

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | Информатика и вычислительная техника | | |
| Направление подготовки | | 09.03.03 | Прикладная информатика | | |
| Направленность (профиль) | |  | Прикладная информатика в химии | | |
| Форма обучения | |  | очная | | |
|  | |  |  | | |
| Факультет | |  | Информационных технологий и управления | | |
| Кафедра | |  | Систем автоматизированного проектирования и управления | | |
| Учебная дисциплина | |  | Разработка ПС | | |
| Курс | I | | | Группа | 415 |

Отчёт по лабораторной работе № 3

Вариант № 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Исполнитель: |  |  |  |  |
| обучающийся группы 415 |  |  |  | Бердиев Эдуард Магометович |
|  |  | (дата, подпись) |  |  |
|  |  |  |  |  |
| Проверили: |  |  |  | Федин Алексей Константинович |
|  |  | (дата, подпись) |  | Корниенко Иван Григорьевич |

Содержание

[1 Постановка задачи 3](#_Toc99067513)

[2 Исходные данные 3](#_Toc99067514)

[3 Особые ситуации 3](#_Toc99067515)

[4 Математические методы и алгоритмы решения задач 3](#_Toc99067516)

[5 Форматы представления данных 4](#_Toc99067517)

[6 Структура программы 4](#_Toc99067518)

[7 Блок-схемы алгоритмов 7](#_Toc99067519)

[8 Результаты работы программы 9](#_Toc99067522)

[9 Описание хода выполнения контрольной работы 14](#_Toc99067523)

[10 Исходный текст программы 14](#_Toc99067524)

## 1 Постановка задачи

Необходимо написать приложение с использованием технологии WinForms реализующие Хранения списка банковских счетов. Программа должна позволять добавлять новые сущности с использованием интерфейса и редактировать существующие. Сущности, добавленные в программу, должны сохраняться между запусками приложения. Для хранения данных необходимо использовать СУБД SqLite. Необходимо предусмотреть возможность сохранения списка существующих сущностей в файл.

## 2 Исходные данные

Функция Верзьера Аньези.

## 3 Особые ситуации

Необходимо рассмотреть следующие особые ситуации:

– Знаменатель не может быть равен нулю.

**4 Математические методы и алгоритмы решения задач**

Согласно постановке задачи для составления программы, будет

использован алгоритм, реализующий:

-Рисовка графика по точкам с помощью формулы

## 5 Форматы представления данных

Таблица 1- Ивенты, используемые в программе

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Описание |
| dataHolders | List <DataHolder<double>> | Границы, шаг и коэфицент |
| graphDatas | List <GraphData> | Кординаты точек |

Данные могут быть предоставлены двумя видами, с помощью консольного ввода и в виде файла с разрешением “.txt”.

## 6 Структура программы

Для оптимизации работы с кодом было принято решение разделить программу на модули.

|  |  |
| --- | --- |
| Имя | Описание |
| Program.cs | Основной модуль |
| Calculator.cs | Расчёт кординат точек |
| DataHolder.cs | Хранение данных |
| IView.cs | Интерфейс View |
| FileManager.cs | Управление файлами |
| GraphData.cs | Данные о графике |
| AppForm.cs | Интерфейс |
| MessagesService.cs | Вывод сообщений |
| Presenter.cs | Управление |
| Validator.cs | Проверка данных |

Таблица 2 – Таблица модулей, составляющих программу

Каждый модуль разделён на функции, выполняющие определённые задачи.

## 7 Блок-схемы алгоритмов

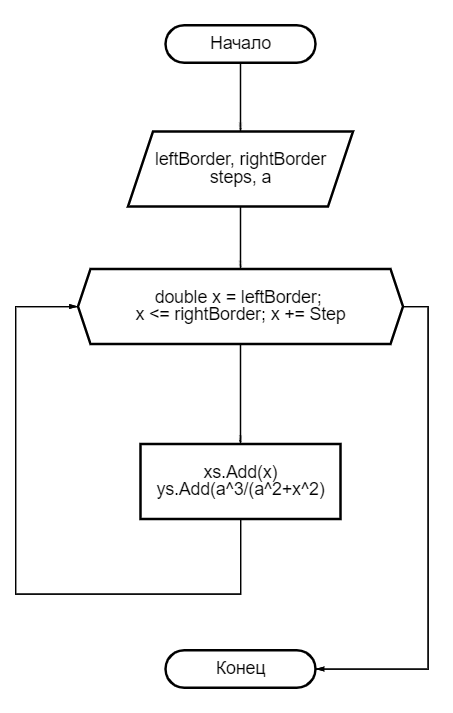


Рисунок 1 - Блок-схема алгоритма

## 8 Результаты работы программы

На экранных копиях работы программы показаны как корректный ввод данных, так и не корректный.

Экранные копии работы программы:

Рисунок 2-Экранная копия результатов работы программы

Рисунок 3-Экранная копия результатов работы программы

## 9 Описание хода выполнения контрольной работы

1) В ходе выполнения контрольной работы было создано решение в среде разработки Microsoft Visual Studio. В решении был создан проект.

2) При разработке программы была обнаружена проблема считывания с консоли данных только одного типа. После прочтения дополнительного материала проблемы были устранены.

3) Возникли трудности в создании модульных тестов, но после продолжительных усилий было найдено решение проблемы.

## 10 Исходный текст программы

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace FourthKR

{

internal class CalculateCheck

{

MyDataBase dataBase = new MyDataBase();

Form2 form2 = new Form2();

public static int CheckId(TextBox textid, DataGridView dgw)

{

int idcheck;

if (!Int32.TryParse(textid.Text, out idcheck))

{

MessageBox.Show("Id must be a number!", "BRUUH!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

for (int i = 0; i < dgw.RowCount; i++)

{

if (idcheck == Int32.Parse(dgw.Rows[i].Cells[0].Value.ToString()))

{

MessageBox.Show("Error...This id exists yet. Generate free id...", "BRUUH!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

idcheck = dgw.RowCount;

return idcheck;

}

}

return idcheck;

}

private static void ReadSingleRow(DataGridView dgw, IDataRecord record)

{

dgw.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetString(2), record.GetString(3), record.GetString(4), record.GetString(5), record.GetString(6),

record.GetString(7), record.GetString(8), record.GetString(9));

}

public static void CreateColums(DataGridView dgw)

{

dgw.Columns.Add("idclient", "Id");

dgw.Columns.Add("surname", "Surname");

dgw.Columns.Add("nameclient", "Name");

dgw.Columns.Add("patronymic", "Patronymic");

dgw.Columns.Add("age", "Age");

dgw.Columns.Add("number", "Number");

dgw.Columns.Add("adress", "Adress");

dgw.Columns.Add("numberpassport", "NumberOfPassport");

dgw.Columns.Add("numbercheck", "NumberOfCheck");

dgw.Columns.Add("balance", "Balance");

}

public static void RefreshDataGrid(DataGridView dgw,MyDataBase dataBase)

{

dgw.Rows.Clear();

string query = "select \* from client";

SqlCommand command = new SqlCommand(query, dataBase.GetConnection());

dataBase.OpenConnection();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

ReadSingleRow(dgw, reader);

}

reader.Close();

}

public static void AddCheck(TextBox id, TextBox textsurname, TextBox textname, TextBox textpatronymic, TextBox textage,

TextBox textnumber, TextBox textadress, TextBox textpassport, TextBox textnumcheck, TextBox textbalance, MyDataBase dataBase,DataGridView dgw)

{

int idcheck = CalculateCheck.CheckId(id, dgw);

var surname = textsurname.Text;

var name = textname.Text;

var patronymic = textpatronymic.Text;

var age = textage.Text;

var number = textnumber.Text;

var adress = textadress.Text;

var numofpassport = textpassport.Text;

var numofcheck = textnumcheck.Text;

var balance = textbalance.Text;

var addQuery = $"INSERT INTO client(idclient, surname, nameclient, patronymic, age, number, adress, numberpassport, numbercheck, balance) VALUES (@IdClient,@Surname,@NameClient,@Patronymic,@Age,@Number,@Adress,@NumPassport,@NumCheck,@Balance)";

var command = new SqlCommand(addQuery, dataBase.GetConnection());

command.Parameters.Add("@IdClient", SqlDbType.Int);

command.Parameters["@IdClient"].Value = idcheck;

command.Parameters.Add("@Surname", SqlDbType.VarChar);

command.Parameters["@Surname"].Value = surname;

command.Parameters.Add("@NameClient", SqlDbType.VarChar);

command.Parameters["@NameClient"].Value = name;

command.Parameters.Add("@Patronymic", SqlDbType.VarChar);

command.Parameters["@Patronymic"].Value = patronymic;

command.Parameters.Add("@Age", SqlDbType.VarChar);

command.Parameters["@Age"].Value = age;

command.Parameters.Add("@Number", SqlDbType.VarChar);

command.Parameters["@Number"].Value = number;

command.Parameters.Add("@Adress", SqlDbType.VarChar);

command.Parameters["@Adress"].Value = adress;

command.Parameters.Add("@NumPassport", SqlDbType.VarChar);

command.Parameters["@NumPassport"].Value = numofpassport;

command.Parameters.Add("@NumCheck", SqlDbType.VarChar);

command.Parameters["@NumCheck"].Value = numofcheck;

command.Parameters.Add("@Balance", SqlDbType.VarChar);

command.Parameters["@Balance"].Value = balance;

command.ExecuteNonQuery();

MessageBox.Show("New check successfully saved", "YEEI!", MessageBoxButtons.OK, MessageBoxIcon.Information);

RefreshDataGrid(dgw, dataBase);

}

public static void EditCheck(TextBox id, TextBox textsurname, TextBox textname, TextBox textpatronymic, TextBox textage,

TextBox textnumber, TextBox textadress, TextBox textpassport, TextBox textnumcheck, TextBox textbalance, MyDataBase dataBase, DataGridView dgw)

{

dataBase.OpenConnection();

var selectedRowIndex = dgw.CurrentCell.RowIndex;

int idcheck = Int32.Parse(id.Text);

var surname = textsurname.Text;

var name = textname.Text;

var patronymic = textpatronymic.Text;

var age = textage.Text;

var number = textnumber.Text;

var adress = textadress.Text;

var numofpassport = textpassport.Text;

var numofcheck = textnumcheck.Text;

var balance = textbalance.Text;

dgw.Rows[selectedRowIndex].SetValues(idcheck, surname, name, patronymic, age, number, adress, numofpassport, numofcheck, balance);

var editQuery = $"UPDATE client SET idclient = @IdClient,surname = @Surname, nameclient = @NameClient, patronymic = @Patronymic, age = @Age, number = @Number, adress = @Adress, numberpassport = @NumPassport, numbercheck = @NumCheck, balance = @Balance WHERE idclient = @IDClient";

var command = new SqlCommand(editQuery, dataBase.GetConnection());

command.Parameters.Add("@IdClient", SqlDbType.Int);

command.Parameters["@IdClient"].Value = idcheck;

command.Parameters.Add("@Surname", SqlDbType.VarChar);

command.Parameters["@Surname"].Value = surname;

command.Parameters.Add("@NameClient", SqlDbType.VarChar);

command.Parameters["@NameClient"].Value = name;

command.Parameters.Add("@Patronymic", SqlDbType.VarChar);

command.Parameters["@Patronymic"].Value = patronymic;

command.Parameters.Add("@Age", SqlDbType.VarChar);

command.Parameters["@Age"].Value = age;

command.Parameters.Add("@Number", SqlDbType.VarChar);

command.Parameters["@Number"].Value = number;

command.Parameters.Add("@Adress", SqlDbType.VarChar);

command.Parameters["@Adress"].Value = adress;

command.Parameters.Add("@NumPassport", SqlDbType.VarChar);

command.Parameters["@NumPassport"].Value = numofpassport;

command.Parameters.Add("@NumCheck", SqlDbType.VarChar);

command.Parameters["@NumCheck"].Value = numofcheck;

command.Parameters.Add("@Balance", SqlDbType.VarChar);

command.Parameters["@Balance"].Value = balance;

command.ExecuteNonQuery();

MessageBox.Show("Check succsessfully edited!", "Cool!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

public static void DeleteCheck (MyDataBase dataBase, DataGridView dgw)

{

int index = dgw.CurrentCell.RowIndex;

dgw.Rows[index].Visible = false;

if (dgw.Rows[index].Cells[0].Value.ToString() == string.Empty)

{

dgw.Rows[index].Cells[9].Value = DataRowState.Deleted;

}

dataBase.OpenConnection();

var id = Convert.ToInt32(dgw.Rows[index].Cells[0].Value);

var deleteQuery = $"DELETE FROM client WHERE idclient = @IdClient";

var command = new SqlCommand(deleteQuery, dataBase.GetConnection());

command.Parameters.Add("@IdClient", SqlDbType.Int);

command.Parameters["@IdClient"].Value = id;

command.ExecuteNonQuery();

}

public static void SearchByNumCheck(MyDataBase dataBase,DataGridView dgw,string value)

{

var numCheck = value;

var searchQuery = $"SELECT \* FROM client WHERE numbercheck = '{numCheck}'";

var command = new SqlCommand(searchQuery, dataBase.GetConnection());

SqlDataReader reader = command.ExecuteReader();

dgw.Rows.Clear();

while (reader.Read())

{

ReadSingleRow(dgw, reader);

}

reader.Close();

}

public static void SearchByFio(MyDataBase dataBase, DataGridView dgw, string value)

{

var numCheck = value;

var searchQuery = $"SELECT \* FROM client WHERE numbercheck = '{numCheck}'";

var command = new SqlCommand(searchQuery, dataBase.GetConnection());

SqlDataReader reader = command.ExecuteReader();

dgw.Rows.Clear();

while (reader.Read())

{

ReadSingleRow(dgw, reader);

}

reader.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

namespace FourthKR

{

internal class MyDataBase

{

SqlConnection sqlConnection = new SqlConnection(@"Data Source=DESKTOP-FKG5H2F;Initial Catalog=BankDB;Integrated Security=True");

public void OpenConnection()

{

if (sqlConnection.State == System.Data.ConnectionState.Closed)

{

sqlConnection.Open();

}

}

public void CloseConnection()

{

if (sqlConnection.State == System.Data.ConnectionState.Open)

{

sqlConnection.Close();

}

}

public SqlConnection GetConnection()

{

return sqlConnection;

}

}

}

using FourthKR.Properties;

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 FourthKR

{

public partial class Form1 : Form

{

MyDataBase dataBase = new MyDataBase();

public string action;

public string value;

public Form1()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen;

}

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

{

if (Settings.Default.Grettings == true)

{

MessageBox.Show("Berdiev Eduard 415 gr. KR-4 V-1, База данных с банковскими счетами", "HELLO", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

log\_in\_load(sender, e);

}

private void log\_in\_load(object sender, EventArgs e)

{

textBox1.PasswordChar = '\*';

pictureBox1.Visible = false;

textBox1.MaxLength = 48;

textBox2.MaxLength = 48;

}

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

{

label1.ForeColor = Color.Blue;

}

private void label2\_Click(object sender, EventArgs e)

{

label2.ForeColor = Color.Yellow;

}

private void label3\_Click(object sender, EventArgs e)

{

label3.ForeColor = Color.Pink;

}

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

{

var userLogin = textBox2.Text;

var userPassword = textBox1.Text;

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string firstquery = $"SELECT id\_user, login\_user, password\_user FROM register WHERE login\_user = '{userLogin}' and password\_user = '{userPassword}'";

SqlCommand command = new SqlCommand(firstquery, dataBase.GetConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

if(table.Rows.Count == 1)

{

MessageBox.Show("Successfully logged in!","Cool!",MessageBoxButtons.OK,MessageBoxIcon.Information);

Form2 form2 = new Form2();

this.Hide();

form2.ShowDialog();

}

else

{

MessageBox.Show("No such user exists!", "Bruuh...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

textBox1.PasswordChar = '\*';

pictureBox1.Update();

pictureBox1.Visible = false;

pictureBox2.Visible = true;

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

textBox1.PasswordChar = '\0';

pictureBox2.Visible = false;

pictureBox1.Visible = true;

}

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

{

button1.BackColor = Color.LightBlue;

Cursor = Cursors.Hand;

}

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

{

button1.BackColor = Color.White;

Cursor = Cursors.Default;

}

private void toolStripMenuItem1\_CheckStateChanged(object sender, EventArgs e)

{

if (toolStripMenuItem1.Checked)

Settings.Default.Grettings = true;

else

Settings.Default.Grettings = false;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Media;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace FourthKR

{

public partial class Form2 : Form

{

SearchForm searchForm = new SearchForm();

MyDataBase dataBase = new MyDataBase();

int selectedRow;

public Form2()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen;

}

private void MouseMoveCase(Label label)

{

label.ForeColor = Color.DarkOrchid;

label.Font = new Font(label.Font, FontStyle.Underline);

Cursor = Cursors.Hand;

}

private void MouseLeaveCase(Label label)

{

label.ForeColor = Color.Black;

label.Font = new Font(label4.Font, FontStyle.Regular);

Cursor = Cursors.Default;

}

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

{

textnumber.Text = "+";

CalculateCheck.CreateColums(dataGridView1);

CalculateCheck.RefreshDataGrid(dataGridView1, dataBase);

}

private void label3\_Click(object sender, EventArgs e)

{

CalculateCheck.AddCheck(textid,textsurname,textname, textpatronymic, textage, textnumber, textadress, textpassport, textnumcheck,textbalance,dataBase,dataGridView1);

}

private void label3\_MouseMove(object sender, MouseEventArgs e)

{

MouseMoveCase(label3);

}

private void label3\_MouseLeave(object sender, EventArgs e)

{

MouseLeaveCase(label3);

}

private void label4\_Click(object sender, EventArgs e)

{

CalculateCheck.EditCheck(textid,textsurname,textname,textpatronymic,textage,textnumber,textadress,textpassport,textnumcheck,textbalance,dataBase,dataGridView1);

}

private void label4\_MouseMove(object sender, MouseEventArgs e)

{

MouseMoveCase(label4);

}

private void label4\_MouseLeave(object sender, EventArgs e)

{

MouseLeaveCase(label4);

}

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

{

CalculateCheck.RefreshDataGrid(dataGridView1,dataBase);

SoundPlayer sound = new SoundPlayer(@"D:\1\Downloads\sound.wav");

sound.Play();

MessageBox.Show("Well! Data updated successfully!!", "Update", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

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

{

selectedRow = e.RowIndex;

if(selectedRow >=0 )

{

DataGridViewRow row = dataGridView1.Rows[selectedRow];

textid.Text = row.Cells[0].Value.ToString();

textsurname.Text = row.Cells[1].Value.ToString();

textname.Text = row.Cells[2].Value.ToString();

textpatronymic.Text = row.Cells[3].Value.ToString();

textage.Text = row.Cells[4].Value.ToString();

textnumber.Text = row.Cells[5].Value.ToString();

textadress.Text = row.Cells[6].Value.ToString();

textpassport.Text = row.Cells[7].Value.ToString();

textnumcheck.Text = row.Cells[8].Value.ToString();

textbalance.Text = row.Cells[9].Value.ToString();

}

}

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

{

var select = MessageBox.Show("Are you really want to delete this check?","Are you sure?",MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if(select == DialogResult.No)

{

return;

}

CalculateCheck.DeleteCheck(dataBase, dataGridView1);

}

private void labelDeleteCheck\_MouseMove(object sender, MouseEventArgs e)

{

MouseMoveCase(labelDeleteCheck);

}

private void labelDeleteCheck\_MouseLeave(object sender, EventArgs e)

{

MouseLeaveCase(labelDeleteCheck);

}

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

{

searchForm.ShowDialog();

var searchValue = searchForm.textBoxSearch.Text;

CalculateCheck.SearchByNumCheck(dataBase, dataGridView1, searchValue);

}

private void labelsearchbynumcheck\_MouseEnter(object sender, EventArgs e)

{

MouseMoveCase(labelsearchbynumcheck);

}

private void labelsearchbynumcheck\_MouseLeave(object sender, EventArgs e)

{

MouseLeaveCase(labelsearchbynumcheck);

}

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

{

}

}

}