МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Одеський національний політехничний університет

Інститут Комп'ьютерних Систем

Кафедра Інформаційних Систем

Отчет по лабораторной работе №3

«Связный список. Хэш таблицы». «Бинарные деревья»

Выполнил:

студент группы АА-171

Поликарпов А.В.

Проверил:

Шибаева Н. О.

Одесса 2019

**Цель :** Приобретение практических навыков программирования с использованием: списков, хеш-таблицы, бинарные деревья.

**Вариант 16.** Разработать базу данных состоящую из 3 таблиц для медицинской организации. Обеспечить связь таблиц методом хеш-таблица. Организовать возможность добавление/удаление/редактирование данных. Разработать поиск и учетом возможности добавления поисковых фильтров. В готовом программном решении должен присутствовать графический интерфейс с навигационными элементами.

**Код:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab3TA

{

public partial class Form1 : Form

{

SqlConnection sqlConnection;

public Form1()

{

InitializeComponent();

}

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

{

string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\Сашка\Kursovaya\kukursovaya\ТА\Lab3\Lab3TA\Lab3TA\Database1.mdf;Integrated Security=True";

sqlConnection = new SqlConnection(connectionString);

await sqlConnection.OpenAsync();

SqlDataReader sqlReader = null;

SqlCommand command = new SqlCommand("SELECT \* FROM[Medicina]", sqlConnection);

try

{

sqlReader = await command.ExecuteReaderAsync();

while (await sqlReader.ReadAsync())

{

listBox1.Items.Add(Convert.ToString(sqlReader["Id"]) + " " + Convert.ToString(sqlReader["Name"]) + " " + Convert.ToString(sqlReader["Price"]));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (sqlReader != null)

sqlReader.Close();

}

}

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

{

if (sqlConnection != null && sqlConnection.State != ConnectionState.Closed)

sqlConnection.Close();

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

if (sqlConnection != null && sqlConnection.State != ConnectionState.Closed)

sqlConnection.Close();

}

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

{

if (label7.Visible)

label7.Visible = false;

if (!string.IsNullOrEmpty(textBox1.Text) && !string.IsNullOrEmpty(textBox1.Text) &&

!string.IsNullOrEmpty(textBox2.Text) && !string.IsNullOrEmpty(textBox2.Text))

{

SqlCommand command = new SqlCommand("INSERT INTO [Medicina] (Name, Price)VALUES(@Name, @Price)", sqlConnection);

command.Parameters.AddWithValue("Name", textBox1.Text);

command.Parameters.AddWithValue("Price", textBox2.Text);

await command.ExecuteNonQueryAsync();

}

else

{

label7.Visible = true;

label7.Text = "Поля 'Имя' и 'Цена' должны быть заполнены";

}

}

private async void обновитьToolStripMenuItem\_Click(object sender, EventArgs e)

{

listBox1.Items.Clear();

SqlDataReader sqlReader = null;

SqlCommand command = new SqlCommand("SELECT \* FROM[Medicina]", sqlConnection);

try

{

sqlReader = await command.ExecuteReaderAsync();

while (await sqlReader.ReadAsync())

{

listBox1.Items.Add(Convert.ToString(sqlReader["Id"]) + " " + Convert.ToString(sqlReader["Name"]) + " " + Convert.ToString(sqlReader["Price"]));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (sqlReader != null)

sqlReader.Close();

}

}

private async void button2\_Click(object sender, EventArgs e)

{

if (label8.Visible)

label8.Visible = false;

if (!string.IsNullOrEmpty(textBox3.Text) && !string.IsNullOrEmpty(textBox3.Text) &&

!string.IsNullOrEmpty(textBox4.Text) && !string.IsNullOrEmpty(textBox4.Text) &&

!string.IsNullOrEmpty(textBox5.Text) && !string.IsNullOrEmpty(textBox5.Text))

{

SqlCommand command = new SqlCommand("UPDATE [Medicina] SET [Name]=@Name, [Price]=@Price WHERE [Id]=@Id", sqlConnection);

command.Parameters.AddWithValue("Id", textBox3.Text);

command.Parameters.AddWithValue("Name", textBox4.Text);

command.Parameters.AddWithValue("Price", textBox5.Text);

await command.ExecuteNonQueryAsync();

}

else if (!string.IsNullOrEmpty(textBox3.Text) && !string.IsNullOrEmpty(textBox3.Text))

{

label8.Visible = true;

label8.Text = "Id должнен быть заполнен!";

}

else

{

label8.Visible = true;

label8.Text = "Поля 'Id', 'Имя' и 'Цена' должны быть заполнены";

}

}

private async void button3\_Click(object sender, EventArgs e)

{

if (label9.Visible)

label9.Visible = false;

if (!string.IsNullOrEmpty(textBox6.Text) && !string.IsNullOrEmpty(textBox6.Text))

{

SqlCommand command = new SqlCommand("DELETE FROM [Medicina] WHERE [Id]=@Id", sqlConnection);

command.Parameters.AddWithValue("Id", textBox6.Text);

await command.ExecuteNonQueryAsync();

}

else

{

label9.Visible = true;

label9.Text = "Id должнен быть заполнен!";

}

}

}

}

**Блок-схема:**



**Скриншоты:**

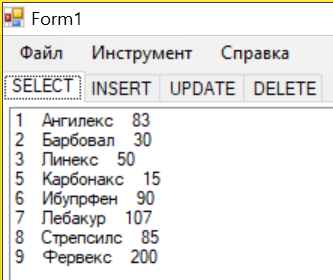


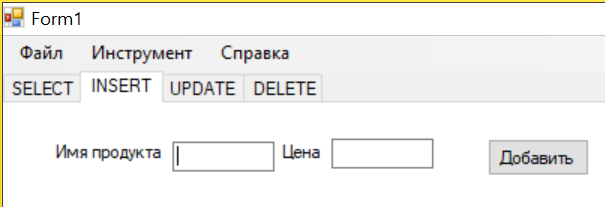
Рис.1 – Вывод БД на экран

Рис.2 – Вкладка добавления продукт в БД

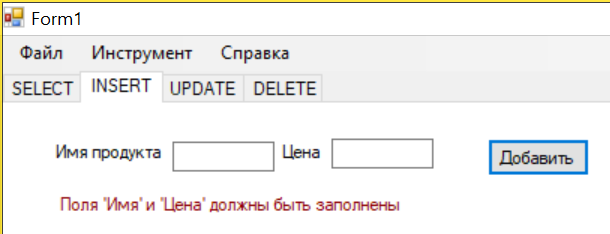


Рис.3 – выведение ошибок незаполненных полей

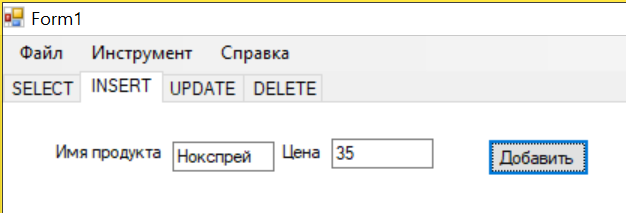


Рис.4 – Добавление нового товара

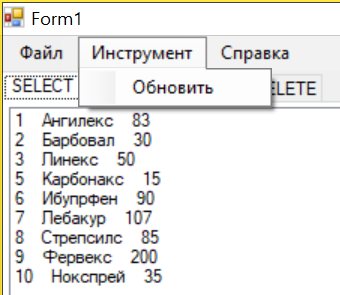


Рис.5 – обновление и добавление поля в БД

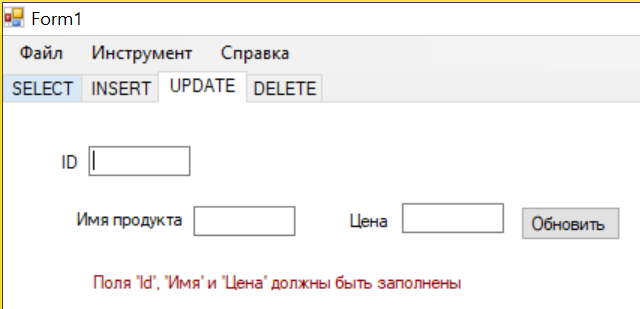


Рис.6 - выведение ошибок незаполненных полей

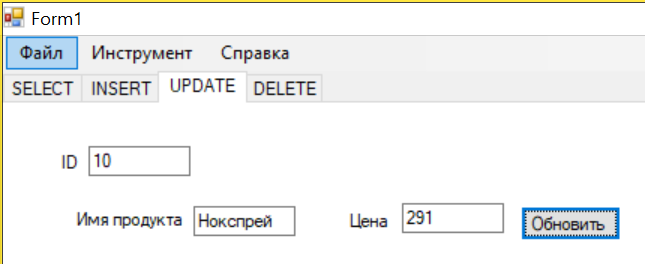


Рис.7 – изменение товара по ID

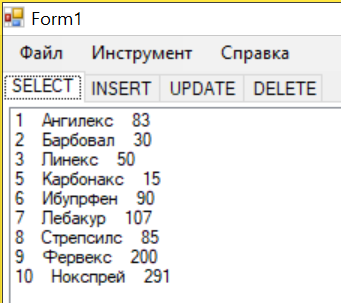


Рис.8 – обновление после изменения строки

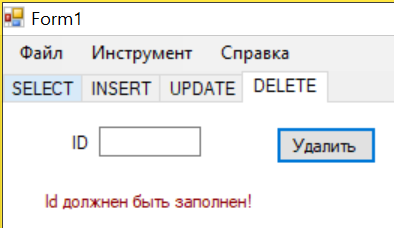


Рис.9 - выведение ошибки незаполненного поля

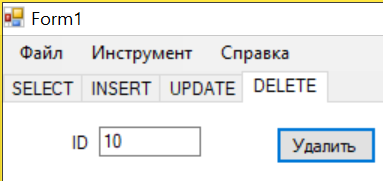


Рис.10 – удаления товара по ID

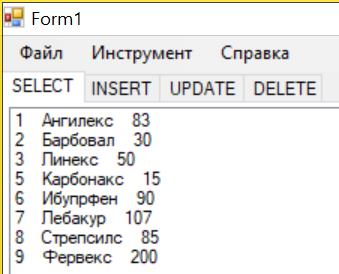


Рис.11 – обновление БД после удаления строки

**Вывод:** в этой лабораторной работе ознакомившись с теоретическим материалом я создал алгоритмы в соответствии с моим вариантом и задание, воспользовавшись данными которые были даны в ходе процесса созданы блок-схемы в соответствии к каждому блоку свое значение, с описание действий к каждой блок-схеме. Это было полезно для меня, я узнал новое и выучил обозначения к блокам.