Додаток А

Вхідні дані. Склад контрольного прикладу

Таблиця А.1 – Вікно в якому відображено інформацію про здання тестування

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ФИО | Номер телефона | Сданый тест | Результаты | Оценка | Дата и время сдачи |
| Круглый С.И. | +380994322735 | Чат-Онлайн | 65 из 70 | 5 | 18.05.2016 16:26:31 |
| Алексеев Р.В. | +380994121412 | Охрана | 25 из 40 | 3 | 21.09.2016 15:30:12 |
| Усатый Т.С. | +380985894357 | HD | 69 из 80 | 4 | 01.10.2016 19:21:07 |
| Слипченко М.А. | +380504123654 | Тестирование ПО | 50 из 60 | 4 | 10.11.2016 11:10:45 |
| Малый В.С. | +380973625147 | Уборщик | 23 из 25 | 5 | 09.10.2016 08:30:25 |
| Назаренко С.В. | +380939878654 | Бизнес Центр | 38 из 50 | 3 | 11.11.2016 12:32:58 |
| Бойко А.С. | +380737845123 | Чат онлайн | 61 из 70 | 4 | 10.11.2016 17:27:36 |
| Петров В.К. | +380661425963 | Чат онлайн | 55 из 70 | 3 | 18.09.2016 11:12:24 |
| Иванов О.Л. | +3804125324 | Фриланс | 12 из 30 | 2 | 1.11.2016 09:14:47 |
| Сидоров Е.К. | +380922145952 | 3700 | 30 из 40 | 4 | 15.10:2016 18:28:14 |

Таблиця А.2 – Сховища та їх характеристики

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер сховища | Об’єм сховища,Т | Вентиляція | Авто | Потягом | Тип завантаження | Об’єм на сховищі, Т | Назва культури |
| 2 | 1000 | + | + | + | Автомобілем | 20 | Рапс |
| 1 | 140 | + | + | - | Транспортером | 74 | Просо |
| 5 | 20000 | + | + | + | Транспортером, Автомобілем | 1500 | Пшениця |
| 7 | 1500 | + | - | + | Автомобілем | 400 | Соняшник |
| 6 | 1850 | - | + | - | Транспортером | 251 | Кукурудза |
| 12 | 214 | + | + | - | Транспортером | 147 | Гречка |
| 13 | 180 | + | + | + | Транспортером | 18 | Ячмінь |

Таблиця А.3 – Завантажене зерно на елеватор

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Вага зерна, Т | Грязість,% | Вологість,% | Клейковина,%(може не бути) | Номер сховища | Дата прибуття | Назва культури |
| 10 | 5 | 14 | 0 | 2 | 13.09.2015 | Рапс |
| 150 | 10 | 11 | 10 | 5 | 19.09.2015 | Пшениця |
| 80 | 11 | 10 | 0 | 13 | 20.08.2015 | Ячмінь |
| 50 | 5 | 9 | 0 | 12 | 23.08.2015 | Гречка |
| 45 | 8 | 10 | 0 | 1 | 01.10.2015 | Просо |
| 120 | 12 | 13 | 0 | 7 | 05.09.2015 | Соняшник |
| 130 | 15 | 11 | 0 | 6 | 10.10.2015 | Кукурудза |

Таблиця А.4 – Вивантажене зерно з елеватора

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Вага зерна, Т | Грязість,% | Вологість,% | Клейковина,%(може не бути) | Номер сховища | Дата прибуття | Назва культури |
| 100 | 6 | 14 | 0 | 2 | 20.09.2015 | Рапс |
| 110 | 12 | 12 | 20 | 5 | 30.09.2015 | Пшениця |
| 75 | 11 | 11 | 0 | 13 | 25.08.2015 | Ячмінь |
| 35 | 6 | 8 | 0 | 12 | 10.09.2015 | Гречка |
| 50 | 9 | 10 | 0 | 1 | 35.10.2015 | Просо |
| 150 | 11 | 12 | 0 | 7 | 06.10.2015 | Соняшник |
| 10 | 14 | 10 | 0 | 6 | 13.11.2015 | Кукурудза |

Додаток Б

Структурна схема алгоритму програми

Обробка події натиснення кнопки «розпочати тестування» на формі «MainForm»

Початок

Перехід до проходження тесту

Кінець

Введення даних

коректно

Не коректно

Вибір тесту

Додаток В

Текст програми

Лістинг – текст програми.

//Лістинг головної форми програми

using System;

using System.Data.SqlClient;

using System.Text.RegularExpressions;

using System.Windows.Forms;

namespace TesterCourseProject

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

\_update();

}

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

{

Form frm = new LogOn(1);

frm.ShowDialog();

\_update();

}

public void \_update()

{

int Index = comboBox1.SelectedIndex;

comboBox1.Items.Clear();

int amount\_tables = 0;

{

SqlConnection conn = new SqlConnection(Conn.Default.connStr);

conn.Open();

String sql = "SELECT TABLE\_NAME FROM information\_schema.TABLES WHERE TABLE\_TYPE LIKE '%TABLE%'";

SqlCommand command = new SqlCommand(sql, conn);

SqlDataReader reader = command.ExecuteReader();

//щщас бы строку удалить)

while (reader.Read())

{

String table = reader.GetString(0);

if ((table != "People") && (table != "Admins"))

{

comboBox1.Items.Add(table);

amount\_tables += 1;

}

}

conn.Close();

conn.Dispose();

}//считываем количество таблиц в базе

int amount\_people=0;

{

SqlConnection conn = new SqlConnection(Conn.Default.connStr);

conn.Open();

String sql = "SELECT \* FROM People";

SqlCommand command = new SqlCommand(sql, conn);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

amount\_people += 1;

}

conn.Close();

conn.Dispose();

}//считываем количество таблиц в базе

if (comboBox1.Items.Count != 0)

{

if (Index == -1)

comboBox1.SelectedIndex = 0;

else

comboBox1.SelectedIndex = Index;

startTest.Enabled = true;

}

else

{

startTest.Enabled = false;

}//если у нас есть хоть 1 таблица с тестами то ставим ее в автовыбор и активируем кнопку,

//если тестов нет кнопка блокируется

toolStripStatusLabel1.Text = String.Format("В данный момент тесты были сданы:{0} раз", amount\_people);

toolStripStatusLabel2.Text = String.Format("Количество тестов в базе:{0}", amount\_tables);

}

private void timer1\_Tick(object sender, EventArgs e)

{

\_update();

}

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

{

bool check = true;

if(nameBox.Text.Length == 0)

{

MessageBox.Show("Введите вашу фамилию и инициалы!", "Не введено имя", MessageBoxButtons.OK, MessageBoxIcon.Error);

check = false;

}

if(groupBox.Text.Length == 0)

{

MessageBox.Show("Введите вашу группу!", "Не введена группа", MessageBoxButtons.OK, MessageBoxIcon.Error);

check = false;

}

if(check)

{

Regex reg = new Regex(@"^[А-Я][а-я]+ [А-Я].[А-Я].$"); //проверка фамилии + инициалов

Match mt = reg.Match(nameBox.Text);

if(!mt.Success)

{

MessageBox.Show("Фамилия и инициалы введенны в некорректном формате!\nВводить в формате:Фамилия И.И.", "Некорректный ввод", MessageBoxButtons.OK, MessageBoxIcon.Error);

check = false;

}

reg = new Regex(@"^\+38[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$");

mt = reg.Match(groupBox.Text);

if(!mt.Success)

{

MessageBox.Show("Номер телефона введен в некорректном формате!\nВводить в формате:+380994322735", "Некорректный ввод", MessageBoxButtons.OK, MessageBoxIcon.Error);

check = false;

}

//end checking name and group

if(check)

{

TestOpened frm = new TestOpened(this, comboBox1.Items[comboBox1.SelectedIndex].ToString(),nameBox.Text,groupBox.Text);

frm.ShowDialog();

nameBox.Text = "";

groupBox.Text = "";

}

}// end if check

}

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

{

Form frm = new LogOn(2);

frm.ShowDialog();

}

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

{

MessageBox.Show("Програмный продукт предназначен для сдачи тестирования новыми сотрудниками банка \n Для помощи обращайтесь по телефону +380994322735", "Помощь", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

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

{

MessageBox.Show("Круглый С.И.", "Разработчик", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

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

{

}

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

{

}

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

{

}

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

{

}

}

}

//Лістинг форми AddRow

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace TesterCourseProject

{

public partial class AddRow : Form

{

CreateTest frm = null;

public AddRow(CreateTest main)

{

InitializeComponent();

frm = main;

}

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

{

bool ok = true;

if(textBox1.Text == "")

{

ok = false;

MessageBox.Show("Не задан вопрос", "No Question", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if(textBox2.Text == "")

{

ok = false;

MessageBox.Show("Не задан 1-ый обязательный вариант", "No Variants", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (textBox3.Text == "")

{

ok = false;

MessageBox.Show("Не задан 2-ой обязательный вариант", "No Variants", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if(ok)

{

if((checkBox1.Checked==false) && (checkBox2.Checked==false) && (checkBox3.Checked == false) && (checkBox4.Checked == false) && (checkBox5.Checked == false) && (checkBox6.Checked == false))

{

ok = false;

MessageBox.Show("Не задан ответ(ы)!", "No Right Answers", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

if(ok)

{

frm.dataGridView1.Rows.Add(textBox1.Text, checkBox1.Checked, textBox2.Text, checkBox2.Checked, textBox3.Text, checkBox3.Checked, textBox4.Text, checkBox4.Checked, textBox5.Text, checkBox5.Checked, textBox6.Text, checkBox6.Checked, textBox7.Text);

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox6.Text = "";

textBox7.Text = "";

checkBox1.Checked = false;

checkBox2.Checked = false;

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (textBox4.Text == "")

{

checkBox3.Checked = false;

textBox5.Enabled = false;

checkBox4.Checked = false;

checkBox4.Enabled = false;

}

else

{

textBox5.Enabled = true;

checkBox4.Enabled = true;

}

if (textBox5.Text == "")

{

checkBox4.Checked = false;

textBox6.Enabled = false;

checkBox5.Checked = false;

checkBox5.Enabled = false;

}

else

{

textBox6.Enabled = true;

checkBox5.Enabled = true;

}

if (textBox6.Text == "")

{

checkBox5.Checked = false;

textBox7.Enabled = false;

checkBox6.Checked = false;

checkBox6.Enabled = false;

}

else

{

textBox7.Enabled = true;

checkBox6.Enabled = true;

}

if(textBox7.Text=="")

checkBox6.Checked = false;

}

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

{

}

}

}

//Лістинг форми CreateTest

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace TesterCourseProject

{

public partial class CreateTest : Form

{

public class Questions

{

public String Question;

public bool True1;

public String Var1;

public bool True2;

public String Var2;

public bool True3;

public String Var3;

public bool True4;

public String Var4;

public bool True5;

public String Var5;

public bool True6;

public String Var6;

}

public List<Questions> All = new List<Questions>();

public CreateTest()

{

InitializeComponent();

}

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

{

if (textBox1.Text.Length == 0)

MessageBox.Show("Нельзя задать пустое имя теста!", "Ошибка в названии", MessageBoxButtons.OK, MessageBoxIcon.Error);

else

{

if(dataGridView1.Rows.Count!=0)

{

SqlConnection conn = new SqlConnection(Conn.Default.connStr);

conn.Open();

SqlCommand cmdCreateTable = new SqlCommand(String.Format("CREATE TABLE [{0}] (Question NVARCHAR (MAX) NOT NULL, True1 BIT NOT NULL, Var1 NVARCHAR (MAX) NOT NULL, True2 BIT NOT NULL, Var2 NVARCHAR (MAX) NOT NULL, True3 BIT NULL, Var3 NVARCHAR (MAX) NULL, True4 BIT NULL, Var4 NVARCHAR (MAX) NULL, True5 BIT NULL, Var5 NVARCHAR (MAX) NULL, True6 BIT NULL, Var6 NVARCHAR (MAX) NULL);", textBox1.Text),conn);

//посылаем запрос

bool ok = true;

try

{

cmdCreateTable.ExecuteNonQuery();

}

catch (SqlException ex)

{

MessageBox.Show("Ошибка при создании таблицы: " + ex.Message);

ok = false;

}

if(ok)

{

InsertingQuestions(conn);

}//if ok

conn.Close();

conn.Dispose();

}//end if have rows

else

{

MessageBox.Show("Нельзя создать тест без вопросов!", "No Questions in test", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}//end button click

public void InsertingQuestions(SqlConnection conn)

{

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

Questions quest = new Questions();

quest.Question = dataGridView1.Rows[i].Cells[0].Value.ToString();

//all variants down

try

{

quest.Var1 = dataGridView1.Rows[i].Cells[2].Value.ToString();

}

catch

{

quest.Var1 = null;

}

//

try

{

quest.Var2 = dataGridView1.Rows[i].Cells[4].Value.ToString();

}

catch

{

quest.Var2 = null;

}

//

try

{

quest.Var3 = dataGridView1.Rows[i].Cells[6].Value.ToString();

}

catch

{

quest.Var3 = null;

}

//

try

{

quest.Var4 = dataGridView1.Rows[i].Cells[8].Value.ToString();

}

catch

{

quest.Var4 = null;

}

//

try

{

quest.Var5 = dataGridView1.Rows[i].Cells[10].Value.ToString();

}

catch

{

quest.Var5 = null;

}

//

try

{

quest.Var6 = dataGridView1.Rows[i].Cells[12].Value.ToString();

}

catch

{

quest.Var6 = null;

}

//all bool down

//1

try

{

quest.True1 = (Boolean)dataGridView1.Rows[i].Cells[1].Value;

}

catch

{

quest.True1 = false;

}

//2

try

{

quest.True2 = (Boolean)dataGridView1.Rows[i].Cells[3].Value;

}

catch

{

quest.True2 = false;

}

//3

try

{

quest.True3 = (Boolean)dataGridView1.Rows[i].Cells[5].Value;

}

catch

{

quest.True3 = false;

}

//4

try

{

quest.True4 = (Boolean)dataGridView1.Rows[i].Cells[7].Value;

}

catch

{

quest.True4 = false;

}

//5

try

{

quest.True5 = (Boolean)dataGridView1.Rows[i].Cells[9].Value;

}

catch

{

quest.True5 = false;

}

//6

try

{

quest.True6 = (Boolean)dataGridView1.Rows[i].Cells[11].Value;

}

catch

{

quest.True6 = false;

}

All.Add(quest);

}//end adding all questions

foreach (Questions quest in All)

{

SqlCommand cmdInsert = new SqlCommand(String.Format("INSERT INTO [{0}](Question, True1, Var1, True2, Var2, True3, Var3, True4, Var4, True5, Var5, True6, Var6) VALUES ('{1}', {2}, '{3}', {4}, '{5}', {6}, '{7}', {8}, '{9}', {10}, '{11}', {12}, '{13}');", textBox1.Text, quest.Question, Convert.ToInt32(quest.True1), quest.Var1, Convert.ToInt32(quest.True2), quest.Var2, Convert.ToInt32(quest.True3), quest.Var3, Convert.ToInt32(quest.True4), quest.Var4, Convert.ToInt32(quest.True5), quest.Var5, Convert.ToInt32(quest.True6), quest.Var6), conn);

try

{

cmdInsert.ExecuteNonQuery();

}

catch (SqlException ex)

{

MessageBox.Show("Oшибка, при выполнении запроса на добавление записи:" + ex.Message);

}

}

MessageBox.Show("Успешно выполнено!", "Done", MessageBoxButtons.OK, MessageBoxIcon.Information);

conn.Close();

conn.Dispose();

this.Close();

}

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

{

AddRow frm = new AddRow(this);

frm.ShowDialog();

}

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

{

}

private void pictureBox1\_Click\_1(object sender, EventArgs e)

{

}

private void pictureBox1\_Click\_2(object sender, EventArgs e)

{

}

}

}

//Лістинг Форми LogOn

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace TesterCourseProject

{

public partial class LogOn : Form

{

public int what;

public LogOn(int what)

{

InitializeComponent();

this.what = what;

}

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

{

bool find = false;

{

SqlConnection conn = new SqlConnection(Conn.Default.connStr);

conn.Open();

String sql = "SELECT \* FROM Admins";

SqlCommand command = new SqlCommand(sql, conn);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

if((reader.GetValue(1).ToString()==textBox1.Text) && (reader.GetValue(2).ToString()==textBox2.Text))

{

find = true;

break;

}

}

conn.Close();

conn.Dispose();

}//проверяем на наличие в базе

if(find)

{

if (what == 1)

{

Form frm = new CreateTest();

this.Hide();

frm.ShowDialog();

this.Close();

}//открываем создание теста

else

{

Form frm = new ViewPeople();

this.Hide();

frm.ShowDialog();

this.Close();

}//открываем сдавших тест

}

else

{

MessageBox.Show("Неправильный логин или пароль!", "Oops", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

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

{

}

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

{

}

}

}

//Лістинг Форми Program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace TesterCourseProject

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}

}

//Лістинг форми TestOpened

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace TesterCourseProject

{

public partial class TestOpened : Form

{

MainForm frm = null;

String testName = null;

public int right\_count = 0;

public int all\_count = 0;

public String name;

public String group;

public class Questions

{

public String Question;

public bool True1;

public String Var1 = null;

public bool True2;

public String Var2 = null;

public bool True3;

public String Var3 = null;

public bool True4;

public String Var4 = null;

public bool True5;

public String Var5 = null;

public bool True6;

public String Var6 = null;

}

public List<Questions> All = new List<Questions>();

public TestOpened(MainForm main, String TestName, String name, String group)

{

InitializeComponent();

frm = main;

testName = TestName;

this.name = name;

this.group = group;

InitializeTest();

}

public void InitializeTest()

{

SqlConnection conn = new SqlConnection(Conn.Default.connStr);

try

{

//пробуем подключится

conn.Open();

}

catch (SqlException se)

{

MessageBox.Show("Ошибка подключения:{0}", se.Message);

}

SqlCommand cmd = new SqlCommand(String.Format("SELECT \* FROM [{0}]",testName), conn);

SqlDataReader dr = cmd.ExecuteReader(CommandBehavior.CloseConnection);

while (dr.Read())

{

Questions quest = new Questions();

quest.Question = dr.GetValue(0).ToString();

quest.True1 = Convert.ToBoolean(dr.GetValue(1));

quest.Var1 = dr.GetValue(2).ToString();

quest.True2 = Convert.ToBoolean(dr.GetValue(3));

quest.Var2 = dr.GetValue(4).ToString();

quest.True3 = Convert.ToBoolean(dr.GetValue(5));

quest.Var3 = dr.GetValue(6).ToString();

quest.True4 = Convert.ToBoolean(dr.GetValue(7));

quest.Var4 = dr.GetValue(8).ToString();

quest.True5 = Convert.ToBoolean(dr.GetValue(9));

quest.Var5 = dr.GetValue(10).ToString();

quest.True6 = Convert.ToBoolean(dr.GetValue(11));

quest.Var6 = dr.GetValue(12).ToString();

All.Add(quest);

}

conn.Close();

conn.Dispose();

//start creating test

{

for(int i=0;i<All.Count;i++)

{

TabPage tp = new TabPage();

tabControl1.TabPages.Add(tp);

Label question = new Label();

question.AutoSize = true;

question.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

question.MinimumSize = new System.Drawing.Size(tabControl1.Width - 20, 10);

question.MaximumSize = new System.Drawing.Size(tabControl1.Width - 20, 200);

question.Text = All[i].Question;

tabControl1.TabPages[i].Controls.Add(question);

bool stop = false;

int trueCount = 0;

if ((All[i].Var3 != "") && (All[i].Var4 != "") && (All[i].Var5 != "") && (All[i].Var6 != ""))

{

stop = true;

if (All[i].True1 == true)

trueCount += 1;

if (All[i].True2 == true)

trueCount += 1;

if (All[i].True3 == true)

trueCount += 1;

if (All[i].True4 == true)

trueCount += 1;

if (All[i].True5 == true)

trueCount += 1;

if (All[i].True6 == true)

trueCount += 1;

if(trueCount == 1)

{

RadioButton rd1 = new RadioButton();

RadioButton rd2 = new RadioButton();

RadioButton rd3 = new RadioButton();

RadioButton rd4 = new RadioButton();

RadioButton rd5 = new RadioButton();

RadioButton rd6 = new RadioButton();

rd1.Text = All[i].Var1;

rd2.Text = All[i].Var2;

rd3.Text = All[i].Var3;

rd4.Text = All[i].Var4;

rd5.Text = All[i].Var5;

rd6.Text = All[i].Var6;

rd1.AutoSize = true;

rd2.AutoSize = true;

rd3.AutoSize = true;

rd4.AutoSize = true;

rd5.AutoSize = true;

rd6.AutoSize = true;

rd1.Top = question.Height+10;

rd2.Top = question.Height + rd1.Height + 20;

rd3.Top = question.Height + rd1.Height + rd2.Height + 30;

rd4.Top = question.Height + rd1.Height + rd2.Height + rd3.Height + 40;

rd5.Top = question.Height + rd1.Height + rd2.Height + rd3.Height + rd4.Height + 50;

rd6.Top = question.Height + rd1.Height + rd2.Height + rd3.Height + rd4.Height + rd5.Height + 60;

tabControl1.TabPages[i].Controls.Add(rd1);

tabControl1.TabPages[i].Controls.Add(rd2);

tabControl1.TabPages[i].Controls.Add(rd3);

tabControl1.TabPages[i].Controls.Add(rd4);

tabControl1.TabPages[i].Controls.Add(rd5);

tabControl1.TabPages[i].Controls.Add(rd6);

}

else

{

CheckBox cb1 = new CheckBox();

CheckBox cb2 = new CheckBox();

CheckBox cb3 = new CheckBox();

CheckBox cb4 = new CheckBox();

CheckBox cb5 = new CheckBox();

CheckBox cb6 = new CheckBox();

cb1.Text = All[i].Var1;

cb2.Text = All[i].Var2;

cb3.Text = All[i].Var3;

cb4.Text = All[i].Var4;

cb5.Text = All[i].Var5;

cb6.Text = All[i].Var6;

cb1.AutoSize = true;

cb2.AutoSize = true;

cb3.AutoSize = true;

cb4.AutoSize = true;

cb5.AutoSize = true;

cb6.AutoSize = true;

cb1.Top = question.Height + 10;

cb2.Top = question.Height + cb1.Height + 20;

cb3.Top = question.Height + cb1.Height + cb2.Height + 30;

cb4.Top = question.Height + cb1.Height + cb2.Height + cb3.Height + 40;

cb5.Top = question.Height + cb1.Height + cb2.Height + cb3.Height + cb4.Height + 50;

cb6.Top = question.Height + cb1.Height + cb2.Height + cb3.Height + cb4.Height + cb5.Height + 60;

tabControl1.TabPages[i].Controls.Add(cb1);

tabControl1.TabPages[i].Controls.Add(cb2);

tabControl1.TabPages[i].Controls.Add(cb3);

tabControl1.TabPages[i].Controls.Add(cb4);

tabControl1.TabPages[i].Controls.Add(cb5);

tabControl1.TabPages[i].Controls.Add(cb6);

}

}//end if all question have

if ((All[i].Var3 != "") && (All[i].Var4 != "") && (All[i].Var5 != "") && (!stop))

{

stop = true;

if (All[i].True1 == true)

trueCount += 1;

if (All[i].True2 == true)

trueCount += 1;

if (All[i].True3 == true)

trueCount += 1;

if (All[i].True4 == true)

trueCount += 1;

if (All[i].True5 == true)

trueCount += 1;

if (trueCount == 1)

{

RadioButton rd1 = new RadioButton();

RadioButton rd2 = new RadioButton();

RadioButton rd3 = new RadioButton();

RadioButton rd4 = new RadioButton();

RadioButton rd5 = new RadioButton();

rd1.Text = All[i].Var1;

rd2.Text = All[i].Var2;

rd3.Text = All[i].Var3;

rd4.Text = All[i].Var4;

rd5.Text = All[i].Var5;

rd1.AutoSize = true;

rd2.AutoSize = true;

rd3.AutoSize = true;

rd4.AutoSize = true;

rd5.AutoSize = true;

rd1.Top = question.Height + 10;

rd2.Top = question.Height + rd1.Height + 20;

rd3.Top = question.Height + rd1.Height + rd2.Height + 30;

rd4.Top = question.Height + rd1.Height + rd2.Height + rd3.Height + 40;

rd5.Top = question.Height + rd1.Height + rd2.Height + rd3.Height + rd4.Height + 50;

tabControl1.TabPages[i].Controls.Add(rd1);

tabControl1.TabPages[i].Controls.Add(rd2);

tabControl1.TabPages[i].Controls.Add(rd3);

tabControl1.TabPages[i].Controls.Add(rd4);

tabControl1.TabPages[i].Controls.Add(rd5);

}

else

{

CheckBox cb1 = new CheckBox();

CheckBox cb2 = new CheckBox();

CheckBox cb3 = new CheckBox();

CheckBox cb4 = new CheckBox();

CheckBox cb5 = new CheckBox();

cb1.Text = All[i].Var1;

cb2.Text = All[i].Var2;

cb3.Text = All[i].Var3;

cb4.Text = All[i].Var4;

cb5.Text = All[i].Var5;

cb1.AutoSize = true;

cb2.AutoSize = true;

cb3.AutoSize = true;

cb4.AutoSize = true;

cb5.AutoSize = true;

cb1.Top = question.Height + 10;

cb2.Top = question.Height + cb1.Height + 20;

cb3.Top = question.Height + cb1.Height + cb2.Height + 30;

cb4.Top = question.Height + cb1.Height + cb2.Height + cb3.Height + 40;

cb5.Top = question.Height + cb1.Height + cb2.Height + cb3.Height + cb4.Height + 50;

tabControl1.TabPages[i].Controls.Add(cb1);

tabControl1.TabPages[i].Controls.Add(cb2);

tabControl1.TabPages[i].Controls.Add(cb3);

tabControl1.TabPages[i].Controls.Add(cb4);

tabControl1.TabPages[i].Controls.Add(cb5);

}

}//end if 5 questions

if ((All[i].Var3 != "") && (All[i].Var4 != "") && (!stop))

{

stop = true;

if (All[i].True1 == true)

trueCount += 1;

if (All[i].True2 == true)

trueCount += 1;

if (All[i].True3 == true)

trueCount += 1;

if (All[i].True4 == true)

trueCount += 1;

if (trueCount == 1)

{

RadioButton rd1 = new RadioButton();

RadioButton rd2 = new RadioButton();

RadioButton rd3 = new RadioButton();

RadioButton rd4 = new RadioButton();

rd1.Text = All[i].Var1;

rd2.Text = All[i].Var2;

rd3.Text = All[i].Var3;

rd4.Text = All[i].Var4;

rd1.AutoSize = true;

rd2.AutoSize = true;

rd3.AutoSize = true;

rd4.AutoSize = true;

rd1.Top = question.Height + 10;

rd2.Top = question.Height + rd1.Height + 20;

rd3.Top = question.Height + rd1.Height + rd2.Height + 30;

rd4.Top = question.Height + rd1.Height + rd2.Height + rd3.Height + 40;

tabControl1.TabPages[i].Controls.Add(rd1);

tabControl1.TabPages[i].Controls.Add(rd2);

tabControl1.TabPages[i].Controls.Add(rd3);

tabControl1.TabPages[i].Controls.Add(rd4);

}

else

{

CheckBox cb1 = new CheckBox();

CheckBox cb2 = new CheckBox();

CheckBox cb3 = new CheckBox();

CheckBox cb4 = new CheckBox();

cb1.Text = All[i].Var1;

cb2.Text = All[i].Var2;

cb3.Text = All[i].Var3;

cb4.Text = All[i].Var4;

cb1.AutoSize = true;

cb2.AutoSize = true;

cb3.AutoSize = true;

cb4.AutoSize = true;

cb1.Top = question.Height + 10;

cb2.Top = question.Height + cb1.Height + 20;

cb3.Top = question.Height + cb1.Height + cb2.Height + 30;

cb4.Top = question.Height + cb1.Height + cb2.Height + cb3.Height + 40;

tabControl1.TabPages[i].Controls.Add(cb1);

tabControl1.TabPages[i].Controls.Add(cb2);

tabControl1.TabPages[i].Controls.Add(cb3);

tabControl1.TabPages[i].Controls.Add(cb4);

}

}//end if 4 questions

if ((All[i].Var3 != "") && (!stop))

{

stop = true;

if (All[i].True1 == true)

trueCount += 1;

if (All[i].True2 == true)

trueCount += 1;

if (All[i].True3 == true)

trueCount += 1;

if (trueCount == 1)

{

RadioButton rd1 = new RadioButton();

RadioButton rd2 = new RadioButton();

RadioButton rd3 = new RadioButton();

rd1.Text = All[i].Var1;

rd2.Text = All[i].Var2;

rd3.Text = All[i].Var3;

rd1.AutoSize = true;

rd2.AutoSize = true;

rd3.AutoSize = true;

rd1.Top = question.Height + 10;

rd2.Top = question.Height + rd1.Height + 20;

rd3.Top = question.Height + rd1.Height + rd2.Height + 30;

tabControl1.TabPages[i].Controls.Add(rd1);

tabControl1.TabPages[i].Controls.Add(rd2);

tabControl1.TabPages[i].Controls.Add(rd3);

}

else

{

CheckBox cb1 = new CheckBox();

CheckBox cb2 = new CheckBox();

CheckBox cb3 = new CheckBox();

cb1.Text = All[i].Var1;

cb2.Text = All[i].Var2;

cb3.Text = All[i].Var3;

cb1.AutoSize = true;

cb2.AutoSize = true;

cb3.AutoSize = true;

cb1.Top = question.Height + 10;

cb2.Top = question.Height + cb1.Height + 20;

cb3.Top = question.Height + cb1.Height + cb2.Height + 30;

tabControl1.TabPages[i].Controls.Add(cb1);

tabControl1.TabPages[i].Controls.Add(cb2);

tabControl1.TabPages[i].Controls.Add(cb3);

}

}//end if 3 questions

if (!stop)

{

stop = true;

if (All[i].True1 == true)

trueCount += 1;

if (All[i].True2 == true)

trueCount += 1;

if (trueCount == 1)

{

RadioButton rd1 = new RadioButton();

RadioButton rd2 = new RadioButton();

rd1.Text = All[i].Var1;

rd2.Text = All[i].Var2;

rd1.AutoSize = true;

rd2.AutoSize = true;

rd1.Top = question.Height + 10;

rd2.Top = question.Height + rd1.Height + 20;

tabControl1.TabPages[i].Controls.Add(rd1);

tabControl1.TabPages[i].Controls.Add(rd2);

}

else

{

CheckBox cb1 = new CheckBox();

CheckBox cb2 = new CheckBox();

cb1.Text = All[i].Var1;

cb2.Text = All[i].Var2;

cb1.AutoSize = true;

cb2.AutoSize = true;

cb1.Top = question.Height + 10;

cb2.Top = question.Height + cb1.Height + 20;

tabControl1.TabPages[i].Controls.Add(cb1);

tabControl1.TabPages[i].Controls.Add(cb2);

}

}//end if 2 questions

Button bt = new Button();

bt.Dock = DockStyle.Bottom;

bt.Height = 80;

if (i != All.Count - 1)//start if not last question

{

bt.Text = "Следующий вопрос";

bt.Click += new EventHandler(this.NextPage\_Click);

}//end if not last question

else

{

bt.Text = "Закончить тест";

bt.Click += new EventHandler(this.LastPage\_Click);

}

tabControl1.TabPages[i].Controls.Add(bt);

}

}

//end creating test

}//end Initialize

public void \_checkAnswer()

{

int helper = 0;

if (All[tabControl1.SelectedIndex].True1)

helper += 1;

if (All[tabControl1.SelectedIndex].True2)

helper += 1;

if ((All[tabControl1.SelectedIndex].True3) && (All[tabControl1.SelectedIndex].Var3 != ""))

helper += 1;

if ((All[tabControl1.SelectedIndex].True4) && (All[tabControl1.SelectedIndex].Var4 != ""))

helper += 1;

if ((All[tabControl1.SelectedIndex].True5) && (All[tabControl1.SelectedIndex].Var5 != ""))

helper += 1;

if ((All[tabControl1.SelectedIndex].True6) && (All[tabControl1.SelectedIndex].Var6 != ""))

helper += 1;

if (helper == 1)

{

if (All[tabControl1.SelectedIndex].True1)

{

if (((RadioButton)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[1]).Checked)

{

right\_count += 1;

}

}

if (All[tabControl1.SelectedIndex].True2)

{

if (((RadioButton)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[2]).Checked)

{

right\_count += 1;

}

}

if ((All[tabControl1.SelectedIndex].True3) && (All[tabControl1.SelectedIndex].Var3!=""))

{

if (((RadioButton)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[3]).Checked)

{

right\_count += 1;

}

}

if ((All[tabControl1.SelectedIndex].True4)&& (All[tabControl1.SelectedIndex].Var4 != ""))

{

if (((RadioButton)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[4]).Checked)

{

right\_count += 1;

}

}

if ((All[tabControl1.SelectedIndex].True5)&& (All[tabControl1.SelectedIndex].Var5 != ""))

{

if (((RadioButton)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[5]).Checked)

{

right\_count += 1;

}

}

if ((All[tabControl1.SelectedIndex].True6)&& (All[tabControl1.SelectedIndex].Var6 != ""))

{

if (((RadioButton)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[6]).Checked)

{

right\_count += 1;

}

}

}

else

{

int help2 = 0;

if (All[tabControl1.SelectedIndex].True1)

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[1]).Checked)

{

help2 += 1;

}

}

if (!All[tabControl1.SelectedIndex].True1)

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[1]).Checked)

{

help2 -= 1;

}

}

if (All[tabControl1.SelectedIndex].True2)

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[2]).Checked)

{

help2 += 1;

}

}

if (!All[tabControl1.SelectedIndex].True2)

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[2]).Checked)

{

help2 -= 1;

}

}

if ((All[tabControl1.SelectedIndex].True3)&& (All[tabControl1.SelectedIndex].Var3 != ""))

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[3]).Checked)

{

help2 += 1;

}

}

if ((!All[tabControl1.SelectedIndex].True3)&& (All[tabControl1.SelectedIndex].Var3 != ""))

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[3]).Checked)

{

help2 -= 1;

}

}

if ((All[tabControl1.SelectedIndex].True4)&& (All[tabControl1.SelectedIndex].Var4 != ""))

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[4]).Checked)

{

help2 += 1;

}

}

if ((!All[tabControl1.SelectedIndex].True4)&& (All[tabControl1.SelectedIndex].Var4 != ""))

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[4]).Checked)

{

help2 -= 1;

}

}

if ((All[tabControl1.SelectedIndex].True5)&& (All[tabControl1.SelectedIndex].Var5 != ""))

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[5]).Checked)

{

help2 += 1;

}

}

if ((!All[tabControl1.SelectedIndex].True5)&& (All[tabControl1.SelectedIndex].Var5 != ""))

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[5]).Checked)

{

help2 -= 1;

}

}

if ((All[tabControl1.SelectedIndex].True6)&& (All[tabControl1.SelectedIndex].Var6 != ""))

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[6]).Checked)

{

help2 += 1;

}

}

if ((!All[tabControl1.SelectedIndex].True6)&& (All[tabControl1.SelectedIndex].Var6 != ""))

{

if (((CheckBox)tabControl1.TabPages[tabControl1.SelectedIndex].Controls[6]).Checked)

{

help2 -= 1;

}

}

if (help2 == helper)

right\_count += 1;

}

}

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

{

all\_count += 1;

\_checkAnswer();

tabControl1.SelectTab(tabControl1.SelectedIndex+1);

}

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

{

all\_count += 1;

\_checkAnswer();

SqlConnection conn = new SqlConnection(Conn.Default.connStr);

try

{

//пробуем подключится

conn.Open();

}

catch (SqlException se)

{

MessageBox.Show("Ошибка подключения:{0}", se.Message);

}

int mark = 0;

DateTime dt = DateTime.Now;

if (((Double)right\_count \* 100 / all\_count) > 90)

{

mark = 5;

}

else if (((Double)right\_count \* 100 / all\_count) > 74)

{

mark = 4;

}

else if (((Double)right\_count \* 100 / all\_count) > 60)

{

mark = 3;

}

else

mark = 2;

SqlCommand cmd = new SqlCommand(String.Format("INSERT INTO People VALUES ('{0}', '{1}', '{2}', '{3} из {4}', {5}, '{6}');", name , group, testName, right\_count, all\_count, mark, dt)

, conn);

try

{

cmd.ExecuteNonQuery();

}

catch(SqlException ex)

{

MessageBox.Show("Ошибка добавления записи:"+ex.Message);

}

String done;

if (mark == 2)

done = "К сожалению, по результатам теста Вы нам не подходите, попробуйте пройти тестирование позже";

else

done = "Поздравляем, Вы успешно прошли тестирование, приходите на стажировку.";

//ок

MessageBox.Show(String.Format("{0} вы ответили правильно на:\n{1} из {2} вопросов\nВаша оценка:{3}\n{4}", name, right\_count, all\_count, mark, done), "", MessageBoxButtons.OK, MessageBoxIcon.Information);

conn.Close();

conn.Dispose();

this.Close();

}

private void tabControl1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

}

//Лістинг форми ViewPeople

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace TesterCourseProject

{

public partial class ViewPeople : Form

{

bool filter = false;

public ViewPeople()

{

InitializeComponent();

\_update();

}

public void \_update()

{

dataGridView1.Rows.Clear();

if(!filter)

{

SqlConnection conn = new SqlConnection(Conn.Default.connStr);

conn.Open();

String sql = "SELECT \* FROM People";

SqlCommand command = new SqlCommand(sql, conn);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

dataGridView1.Rows.Add(reader.GetValue(1).ToString(), reader.GetValue(2).ToString(), reader.GetValue(3).ToString(), reader.GetValue(4).ToString(), reader.GetValue(5).ToString(), reader.GetValue(6).ToString());

}

conn.Close();

conn.Dispose();

}//заполняем таблицу

else

{

SqlConnection conn = new SqlConnection(Conn.Default.connStr);

conn.Open();

String sql = String.Format("SELECT \* FROM People WHERE Cast(Cast([When] As VarChar(11)) As DateTime)>='{0}' and Cast(Cast([When] As VarChar(11)) As DateTime)<='{1}'", dateTimePicker1.Value.Date.ToShortDateString(),dateTimePicker2.Value.Date.ToShortDateString());

SqlCommand command = new SqlCommand(sql, conn);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

dataGridView1.Rows.Add(reader.GetValue(1).ToString(), reader.GetValue(2).ToString(), reader.GetValue(3).ToString(), reader.GetValue(4).ToString(), reader.GetValue(5).ToString(), reader.GetValue(6).ToString());

}

conn.Close();

conn.Dispose();

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

\_update();

}

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

{

filter = true;

if(dateTimePicker1.Value>dateTimePicker2.Value)

{

DateTime dt = dateTimePicker1.Value;

dateTimePicker1.Value = dateTimePicker2.Value;

dateTimePicker2.Value = dt;

}

\_update();

}

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

{

filter = false;

\_update();

}

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

{

}

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

{

}

}

}