**ЛАБОРАТОРНОЯ РАБОТА 6**

**«Работа с SQLite»**

**Цель:** Научится работать с GitHub репозиторием.

**Задание:**

Разработать программу для учета расходов с реализацией хранения данных в локальной БД SQLite и имеющую графический интерфейс на основе Windows Forms.

Предусмотреть следующий функционал:

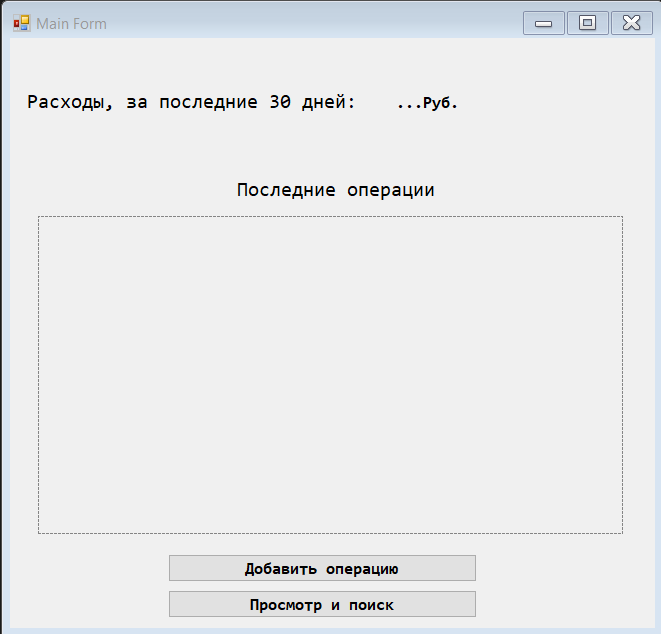
* Добавление новой записи по расходам (это могут быть любые траты, платежи, покупки), которая характеризуется следующими параметрами:  
  1. Размер платежа  
  2. Дата платежа (на выбор: текущая дата или другая)  
  3. Категория (еда, транспорт, жилье, другое и т.д)

4. Название

5. Комментарий

* Удаление выбранной записи
* Просмотр всех записей
* Поиск и фильтрация записей по выбранным параметрам (по названию, по дате, по размеру платежа, по категории)

**Выполнение задания:**

****

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Drawing;

using System.Windows.Forms;

namespace ExpensesApp

{

public partial class Form1 : Form

{

ExpenseTrackerDatabase db;

public Form1()

{

InitializeComponent();

string projectPath = AppContext.BaseDirectory + @"\ExpensesDatabase.db";

db = new ExpenseTrackerDatabase(projectPath);

db.CreateExpenseTable();

CountExpenses();

SumUpdate();

}

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

{

SumUpdate();

}

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

{

AddOperationForm add = new AddOperationForm(db, this);

add.Show();

}

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

{

SearchOperationsForm search = new SearchOperationsForm(db);

search.Show();

}

public void CountExpenses()

{

flowLayoutPanel1.Controls.Clear();

List<string> lastNExpenses = db.GetLastNExpenses(7);

foreach (string expense in lastNExpenses)

{

Debug.WriteLine(expense);

Label label = new Label

{

AutoSize = true,

ForeColor = Color.Green,

Font = label3.Font,

Text = expense

};

flowLayoutPanel1.Controls.Add(label);

}

}

public void SumUpdate()

{

int sum = 0;

sum = db.GetSum(sum);

label2.Text = sum.ToString();

}

private void flowLayoutPanel1\_ControlAdded(object sender, ControlEventArgs e)

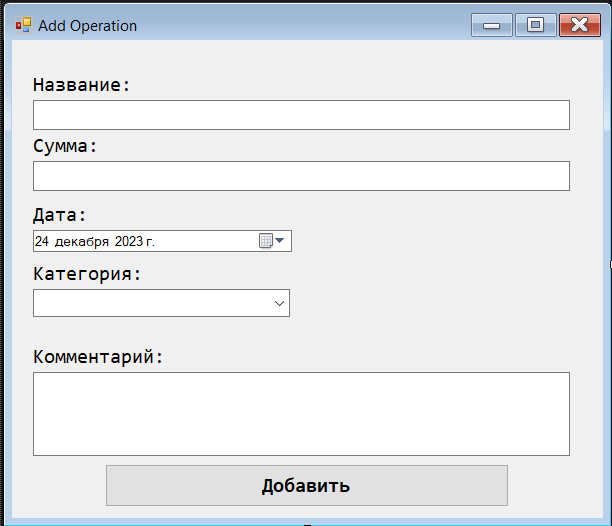
{

SumUpdate();

}

}

}

****

using System;

using System.Linq;

using System.Windows.Forms;

namespace ExpensesApp

{

public partial class AddOperationForm : Form

{

DateTime selectedDate;

ExpenseTrackerDatabase database;

Form1 formMain;

public AddOperationForm(ExpenseTrackerDatabase db, Form1 form)

{

InitializeComponent();

comboBox1.DropDownStyle = ComboBoxStyle.DropDownList;

this.database = db;

this.formMain = form;

}

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

{

if (selectedDate != null && !string.IsNullOrEmpty(textBoxNameOp.Text) && !string.IsNullOrEmpty(textBoxCommentOp.Text)

&& !string.IsNullOrEmpty(textBoxSumOp.Text) && !string.IsNullOrEmpty(comboBox1.Text))

{

if (!textBoxSumOp.Text.Any(char.IsLetter) && !textBoxSumOp.Text.Any(c => !char.IsLetterOrDigit(c) && !char.IsWhiteSpace(c)))

{

string name = textBoxNameOp.Text;

double sum = Convert.ToDouble(textBoxSumOp.Text);

string category = comboBox1.Text;

string comment = textBoxCommentOp.Text;

database.InsertExpense(name, sum, category, selectedDate, comment);

formMain.CountExpenses();

this.Close();

}

else

{

MessageBox.Show("Введите корректную сумму!");

}

}

else

{

MessageBox.Show("Не все данные были заполненны");

}

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

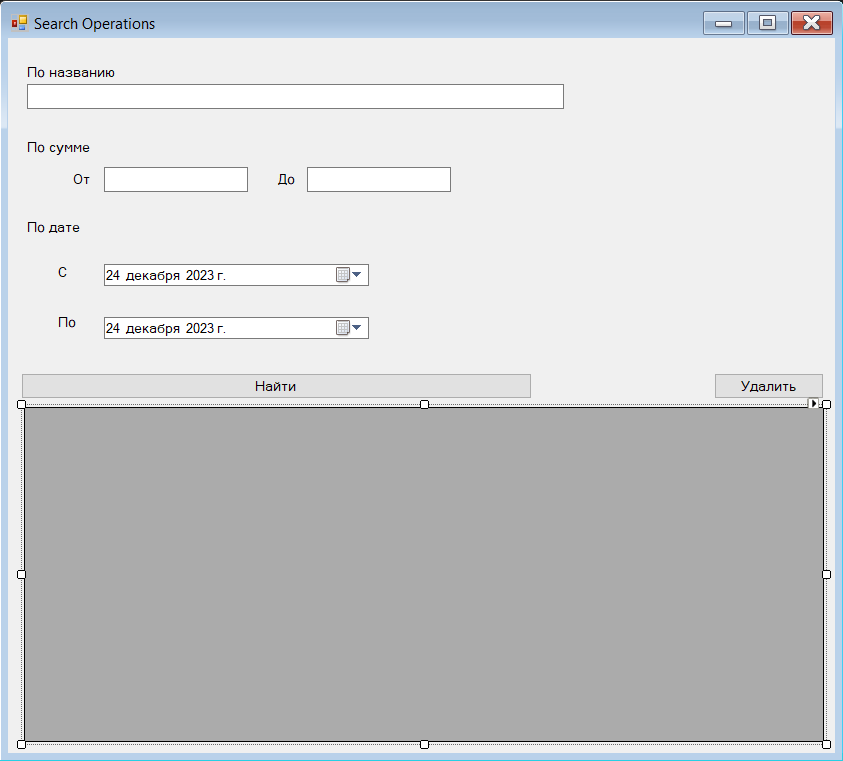
{

selectedDate = dateTimePicker1.Value;

}

}

}

****

using System;

using System.Data;

using System.Data.SQLite;

using System.Windows.Forms;

namespace ExpensesApp

{

public partial class SearchOperationsForm : Form

{

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

{

filterMax.Text = "0";

filterMax.Text = "100000";

DateTime old = DateTime.Now;

filterDateStart.Value = old.AddMonths(-1);

filterDateEnd.Value = DateTime.Now;

}

ExpenseTrackerDatabase db;

int deleteId;

public SearchOperationsForm(ExpenseTrackerDatabase \_db)

{

InitializeComponent();

LoadData();

this.db = \_db;

}

private void LoadData()

{

DataTable dataTable = new DataTable("Расходы");

dataTable.Columns.Add("Название", typeof(string));

dataTable.Columns.Add("Сумма", typeof(string));

dataTable.Columns.Add("Категория", typeof(string));

dataTable.Columns.Add("Дата", typeof(string));

dataTable.Columns.Add("Комментарий", typeof(string));

dataGridView1.DataSource = dataTable;

}

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

{

double max = 0;

double min = 0;

string name = filterName.Text;

DateTime startDate = filterDateStart.Value;

DateTime endDate = filterDateEnd.Value;

if (!string.IsNullOrWhiteSpace(filterMax.Text))

{

max = Convert.ToDouble(filterMax.Text);

}

if (!string.IsNullOrWhiteSpace(filterMin.Text))

{

min = Convert.ToDouble(filterMin.Text);

}

dataGridView1.DataSource = db.GetDataGridWithFilters(name, min, max, startDate, endDate);

}

static string connectionString = "Data Source=ExpensesDatabase.db;Version=3;";

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

{

using (SQLiteConnection connection = new SQLiteConnection(connectionString))

{

connection.Open();

string deleteProductQuery = "DELETE FROM Expenses WHERE Id = @ProductId";

using (SQLiteCommand command = new SQLiteCommand(deleteProductQuery, connection))

{

command.Parameters.AddWithValue("@ProductId", deleteId);

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Удалено.");

}

else

{

MessageBox.Show("Не найдено.");

}

}

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (dataGridView1.Columns[e.ColumnIndex].Name == "Название")

{

deleteId = 0;

bool FirstValue = true;

foreach (DataGridViewCell cell in dataGridView1.SelectedCells)

{

if (!FirstValue)

{

deleteId = 0;

}

try

{

deleteId = Convert.ToInt32(cell.Value.ToString());

}

catch { }

Console.WriteLine(deleteId);

FirstValue = false;

}

}

}

}

}