuy: Tf_StatBuy;

implementation

uses buy, dates;

{$R *.dfm}

procedure Tf_StatBuy.butt_ExitClick(Sender: TObject);
begin
  close
end;
```

```

procedure Tf_StatBuy.butt_BuyClick(Sender: TObject);
begin
    f_buy.show;
end;

procedure Tf_StatBuy.butt_RepBuyClick(Sender: TObject);
begin
    Application.CreateForm(Tf_Dates, f_Dates);
end;

procedure Tf_StatBuy.cb_FromClick(Sender: TObject);
begin
    selection();
end;

procedure Tf_StatBuy.cb_ToClick(Sender: TObject);
begin
    selection();
end;

procedure selection();
var from,too: string;
begin
    from:=datetostr(f_Statbuy.DateTimePicker1.Date);
    too:=datetostr(f_Statbuy.DateTimePicker2.Date);

    f_Statbuy.Query1.Close;
    f_Statbuy.Query1.SQL.Text:='SELECT I.ID, I.Data, K.Name,G.Name, C.QuantityInc,
C.PricePurch, C.QuantityInc*C.PricePurch FROM Income I, IncomeComp C, Goods G,
Contragent K WHERE I.ID=C.IncomeID AND C.GoodsID=G.ID AND I.ContrID=K.ID';
    if f_StatBuy.cb_From.Checked then
        f_Statbuy.Query1.SQL.Text := f_Statbuy.Query1.SQL.Text + ' AND
I.Data>="'+from+'"';
    if f_Statbuy.cb_To.Checked then
        f_Statbuy.Query1.SQL.Text := f_Statbuy.Query1.SQL.Text + ' AND
I.Data<="'+too+'"';
    f_Statbuy.Query1.Open;
    f_Statbuy.Query1.Refresh;
end;

procedure Tf_StatBuy.DateTimePicker1CloseUp(Sender: TObject);
begin
    selection()
end;

procedure Tf_StatBuy.DateTimePicker2CloseUp(Sender: TObject);
begin
    selection()
end;

procedure Tf_StatBuy.DateTimePicker1UserInput(Sender: TObject;
const UserString: String; var DateAndTime: TDateTime;
var AllowChange: Boolean);
begin
    selection()
end;

procedure Tf_StatBuy.DateTimePicker2UserInput(Sender: TObject;
const UserString: String; var DateAndTime: TDateTime;
var AllowChange: Boolean);

```

```

begin
    selection()
end;

end.

```

Текст модуля обліку статистики продажу товарів

Ім'я файлу — statsell.pas

```

unit statsell;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ComCtrls, StdCtrls, Buttons, ExtCtrls, DBCtrls, Grids, DBGrids,
    DB, DBTables;

type
    Tf_StatSell = class(TForm)
        DBGrid1: TDBGrid;
        DBNavigator1: TDBNavigator;
        GroupBox1: TGroupBox;
        cb_From: TCheckBox;
        cb_To: TCheckBox;
        DateTimePicker1: TDateTimePicker;
        DateTimePicker2: TDateTimePicker;
        butt_Sell: TBitBtn;
        butt_Exit: TBitBtn;
        DataSource1: TDataSource;
        Query1: TQuery;
        procedure butt_SellClick(Sender: TObject);
        procedure butt_RepSellClick(Sender: TObject);
        procedure butt_ExitClick(Sender: TObject);
        procedure DateTimePicker1UserInput(Sender: TObject;
            const UserString: String; var DateAndTime: TDateTime;
            var AllowChange: Boolean);
        procedure cb_FromClick(Sender: TObject);
        procedure cb_ToClick(Sender: TObject);
        procedure DateTimePicker2UserInput(Sender: TObject;
            const UserString: String; var DateAndTime: TDateTime;
            var AllowChange: Boolean);
        procedure DBGrid1DblClick(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
    procedure selection();
var
    f_StatSell: Tf_StatSell;

implementation

uses Sell, dates;

{$R *.dfm}

```

```

procedure Tf_StatSell.butt_SellClick(Sender: TObject);
begin
    f_sell.show
end;

procedure Tf_StatSell.butt_RepSellClick(Sender: TObject);
begin
    Application.CreateForm(Tf_dates, f_dates);
end;

procedure Tf_StatSell.butt_ExitClick(Sender: TObject);
begin
    close
end;

procedure Tf_StatSell.DateTimePicker1UserInput(Sender: TObject;
    const UserString: String; var DateAndTime: TDateTime;
    var AllowChange: Boolean);
begin
    selection();
end;

procedure selection();
var from,too: string;
begin
    from:=datetostr(f_Statsell.DateTimePicker1.Date);
    too:=datetostr(f_Statsell.DateTimePicker2.Date);
    f_Statsell.Query1.Close;
    f_Statsell.Query1.SQL.Text:='SELECT R.ID, R.Data, K.Name,G.Name, C.Quantity,
C.PriceRealiz, C.Quantity*C.PriceRealiz FROM Realiz R, RealizComp C, Goods G,
Contragent K WHERE R.ID=C.RealizID AND C.GoodsID=G.ID AND R.ContrID=K.ID';
    if f_Statsell.cb_From.Checked then
        f_Statsell.Query1.SQL.Text := f_Statsell.Query1.SQL.Text + ' AND
R.Data>="'+from+'";
    if f_statsell.cb_To.Checked then
        f_Statsell.Query1.SQL.Text := f_Statsell.Query1.SQL.Text + ' AND
R.Data<="'+too+'";
    f_Statsell.Query1.Open;
    f_Statsell.Query1.Refresh;
end;

procedure Tf_StatSell.cb_FromClick(Sender: TObject);
begin
    selection();
end;
procedure Tf_StatSell.cb_ToClick(Sender: TObject);
begin
    selection();
end;

procedure Tf_StatSell.DateTimePicker2UserInput(Sender: TObject;
    const UserString: String; var DateAndTime: TDateTime;
    var AllowChange: Boolean);
begin
    selection();
end;

procedure Tf_StatSell.DBGrid1DblClick(Sender: TObject);

end.

```

Текст модуля обліку товару на складі (наявність товару)

Ім'я файлу — store.pas

```
unit store;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, DBCtrls, StdCtrls, Buttons, Grids, DBGrids, DB,
  DBTables;

type
  Tf_Store = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    gb_Selection: TGroupBox;
    cb_Group: TCheckBox;
    cb_Name: TCheckBox;
    ComboBox1: TComboBox;
    Edit1: TEdit;
    butt_Groups: TBitBtn;
    butt_Goods: TBitBtn;
    butt_Exit: TBitBtn;
    Query1: TQuery;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    Query2: TQuery;
    procedure FormCreate(Sender: TObject);
    procedure butt_ExitClick(Sender: TObject);
    procedure butt_GroupsClick(Sender: TObject);
    procedure butt_AvailGoodsClick(Sender: TObject);
    procedure butt_GoodsClick(Sender: TObject);
    procedure ComboBox1Select(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure cb_GroupClick(Sender: TObject);
    procedure cb_NameClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

procedure selection();

var
  f_Store: Tf_Store;

implementation

uses groups, goods;

{$R *.dfm}

procedure Tf_Store.FormCreate(Sender: TObject);
begin
  Query2.Close;
  Query2.SQL.Text:='SELECT DISTINCT GoodGroups.TypeName FROM GoodGroups';
```

```

Query2.Open;
with Query2 do
begin
    open;
    while not EOF do
    begin
        ComboBox1.items.add(FieldByName('TypeName').AsString);
        next;
    end;
end;
end;

procedure Tf_Store.butt_ExitClick(Sender: TObject);
begin
    close
end;

procedure Tf_Store.butt_GroupsClick(Sender: TObject);
begin
    f_Groups.Show
end;

procedure Tf_Store.butt_GoodsClick(Sender: TObject);
begin
    f_Goods.Show;
end;

procedure Tf_Store.ComboBox1Select(Sender: TObject);
begin
    selection();
end;

procedure Tf_Store.Edit1Change(Sender: TObject);
begin
    selection();
end;

procedure Tf_Store.cb_GroupClick(Sender: TObject);
begin
    selection();
end;

procedure Tf_Store.cb_NameClick(Sender: TObject);
begin
    selection();
end;

procedure selection();
begin
    f_Store.Query1.Close;
    f_Store.Query1.SQL.Clear;
    f_Store.Query1.SQL.Text:='SELECT  T.ID, G.TypeName, T.Name, (Q.QuantityInc-
Q.QuantityRealiz), T.Unit, T.PricePurch, T.PriceRealiz FROM Goods T, GoodGroups
G, IncomeComp Q WHERE T.GroupID = G.ID AND (Q.QuantityInc-Q.QuantityRealiz) >
0';
    if f_Store.cb_Group.Checked then
    begin
        f_Store.Query1.SQL.Text := f_store.Query1.SQL.Text + ' AND G.TypeName =

```

```

:ChooseGroup';
    f_Store.Query1.ParamByName('ChooseGroup').AsString :=
f_store.ComboBox1.Text;
end;
if f_store.cb_Name.Checked then
begin
    f_store.Query1.SQL.Text := f_store.Query1.SQL.Text + ' AND lower(T.Name)
LIKE lower(:ChooseName)';
    f_store.Query1.ParamByName('ChooseName').AsString := '%' +
f_store.Edit1.Text + '%';
end;
f_Store.Query1.Open;
f_Store.Query1.Refresh;

end;

end.

```