Документация

по ТСП

в Задание 8

на тема :

Осветителни тела

Име:Илиян Костов

Ф.н:19621609

Група:4б

Курс:3

Спец: СИТ

Условия на заданието :

2. Указания за изпълнение на практическото задание:

Таблица за осветителните тела:

2042 Луминисцентна лампа RELAX 59 см 18W 100 350 6,49 20 ЛО брой

2043 Прожектор MASSIVE 500W 341 300 49,99 20 ПО брой

2044 Спот система ARIZONA 3x69 85 90 59,99 20 ОТ брой

2045 Аплик двоен 2хЕ27 60W 420 300 22,99 20 ОТ брой

2046 Халогенна крушка 50W 317 400 4,99 20 ЛО 3 броя

2047 Пендел “LAVIDA 1” 6x69 97 100 199,99 20 OT брой

2048 GRANADA аплик 2х40W хром 132 100 74,99 20 ОТ брой

2049 Пендел Кехлибар 460 300 45,99 20 ОТ брой

2050 Крушка софтон ES 16W E27 34 40 24,99 20 ЛO брой

2051 Светодиоден фенер CROSSER 205 300 35,99 20 ПО брой

2052 Крушка ENERGY SAVER 11W, E14 38 40 17,99 20 ЛО брой

т.е.

No тяло

Наименование

Налично количество

Минимално количество

Нето цена

ДДС

Код на осветителното тяло

Мярка

ОБРАБОТКИ:

1. Изведете данните за Наименование и Нето цена на всички стоки, чиято цена е над 30,00 лв;

2. Направете сведения с полетата : No тяло, Наименование, Нето цена, Код на осветителното тяло;

3. Сортирайте таблица “Осветителни тела” по Наименование и Нето цена;

4. Планира се намаление на цените за стоки от група осветителни тела (ОТ). Намерете цените на

осветителните тела при намаление с 5%. Използвайте за случая параметрична справка. Включва

полета Наименование и редуцираната чрез намалението Нето цена;

5. Направете справка – сведение за цените на стоките (No тяло, наименование, нето цена, ДДС;

Бруто цена). Създайте отчет върху тази справка;

6. Направете списък на стоките, чиято наличност е под min. Включете Наименование и с колко е под

min;

7. Разширете БД с поле Склад, въведете следните данни: лампите и крушките са в склад Б,

осветителните тела са в склад А, останалите стоки са в склад Е;

8. Направете списък за инвентаризация - Наименование, Налично количество, Нето цена, Обща

стойност. Да има графа за общата цена на стоки на склад. Осветителните тела да са групирани по

адрес на склад;

9. Направете меню със страници: Въвеждане, Извеждане, Справки, Отпечатване на справките (т.е.

отчетите) и всяко от тези менюта да съдържа свое подменю за съответните подоперации.

Резюме:

**Предназначение**

Направеното windows приложение предлага съвкупност от опции за съхраняване , модификация , извеждане и изтриване на осветителни тела със сътоветните му атрибути, като взаимодейства с потребителя чрез немодални диалогови кутии .

Едните от опциите , на които са предоставени на потребитля , с които той може да прави отчети на осветителните тела е създаването на справки под определени фиксирани критерии , за които те могат да бъдат ежедневно използвани от него .

Данни , с които програмата работи:

Програмата работи както с входни данни от потребителя ( чрез текстово поле ) , така и с избор на опция от съдържателно меню , заедно с натискане от мишката на потребителя бутон върху виртуален бутон , извършвайки необходимите действия , наименована кратко върху нея.

Основните данни , с които програмата работи е релационна база данни с определен тип данни, в зависимост от спецификата на атрибутите на дадения обект – в този случай е осветително тяло : то включва :

* Номер на тяло
* Наименование
* Налично количество
* Минимално количество
* Нето цена
* ДДС
* Код на осветително тяло
* Мярка

Възможни грешки и защита , включена в програмата

В програмата е предвидено защита в случай , че потребителят не е обучен или направи неволна грешка , като въведе някаква данна в текстово поле , невалидна за базата данни , следвайки непредвидена грешка или от базата данни , или от самото приложение.

За предотвратяване на това да се случи , програмата проверява предварително от потребителят въведеното и следователно се появява съобщение , информирайки го , че този вид данни не може да работи програмата с ( и базата данни ) като го връща обратно в първоначалното състояние , където потребителят е започнал .

Възможности за развитие на програмата

Програмата може му бъде добавена допълнителни опции на потребителя , с които той би желал да бъдат заложени в нея , като :

* допълнителни справки , с които потребителят дневно се сблъсква
* необходимост от регистрация в самото приложение , за неговото използване чрез въвеждане на потребителско име и парола ( регистрацията трябва да се направи лично от ръководителя ).
* издаване на права на използване на дадени опции , в зависимост колко важни данни се извеждат .

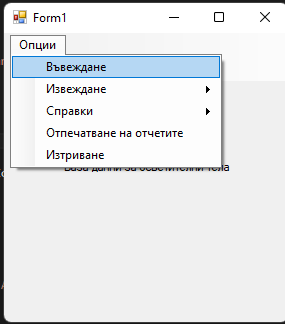
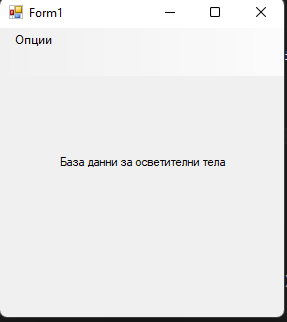
Опции:

Предоставят се възможности за взаимодействия със самата програма като:

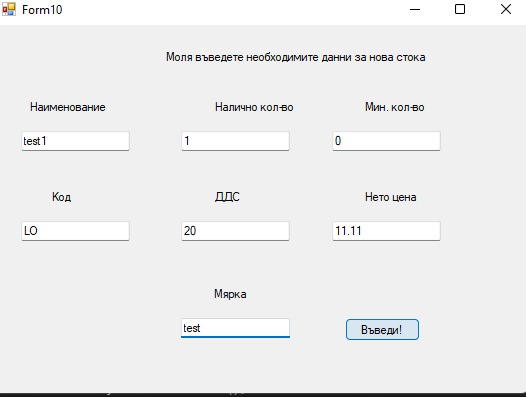
* добавяне на осветително тяло
* извеждане на всички светително тяло
* извеждане на осветителни тела с цена над 30 лв.
* Сведения на осветителни тела по атрибути
* Сортиране на осветителни тела по име и цена
* Намаление 5% по код на осветително тяло ЛО
* Справка на цените на осветително тяло с отчет
* Списък на стоки под минимум
* Добавяне на поле склад на всяко осветително тяло
* Списък на инвертиризации
* Меню със страници по зависимост от категории

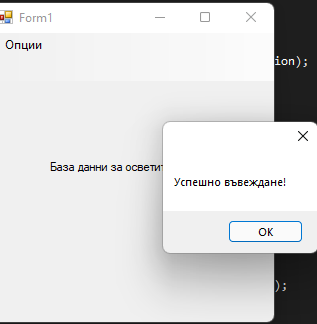
Тестов резултат

*Основна форма , с където се избират желаните резултати*

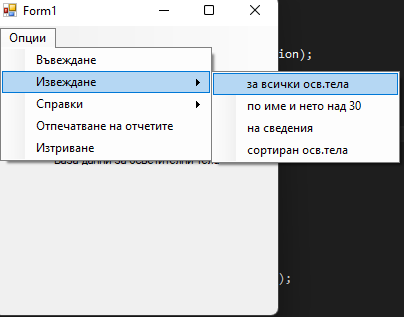


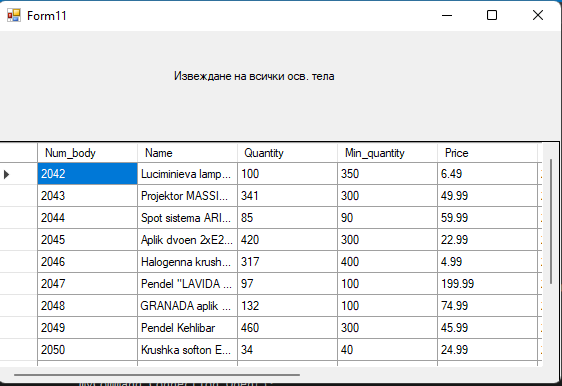
*Форма за въвеждане на ОТ*



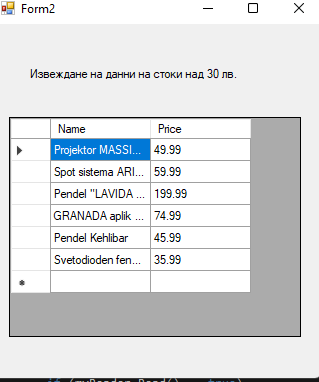


*Форма за извеждане на ОТ*

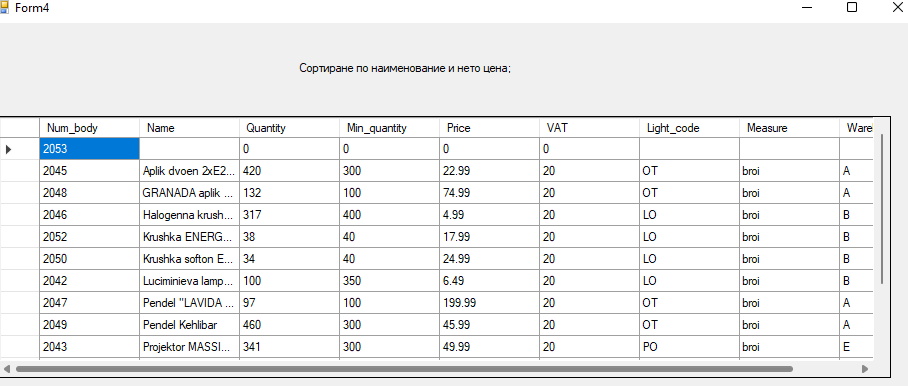




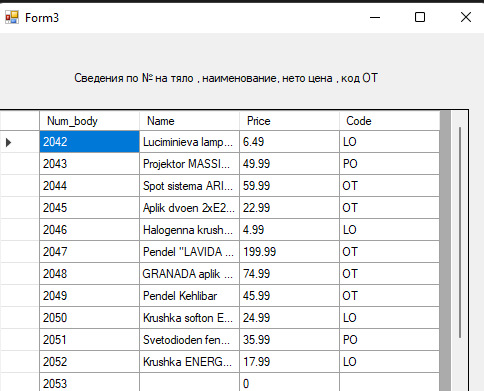
*Форма за извеждане на ОТ над 30лв*



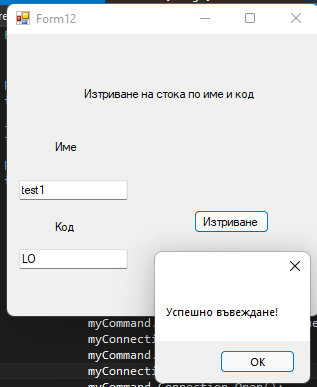
*Форма за извеждане на ОТ , сортирана на наименование и цена*



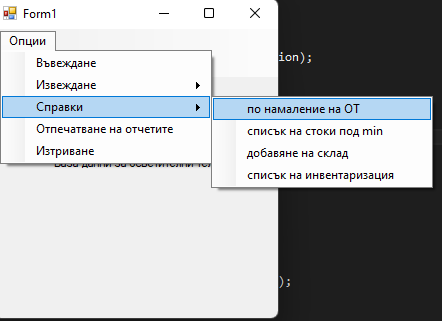
*Форма за извеждане на номер на тяло , наименование , цена и код на ОТ над 30лв*



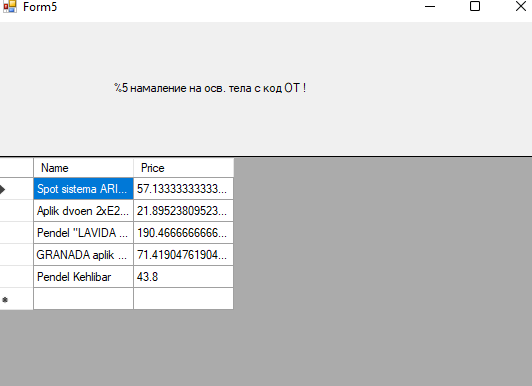
*Форма за изтриване на ОТ*



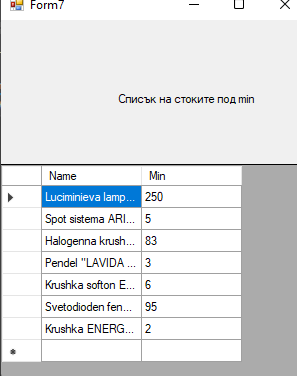
*Справки*



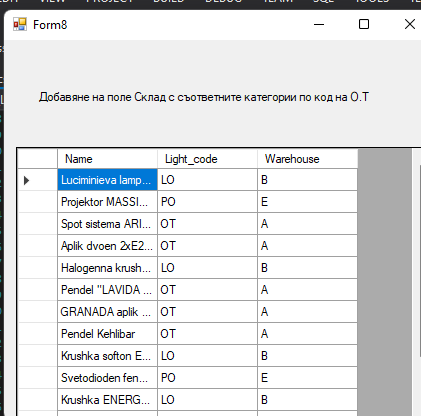
*Справки за намаление на ОТ с код ОТ*



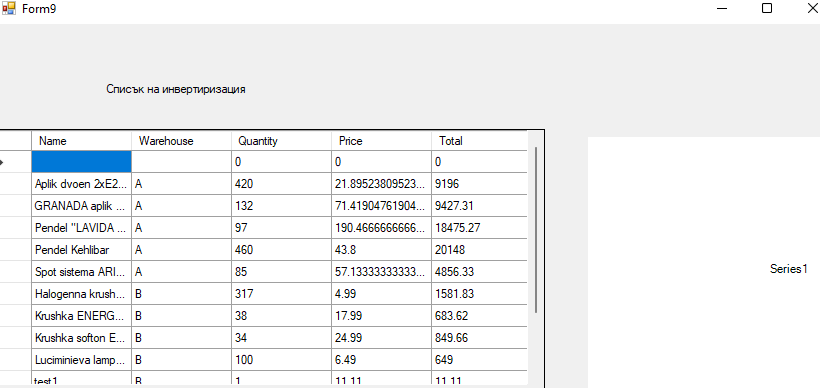
*Справки ОТ под минимум*



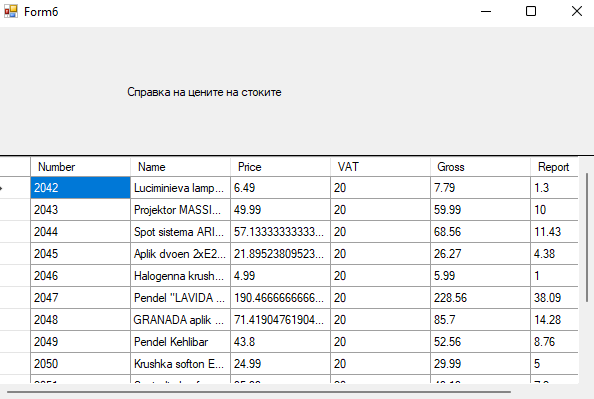
*Добавяне на поле склад към ОТ*



*Списък на инвертаризация на ОТ*



*Справка на цените на стоките*



Код

Form1

namespace WFA\_Lighting\_fixtures

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public string cs = @"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Local0\Documents\Visual Studio 2012\Projects\WFA\_Lighting fixtures\WFA\_Lighting fixtures\Database1.mdf;Integrated Security=True";

public SqlConnection myConnection = default(SqlConnection);

public SqlCommand myCommand = default(SqlCommand);

private void label1\_Click(object sender, EventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

label1.Text = "База данни за осветителни тела";

}

private void задача1ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form10 frm10 = new Form10();

frm10.Show();

}

private void имеИНетоНад30ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form frm2 = new Form2();

frm2.Show();

}

private void наСведенияToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form3 frm3 = new Form3();

frm3.Show();

}

private void сортиранОсвтелаToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form4 frm4 = new Form4();

frm4.Show();

}

private void поНамалениеНаОТToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form5 frm5 = new Form5();

frm5.Show();

}

private void списъкНаСтокиToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form7 frm7 = new Form7();

frm7.Show();

}

private void добавянеНаСкладToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form8 frm8 = new Form8();

frm8.Show();

}

private void списъкНаИнвертиризацияToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form9 frm9 = new Form9();

frm9.Show();

}

private void всичкиОстелаToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form11 frm11 = new Form11();

frm11.Show();

}

private void задача4ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form6 frm6 = new Form6();

frm6.Show();

}

private void изтриванеToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form12 frm12 = new Form12();

frm12.Show();

}

}

}

Form 2

namespace WFA\_Lighting\_fixtures

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

label1.Text = "Извеждане на данни на стоки над 30 лв.";

}

public SqlConnection myConnection;

public SqlCommand myCommand;

SqlDataAdapter adapt;

Form1 frm = new Form1();

private void displayData()

{

DataTable dt = new DataTable();

adapt = new SqlDataAdapter("select Name , Price from Light\_fixtures where Price >30.00", myConnection);

adapt.Fill(dt);

dataGridView1.DataSource = dt;

myConnection.Close();

}

private void Form2\_Load(object sender, EventArgs e)

{

try

{

myConnection = new SqlConnection(frm.cs);

displayData();

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();

}

}

}

Form 3

namespace WFA\_Lighting\_fixtures

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

label1.Text = "Сведения по № на тяло , наименование, нето цена , код ОТ";

}

public SqlConnection myConnection;

public SqlCommand myCommand;

SqlDataAdapter adapt;

Form1 frm = new Form1();

private void displayData()

{

DataTable dt = new DataTable();

adapt = new SqlDataAdapter("select Num\_body, Name, Price , Light\_code AS Code from Light\_fixtures", myConnection);

adapt.Fill(dt);

dataGridView1.DataSource = dt;

myConnection.Close();

}

private void Form3\_Load(object sender, EventArgs e)

{

try

{

myConnection = new SqlConnection(frm.cs);

displayData();

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();

}

}

}

Form 4

namespace WFA\_Lighting\_fixtures

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

label1.Text = "Сортиране по наименование и нето цена;";

}

public SqlConnection myConnection;

public SqlCommand myCommand;

SqlDataAdapter adapt;

Form1 frm = new Form1();

private void displayData()

{

DataTable dt = new DataTable();

adapt = new SqlDataAdapter("select \* from Light\_fixtures ORDER BY Name , Price DESC", myConnection);

adapt.Fill(dt);

dataGridView1.DataSource = dt;

myConnection.Close();

}

private void Form4\_Load(object sender, EventArgs e)

{

try

{

myConnection = new SqlConnection(frm.cs);

displayData();

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();

}

}

}

Form 5

namespace WFA\_Lighting\_fixtures

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

label1.Text = "%5 намаление на осв. тела с код ОТ !";

}

public SqlConnection myConnection;

public SqlCommand myCommand;

SqlDataAdapter adapt;

Form1 frm = new Form1();

private void displayData()

{

DataTable dt = new DataTable();

adapt = new SqlDataAdapter("select Name , Price from Light\_fixtures Where Light\_code='OT'", myConnection);

adapt.Fill(dt);

dataGridView1.DataSource = dt;

myConnection.Close();

}

private void Form5\_Load(object sender, EventArgs e)

{

try

{

myConnection = new SqlConnection(frm.cs);

myCommand = new SqlCommand("update Light\_fixtures set Price=Price / 1.05 where Light\_code='OT'", myConnection);

myConnection.Open();

myCommand.ExecuteNonQuery();

myConnection.Close();

displayData();

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();

}

}

}

Form 6

namespace WFA\_Lighting\_fixtures

{

public partial class Form6 : Form

{

public Form6()

{

InitializeComponent();

label1.Text = "Справка на цените на стоките";

}

public SqlConnection myConnection;

public SqlCommand myCommand;

SqlDataAdapter adapt;

Form1 frm = new Form1();

private void displayData()

{

DataTable dt = new DataTable();

adapt = new SqlDataAdapter("select Num\_body AS Number , Name , Price , VAT , ROUND(Price \* 1.20 ,2) AS Gross , ROUND( (Price\*1.20) - Price , 2) AS Report from Light\_fixtures", myConnection);

adapt.Fill(dt);

dataGridView1.DataSource = dt;

myConnection.Close();

}

private void Form6\_Load(object sender, EventArgs e)

{

try

{

myConnection = new SqlConnection(frm.cs);

displayData();

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();

}

}

}

Form 7

namespace WFA\_Lighting\_fixtures

{

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

label1.Text = "Списък на стоките под min";

}

private void Form7\_Load(object sender, EventArgs e)

{

try

{

myConnection = new SqlConnection(frm.cs);

displayData();

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

public SqlConnection myConnection;

public SqlCommand myCommand;

SqlDataAdapter adapt;

Form1 frm = new Form1();

private void displayData()

{

DataTable dt = new DataTable();

adapt = new SqlDataAdapter("select Name , ROUND( Min\_Quantity - Quantity ,2) AS Min from Light\_fixtures where Quantity < Min\_Quantity", myConnection);

adapt.Fill(dt);

dataGridView1.DataSource = dt;

myConnection.Close();

}

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();

}

}

}

Form 8

namespace WFA\_Lighting\_fixtures

{

public partial class Form8 : Form

{

public Form8()

{

InitializeComponent();

label1.Text = "Добавяне на поле Склад с съответните категории по код на О.Т ";

}

public SqlConnection myConnection;

public SqlCommand myCommand;

SqlDataAdapter adapt;

Form1 frm = new Form1();

private void displayData()

{

DataTable dt = new DataTable();

adapt = new SqlDataAdapter("select Name , Light\_code , Warehouse from Light\_fixtures", myConnection);

adapt.Fill(dt);

dataGridView1.DataSource = dt;

myConnection.Close();

}

private void Form8\_Load(object sender, EventArgs e)

{

try

{

myConnection = new SqlConnection(frm.cs);

myCommand = new SqlCommand("update Light\_fixtures set Warehouse='B' where Light\_code='LO'", myConnection);

myConnection.Open();

myCommand.ExecuteNonQuery();

myCommand = new SqlCommand("update Light\_fixtures set Warehouse='A' where Light\_code='OT'", myConnection);

myCommand.ExecuteNonQuery();

myCommand = new SqlCommand("update Light\_fixtures set Warehouse='E' where Light\_code='PO'", myConnection);

myCommand.ExecuteNonQuery();

myConnection.Close();

displayData();

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();

}

}

}

Form 9

namespace WFA\_Lighting\_fixtures

{

public partial class Form9 : Form

{

public Form9()

{

InitializeComponent();

label1.Text = "Списък на инвертиризация";

}

public SqlConnection myConnection;

public SqlCommand myCommand;

SqlDataAdapter adapt;

Form1 frm = new Form1();

private void displayData()

{

DataTable dt = new DataTable();

adapt = new SqlDataAdapter("select Name , Warehouse , Quantity , Price , ROUND (SUM (Price\*Quantity),2) AS Total from Light\_fixtures GROUP BY Warehouse , Name , Quantity , Price", myConnection);

adapt.Fill(dt);

dataGridView1.DataSource = dt;

myConnection.Close();

}

private void Form9\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'database1DataSet1.Light\_fixtures' table. You can move, or remove it, as needed.

this.light\_fixturesTableAdapter.Fill(this.database1DataSet1.Light\_fixtures);

try

{

myConnection = new SqlConnection(frm.cs);

displayData();

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

private void chart1\_Click(object sender, EventArgs e)

{

}

private void chart1\_ControlAdded(object sender, ControlEventArgs e)

{

}

private void chart1\_Click\_1(object sender, EventArgs e)

{

}

}

}

Form 10

namespace WFA\_Lighting\_fixtures

{

public partial class Form10 : Form

{

public Form10()

{

InitializeComponent();

label1.Text = "Моля въведете необходимите данни за нова стока";

label2.Text = "Наименование";

label3.Text = "Налично кол-во";

label4.Text = "Мин. кол-во";

label5.Text = "Нето цена";

label6.Text = "ДДС";

label7.Text = "Код";

label8.Text = "Мярка";

button1.Text = "Въведи!";

}

public SqlConnection myConnection;

public SqlCommand myCommand;

Form1 frm = new Form1();

private void Form10\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

if (label1.Text != "" && label2.Text != "" && label3.Text != "" && label4.Text != "" && label5.Text != "" && label6.Text != "" && label7.Text != "" && label8.Text != "")

{

try

{

myConnection = new SqlConnection(frm.cs);

myCommand = new SqlCommand("insert into Light\_fixtures(Name,Quantity,Min\_quantity,Price,VAT,Light\_code,Measure) values(@Name,@Quantity,@Min\_quantity,@Price,@VAT,@Light\_code,@Measure)", myConnection);

myCommand.Parameters.AddWithValue("@Name" , textBox1.Text);

myCommand.Parameters.AddWithValue("@Quantity" , textBox2.Text);

myCommand.Parameters.AddWithValue("@Min\_quantity", textBox3.Text);

myCommand.Parameters.AddWithValue("@Price" , textBox4.Text);

myCommand.Parameters.AddWithValue("@VAT" , textBox5.Text);

myCommand.Parameters.AddWithValue("@Light\_code" , textBox6.Text);

myCommand.Parameters.AddWithValue("@Measure" , textBox7.Text);

myConnection.Open();

myCommand.ExecuteNonQuery();

myConnection.Close();

MessageBox.Show("Успешно въвеждане!");

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

else MessageBox.Show("Моля въведете необходимите данни!");

}

}

}

Form 11

namespace WFA\_Lighting\_fixtures

{

public partial class Form11 : Form

{

public Form11()

{

InitializeComponent();

label1.Text = "Извеждане на всички осв. тела";

}

public SqlConnection myConnection;

public SqlCommand myCommand;

SqlDataAdapter adapt;

Form1 frm = new Form1();

private void displayData()

{

DataTable dt = new DataTable();

adapt = new SqlDataAdapter("select \* from Light\_fixtures", myConnection);

adapt.Fill(dt);

dataGridView1.DataSource = dt;

myConnection.Close();

}

private void Form11\_Load(object sender, EventArgs e)

{

try

{

myConnection = new SqlConnection(frm.cs);

displayData();

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();

}

}

}

Form 12

namespace WFA\_Lighting\_fixtures

{

public partial class Form12 : Form

{

public Form12()

{

InitializeComponent();

label1.Text="Изтриване на стока по име и код";

label4.Text="Име";

label5.Text="Код";

button1.Text="Изтриване";

}

public SqlConnection myConnection;

public SqlCommand myCommand;

Form1 frm = new Form1();

private void Form12\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

if (textBox3.Text != "" && textBox4.Text != "")

{

myConnection = new SqlConnection(frm.cs);

myCommand = new SqlCommand("select \* from Light\_fixtures where Name=@Name AND Light\_code=@Light\_code", myConnection);

myCommand.Parameters.AddWithValue("@Name", textBox3.Text);

myCommand.Parameters.AddWithValue("@Light\_code", textBox4.Text);

myConnection.Open();

myCommand.ExecuteNonQuery();

myConnection.Close();

myCommand.Connection.Open();

SqlDataReader myReader = myCommand.ExecuteReader(CommandBehavior.CloseConnection);

if (myReader.Read() == true)

{

MessageBox.Show("Успешно въвеждане!");

myConnection.Close();

myConnection.Open();

myCommand = new SqlCommand("delete Light\_fixtures where Name=@Name AND Light\_code=@Light\_code", myConnection);

myCommand.Parameters.AddWithValue("@Name", textBox3.Text);

myCommand.Parameters.AddWithValue("@Light\_code", textBox4.Text);

myCommand.ExecuteNonQuery();

}

else MessageBox.Show("Стоката не е намерена!");

myConnection.Close();

if (myConnection.State == ConnectionState.Open)

{

myConnection.Dispose();

}

}

else MessageBox.Show("Моля въведете необходимите данни!");

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Error"); }

}

}

}