**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

ФГБОУ ВПО ГЛАЗОВСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ ИМЕНИ В.Г. КОРОЛЕНКО

Колледж информационных и социальных коммуникаций

Отчёт

по дисциплине “Прикладное программирование”

Проект

Информационно поисковые системы “Склад”

Выполнил: студент

группы 33

Печенкин Н.А.

Научный руководитель:

Касаткин К.А

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Глазов 2017

**СПЕЦИФИКАЦИИ**

Название приложения: Склад.

База данных товаров, хранящихся на складе: наименование, единица измерения, цена единицы, количество, дата последнего завоза. Регистрация поступления товара (формирование приходной накладной) и отгрузки (расходная накладная). Вывод инвентарной ведомости.

Для работы с базой данных ничего устанавливать не нужно.

КОД ПРОГРАММЫ

unit main;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls,

DBGrids, EditBtn, Buttons, ExtCtrls, sqldb, Sqlite3DS, db, sqlite3conn;

type

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

Button2: TButton;

ButtonFind: TButton;

ButtonDel: TButton;

ButtonAdd: TButton;

DataSource: TDataSource;

DBGrid: TDBGrid;

Editprice: TEdit;

Editquantity: TEdit;

Editmodel: TEdit;

EditID: TEdit;

Editfirm: TEdit;

Edittype: TEdit;

Image1: TImage;

SQLite3Dataset: TSqlite3Dataset;

SQLite3Connection: TSQLite3Connection;

SQLQuery: TSQLQuery;

SQLTransaction: TSQLTransaction;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure ButtonAddClick(Sender: TObject);

procedure ButtonDelClick(Sender: TObject);

procedure ButtonFindClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ private declarations }

public

var nakl: text;

namen,vrem,buff,po:string;

{ public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.lfm}

{ TForm1 }

function p(a:string):string;

var i:integer;

s:string;

begin

p:='';

for i:=1 to 15-Length(a) do

s:=s+' ';

p:=s+' ';

end;

procedure TForm1.ButtonAddClick(Sender: TObject);

var i,quant,symm:integer;

ds:string;

begin

quant:=0;

symm:=0;

with SQLQuery do

begin

SQL.Clear;

SQL.Add('SELECT \* FROM sklad where id = :id');

ParamByName('id').Text:=Editid.Text;

Open;

end;

ds:=SQLQuery.FieldByName('id').AsString;

quant:=strtoint(SQLQuery.FieldByName('quantity').AsString)+strtoint(Editquantity.Text);

Edittype.Text:=SQLQuery.FieldByName('type').AsString;

Editfirm.Text:=SQLQuery.FieldByName('firm').AsString;

Editmodel.Text:=SQLQuery.FieldByName('name').AsString;

Editprice.Text:=SQLQuery.FieldByName('price').AsString;

symm:=strtoint(Editquantity.Text)\*strtoint(SQLQuery.FieldByName('price').AsString);

SQLQuery.Close;

if ds<>'' then

begin

SQLQuery.SQL.Text := 'update sklad set quantity =:quantity where id=:id;';

SQLQuery.ParamByName('quantity').AsString := inttostr(quant);

SQLQuery.ParamByName('id').AsString := ds;

SQLQuery.ExecSQL;

SQLTransaction.Commit;

SQLite3Dataset.Open;

end

else begin

ShowMessage('Успешно добавлена новая запись');

with SQLQuery do

begin

SQL.Clear;

SQL.Add('INSERT INTO sklad(id,type,firm,name,price,quantity,date) VALUES(:id,:type,:firm,:name,:price,:quantity,:date)');

ParamByName('id').Text:=Editid.Text;

ParamByName('type').Text:=Edittype.Text;

ParamByName('firm').Text:=Editfirm.Text;

ParamByName('name').Text:=Editmodel.Text;

ParamByName('date').Text:=datetostr(date);

ParamByName('price').Text:=(Editprice.Text);

ParamByName('quantity').Text:=(Editquantity.Text);

quant:=strtoint(Editquantity.Text);

symm:=strtoint(Editprice.Text)\*strtoint(Editquantity.Text);

ExecSQL;

SQLTransaction.Commit;

Close;

end;

SQLite3Dataset.Open;

end;

vrem:=timetostr(now);

for i:=1 to 10 do

if vrem[i]=':' then vrem[i]:='.';

namen:='kakietonakladnie\'+'nakladnaya\_vvoza\_za\_'+datetostr(date)+'\_'+vrem+'.txt';

AssignFile(nakl,namen);

Rewrite(nakl);

for i:=1 to 50 do

Write(nakl,'\*');

Writeln(nakl,' ');

Writeln(nakl,'это накладная ввоза ');

Writeln(nakl,'от try MLG Sklaga');

Writeln(nakl,'на склад завезено ');

Writeln(nakl,Editfirm.Text,' ',Edittype.Text,' ',Editmodel.Text);

Writeln(nakl,'в количестве ',Editquantity.Text,' с ценой ',Editprice.Text,' за еденицу');

Writeln(nakl,'id завезенного товара= ',Editid.Text);

Writeln(nakl,'Общая стоимость завезенного товара= ',symm);

Writeln(nakl,'Всего на складе данного товара находится ',quant,' штук');

for i:=1 to 50 do

Write(nakl,'\*');

CloseFile(nakl);

quant:=0;

symm:=0;

SQLQuery.Close;

SQLite3Dataset.Open;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

Editid.Text:='';

Edittype.Text:='';

Editfirm.Text:='';

Editmodel.Text:='';

Editprice.Text:='';

Editquantity.Text:='';

end;

procedure TForm1.Button2Click(Sender: TObject);

var i,ii:integer;

begin

vrem:=timetostr(now);

for i:=1 to 10 do

if vrem[i]=':' then vrem[i]:='.';

namen:='kakietonakladnie\'+'inventarnaya\_vedomost\_za\_'+datetostr(date)+'\_'+vrem+'.txt';

AssignFile(nakl,namen);

Rewrite(nakl);

for i:=1 to 113 do

Write(nakl,'\*');

writeln(nakl,' ');

Writeln(nakl,' ');

Writeln(nakl,' Инвентарная ведомость от try mlg склада');

Writeln(nakl,' За ',datetostr(date),', время создания ',timetostr(now));

writeln(nakl,'');

for i:=1 to 113 do

Write(nakl,'\*');

writeln(nakl,' ');

writeln(nakl,' ');

for i:=1 to 114 do

Write(nakl,'-');

writeln(nakl,' ');

writeln(nakl,'| id товара | тип товара | фирма | название | цена | количество | время завоза |');

for ii:=1 to 500 do begin

with SQLQuery do

begin

SQL.Clear;

SQL.Add('SELECT \* FROM sklad where id = :id');

ParamByName('id').Text:=inttostr(ii);

Open;

end;

if SQLQuery.FieldByName('firm').AsString<>'' then begin

buff:=''+SQLQuery.FieldByName('id').AsString+p(SQLQuery.FieldByName('id').AsString)+SQLQuery.FieldByName('type').AsString+p(SQLQuery.FieldByName('type').AsString)+SQLQuery.FieldByName('firm').AsString+p(SQLQuery.FieldByName('firm').AsString)+SQLQuery.FieldByName('name').AsString+p(SQLQuery.FieldByName('name').AsString)+SQLQuery.FieldByName('price').AsString+p(SQLQuery.FieldByName('price').AsString)+SQLQuery.FieldByName('quantity').AsString+p(SQLQuery.FieldByName('quantity').AsString)+SQLQuery.FieldByName('date').AsString+p(SQLQuery.FieldByName('date').AsString);

end;

SQLQuery.Close;

if( buff<>'') then begin

for i:=1 to 114 do

Write(nakl,'-');

writeln(nakl,'');

Writeln(nakl,'|',buff,'|');

buff:='';

end;

end;

for i:=1 to 114 do

Write(nakl,'-');

writeln(nakl,' ');

writeln(nakl,' ');

for i:=1 to 113 do

Write(nakl,'\*');

CloseFile(nakl);

end;

procedure TForm1.ButtonDelClick(Sender: TObject);

var i,quant,id,symm:integer;

begin

vrem:=timetostr(now);

for i:=1 to 10 do

if vrem[i]=':' then vrem[i]:='.';

namen:='kakietonakladnie\'+'nakladnaya\_vivoza\_za\_'+datetostr(date)+'\_'+vrem+'.txt';

AssignFile(nakl,namen);

Rewrite(nakl);

SQLite3Dataset.Close;

with SQLQuery do

begin

SQL.Clear;

SQL.Add('SELECT \* FROM sklad where id = :id');

ParamByName('id').Text:=Editid.Text;

Open;

end;

id:=strtoint(Editid.Text);

symm:=strtoint(editquantity.text)\*strtoint(SQLQuery.FieldByName('price').AsString);

for i:=1 to 50 do

Write(nakl,'\*');

Writeln(nakl,' ');

Writeln(nakl,'это накладная вывоза ');

Writeln(nakl,'от try MLG Sklaga');

writeln(nakl,'со склада вывзено ');

buff:='';

quant:=strtoint(SQLQuery.FieldByName('quantity').AsString)-strtoint(Editquantity.Text);

buff:=SQLQuery.FieldByName('firm').AsString+' '+SQLQuery.FieldByName('type').AsString+' '+SQLQuery.FieldByName('name').AsString;

writeln(nakl,buff);

buff:='в количестве '+editquantity.text+' '+'с ценой '+SQLQuery.FieldByName('price').AsString;

writeln(nakl,buff);

Write(nakl,'id товара ');

writeln(nakl,SQLQuery.FieldByName('id').AsString);

writeln(nakl,'На складе таких товаров осталось ',quant,' штук');

Writeln(nakl,'Общая сумма накладной= ',symm);

for i:=1 to 50 do

Write(nakl,'\*');

SQLQuery.Close;

//++

SQLQuery.SQL.Text := 'update sklad set quantity =:quantity where id=:id;';

SQLQuery.ParamByName('quantity').AsString := inttostr(quant);

SQLQuery.ParamByName('id').AsString := inttostr(id);

SQLQuery.ExecSQL;

SQLTransaction.Commit;

//++

SQLQuery.Close;

if quant<=0 then begin

with SQLQuery do

begin

SQL.Clear;

SQL.Add('DELETE FROM sklad where id=:ID');

ParamByName('ID').Text:=EditID.Text;

ExecSQL;

SQLTransaction.Commit;

Close;

end;

end;

buff:='';

SQLite3Dataset.Open;

CloseFile(nakl);

i:=0;

quant:=0;

id:=0;

symm:=0;

end;

procedure TForm1.ButtonFindClick(Sender: TObject);

begin

with SQLQuery do

begin

SQL.Clear;

SQL.Add('SELECT \* FROM sklad where name = :name');

ParamByName('name').Text:=Editmodel.Text;

Open;

end;

ShowMessage(SQLQuery.FieldByName('id').AsString);

SQLQuery.Close;

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

SQLite3Dataset.FileName:='sklad.db';

SQLite3Dataset.TableName:='sklad';

DataSource.DataSet:=SQLite3Dataset;

SQLite3Connection.DatabaseName:='sklad.db';

SQLite3Connection.Transaction:=SQLTransaction;

SQLTransaction.DataBase:=SQLite3Connection;

SQLQuery.DataBase:=SQLite3Connection;

SQLQuery.Transaction:=SQLTransaction;

try

SQLite3Dataset.Open;

SQLite3Connection.Connected:=True;

except

On E:Exception do

ShowMessage('Ошибка открытия базы: '+ E.Message);

end;

end;

end.

ОПИСАНИЕ ПРОГРАММЫ

Программа Склад может регистрировать поступление товара на склад, при вводе id, типа товара, производителя, модели, цены, количества, если товар ранее не был на складе, если был то достаточно ввести id и количество. При поступлении товара формируется приходная накладная. Выполнять поиск id товара по модели. Регистрировать отгрузки товара со склада в определенном количестве, для отгрузки необходимо указать id и количество необходимого товара, при этом формируется расходная накладная. Сгенерировать инвентарную ведомость по всему товару на складе.

**ФОРМУЛЯР**

Операционная система: Windows® XP

* Процессор: Intel Core 2 duo E4400
* Видеокарта: Intel HD Graphics
* ОЗУ — 512 MB
* Жесткий диск: 1 GB свободного места
* Наличие манипулятора мышь
* Наличие клавиатуры

**ОПИСАНИЕ ПРИМЕНЕНИЯ**

Программа Склад применяется в сфере торговли. Программа склад будет хорошим дополнением для любой торговой фирмы, так как она может автоматизировать приходные и расходные накладные, а также формировать инвентарную ведомость.

**РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА**

Использовать операционная систему: Windows® XP. Запустить программу склад. Программа может запускаться с USB-flash накопителя.

**РУКОВОДСТВО ПРОГРАММИСТА**

В случае замены базы данных нужно в программе изменить название базы данных и все поля что бы они были одинаковы как в программе и в базе данных. В запросах к базе данных нужно поменять поля, при этом сохраняя количество полей в запросе такое же, как количеством полей в базе данных. В базу данных можно добавлять новые таблицы и поля, при этом изменяя все запросы в программе. Уже в созданные таблицы можно добавлять новые поля, так же изменяя запросы. Модифицировать базу данных можно с помощью программы SQLite3.

**РУКОВОДСТВО ОПЕРАТОРА**

Для того что бы завести товар на склад нужно ввести все поля, если товара нет на складе или только id и количество, если товар на складе есть и нажать на кнопку завести товар. Для выполнения поиска необходимо указать модель необходимого товара и нажать на кнопку поиск, после программа выдаст id товара. Для того чтобы вывести товар необходимо указать только id и количество и нажать кнопку вывезти товар, если вывезти весь товар то запись с этим товаром будет удалена из базы данных. Чтобы получить инвентарную ведомость необходимо нажать на кнопку инвентарная ведомость.

**Описание языка**

Для создания программного продукта Склад использовался язык программирования Object Pascal и среда программирования Lazarus 1.6.4. Библиотека компонентов для SQLite3.

**РУКОВОДСТВО ПО ТЕХНИЧЕСКОМУ ОБСЛУЖИВАНИЮ**

Единственное, что нужно выяснить это минимальные требования к компьютеру они были указаны в формуляре. Для тестирования компьютера можно использовать программу (AIDA), если ваш компьютер удовлетворяет минимальным требованиям, то можно использовать программу Склад.

**Программа и методика испытания**

**Тест план.**

Проверить все функции программы Склад на корректность работы:

1. Ввоз товара
2. Поиск товара.
3. Вывоз товара.
4. Поиск вакансий.
5. Получение ведомости.

**Чек лист: Ввоз товара**

1. Проверка корректности ввода данных в поле id
2. Проверка корректности ввода данных в поле Тип товара.
3. Проверка корректности ввода данных в поле Производитель.
4. Проверка корректности ввода данных в поле Модель.
5. Проверка корректности ввода данных в поле Цена.
6. Проверка корректности ввода данных в поле Количество.

**Чек лист: Вывоз товара**

1. Проверка корректности удаления данных из поля id
2. Проверка корректности удаления данных из поля Тип товара.
3. Проверка корректности удаления данных из поля Производитель.
4. Проверка корректности удаления данных из поля Модель.
5. Проверка корректности удаления данных из поля Цена.
6. Проверка корректности удаления данных из поля Количество.

**Чек лист: Поиск id товара**

1. Проверка корректности поиска по полю Модель.

**Чек лист: Получения ведомости**

1. Проверка корректности вывода инвентарной ведомости

**Тест case: Добавление товара**

**Условия воспроизведения:**  
1.Установить программу Склад.

**Как воспроизвести:**  
2. Заполнить поле id  
3. Заполнить поле модель.

4. Заполнить поле Цена.

**Фактический результат:**

**Выдаётся ошибка**

**Ожидаемый результат:**  
При не корректном вводе данных в поле выводилось уведомление об это.

**Тест case: Вывоз товара**

**Условия воспроизведения:**  
1.Установить программу Склад.

**Как воспроизвести:**.  
2. Заполнить поле id  
3. Не заполнить поле количество

**Фактический результат:**  
**Выдаётся ошибка.**

**Ожидаемый результат:**  
При не корректном вводе данных в поле выводилось уведомление об это.

**Тест case:Поиск товара**

**Условия воспроизведения:**  
1.Установить программу Склад.

**Как воспроизвести:**  
2. Заполнить поле модель

**Фактический результат:**  
Корректные результаты при поиске товара

**Ожидаемый результат:**  
Корректные результаты при поиске товара.

**Тест case: Вывод накладной**

**Условия воспроизведения:**  
1.Установить программу Склад.

**Как воспроизвести:**  
1. Нажать на вкладку вывод накладной.

2. Проверить на корректность выводящиеся данные.

**Фактический результат:**  
**Выводятся корректные данные в накладной.**

**Ожидаемый результат:**  
**Выводятся корректные данные в накладной.**

**Пояснительная записка**

**Введение**

**Название программы: Склад.**

**Назначение и область применения**

Программа Склад применяется в сфере торговли. Программа склад будет хорошим дополнением для любой торговой фирмы, так как она может автоматизировать приходные и расходные накладные, а также формировать инвентарную ведомость.

**Технические характеристики**

1. Задачи: создать функционал для работы программы Склад, создать базу данных, написать пользовательское приложение.
2. Алгоритмы: алгоритмы в данной программе не применялись, потому что все алгоритмы были реализованы разработчиками SQLite.
3. Описание и обоснованный выбор организации входных и выходных данных: выходные и входные данные хранятся в 1 таблицах базы данных sklad. Описание и обоснованный выбор технических и программных средств: Lazarus был выбран, потому что там есть компоненты для работы с базой данных, SQLite был выбран, потому что он свободно распространяемый и имеет широкий спектр запросов, не требует сервера, легко переносится.

**Технико-экономические показатели.**

Программа Склад имеет преимущество перед ручной работой с товаром, так как действия автоматизированы, поиск данных станет легче, возможность формирования накладных и ведомостей.

**Источники**

1. Иванова Г.С. Технология программирования: учебник / Г.С. Иванова. — М. : КНОРУС, 2011. — 336 с. ISBN 978-5-406-00519-4
2. SQLite в Lazarus [Электронный ресурс]. – Режим доступа: <https://devlaz.ru/lazarus_sqlite/> (дата обращения 12.05.2017)
3. Lazarus - работа с базами данных [Электронный ресурс]. – Режим доступа: http://www.delphi-dev.ru/index.php/lazarus/lazarus-rabota-s-bazami-dannykh.html (дата обращения 12.05.2017)

**Словарь предметной области**

БД – База данных.

SQLite – Компактная встраиваемая реляционная база данных.

Sklad – название базы данных.

Id – ­идентификационный номер