Министерство науки и высшего образования Российской Федерации

Государственное бюджетное образовательное учреждение  
высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)»

Московский техникум космического приборостроения

**КУРСОВОЙ ПРОЕКТ**

по теме: «Интеллектуальные мехатронные системы»

Пояснительная записка

Специальность 09.02.05 Прикладная информатика

МТКП.130311.000 ПЗ

|  |  |  |
| --- | --- | --- |
| Руководитель разработки |  | С.Р. Гуриков |
|  | (подпись, дата) |  |
| Студент |  | Н.А. Аксенов |
|  | (подпись, дата) |  |

Москва 2018

|  |
| --- |
|  |

# **СОДЕРЖАНИЕ**

[**СОДЕРЖАНИЕ** 2](#_Toc185652)

[**Введение** 3](#_Toc185653)

[**1** **Общий раздел** 5](#_Toc185654)

[**1.1 Системные требование** 5](#_Toc185655)

[**1.1** **Характеристики системы программирования** 5](#_Toc185656)

[**2 Технологический раздел** 8](#_Toc185657)

[**2.1 Постановка задач** 8](#_Toc185658)

[**2.1.1 Описательная модель задачи** 8](#_Toc185659)

[**2.1.2 Описание входной информации** 8](#_Toc185660)

[**2.1.3 Описание выходной информации** 8](#_Toc185661)

[**3 Руководство по использованию программы** 11](#_Toc185662)

[**3.1Руководство пользователя** 11](#_Toc185663)

[**3.1.1 Назначение программы** 11](#_Toc185664)

[**3.1.2 Условия выполнения программы** 11](#_Toc185665)

[**3.1.3 Входные данные** 11](#_Toc185666)

[**3.1.4 Выполнение программы** 11](#_Toc185667)

[**Приложение Б** 12](#_Toc185668)

[**3.1.5 Сообщения** 19](#_Toc185669)

[**3.1.6 Выходные данные** 20](#_Toc185670)

[**Заключение** 21](#_Toc185671)

[**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ** 22](#_Toc185672)

[**Приложение А листинг программы** 23](#_Toc185673)

# **Введение**

|  |
| --- |
|  |

В настоящее время мехатроника наряду с информатикой, био–инженерией и нано технологиями оказывает существенное влияние на развитие производственной и бытовой техно сфер в направлении все более широкого внедрения систем мехатронной автоматизации и роботизации в различные физико-технические процессы во всех сферах деятельности общества. Целью мехатроники является создание интеллектуальных физико-технических изделий, систем и процессов, обладающих качественно новыми функциями, свойствами и возможностями. Частным случаем таких физико-технических изделий являются роботы и робототехнические системы различного назначения и размерностей. Современная мехатроника и робототехника охватывает все размерные масштабы технических систем от «макро» до «микро» (размеры от 1 мк до 1 мм) с последующим переходом от микросистем к нано-системам (порядка ста нанометров – 10–7 м): переход «макро-микро нано-мехатроника и робототехника». Мехатроника – область науки и техники, основанная на системном объединении узлов точной механики, датчиков состояния внешней среды и самого объекта, источников энергии, исполнительных механизмов, усилителей, вычислительных устройств (ЭВМ и микропроцессоры). Мехатронная система – единый комплекс электромеханических, электрогидравлических, электронных элементов и средств вычислительной техники, между которыми осуществляется постоянный динамически меняющийся обмен энергией и информацией, объединенный общей системой автоматического управления, обладающей элементами искусственного интеллекта

**Актуальность темы** Успехи в области мехатроники, микро (нано) процессорной техники и информационных технологий приводят к необходимости разработки и создания нового типа систем обработки информации и управления – интеллектуальных. Этот тип систем особенно важен в мехатронике, поскольку проектирование механических систем и их систем управления должно осуществляться как единое целое – интегрированные системы. При этом надо учитывать, что одни проблемы могут быть решены легко и просто в физических и механических образцах, другие в «информационных» образцах – в микроконтроллерах, объединенных информационным процессом, реализованным в том числе в программном обеспечении.

Современная мехатроника появилась не на пустом месте, ей предшествовал огромный опыт комплексирования механических устройств, радио- и оптоэлектроники, аналоговых и цифровых вычислительных средств в сложных системах различного назначения. Компоненты этих систем представляют собой отдельные блоки, которые соединялись с помощью различных преобразователей. Однако, как и в нынешней мехатронике, все эти блоки в системе объединялись информационным процессом, включающим получение и обработку измерительной информации, ее распознавание, прогнозирование, выработку управления, исполнение управления действием и контроль результатов действия.

**Предметом исследования**, «Интеллектуальные мехатронные системы»

**Цель:** Исследовать теоретический материал и разработать электронное пособие по теме «Интеллектуальные мехатронные системы» и тест по данной теме.

**Задачи:**

1. Изучить теоретический материал по теме «Интеллектуальные мехатронные системы».
2. Разработать электронное учебное пособие, а также тестовую программу.

# **Общий раздел**

## **1.1 Системные требование**

**Рекомендуемые системные требования:** к среде разработке

* Процессор с тактовой частотой не ниже 1,8 ГГц. Рекомендуется использовать как минимум двухъядерный процессор.
* ГБ ОЗУ; рекомендуется 4 ГБ ОЗУ (минимум 2,5 ГБ при выполнении на виртуальной машине)
* Место на жестком диске: до 130 ГБ свободного места в зависимости от установленных компонентов, обычно для установки требуется от 20 до 50 ГБ свободного места
* Скорость жесткого диска: для повышения производительности установите Windows и Visual Studio на твердотельный накопитель (SSD)
* Видеоадаптер с минимальным разрешением 720p (1280 на 720 пикселей); для оптимальной работы Visual Studio рекомендуется разрешение WXGA (1366 на 768 пикселей) или более высокое.

## **Характеристики системы программирования**

Система программирования – это система для разработки новых программ на конкретном языке программирования.

Современные системы программирования обычно предоставляют пользователям мощные и удобные средства разработки программ.

В них входят:

* Компилятор или интерпретатор
* Интегрированная среда разработки
* Средства создания и редактирования текстов программ
* Обширные библиотеки стандартных программ и функций
* Отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе
* Утилиты для работы с библиотеками

**C#** — это типизированный, объектно-ориентированный, простой и в то же время мощный язык программирования, который позволяет разработчикам создавать многофункциональные приложения.

Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET (программной платформы от компании Microsoft, предназначенной для создания обычных программ и веб-приложений). C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java.

**Visual Studio** — это интегрированная среда разработки (IDE) от Microsoft, основной инструмент разработки приложений для платформы .NET и Windows в целом. Вы можете разрабатывать приложения на языках C#, VB.NET, F# и C++/CLI.

## **2 Технологический раздел**

## **2.1 Постановка задач**

### **2.1.1 Описательная модель задачи**

Целью курсовой работы стало, исследование теоретического материала по теме «Интеллектуальные мехатронные системы», а также создание тестового приложения по данному материалу.

Следующие задачи по курсовому проекту:

1. Изучить теоретический материал по теме «Интеллектуальные мехатронные системы».
2. Разработать тестовую программу по теме «Интеллектуальные мехатронные системы».
3. Реализовать вывод теста в txt файл.

### **2.1.2 Описание входной информации**

Входными данными в данном случае являются материалы из пособия «Интеллектуальные мехатронные системы»

### **2.1.3 Описание выходной информации**

Выходной информацией являются результаты тестирования, полученные входе выполнения тестовых заданий.

По завершению пользователем программы теста выходными данными будут являться результаты теста, выведенные в формате TXT.

**2.1.5 Требования к программе**

На этапе разработки технического задания к программе предъявляются следующие функциональные требования:

Разработка приложения в среде Visual Studio.

1. Использовать язык программирования C#.
2. Реализовать вывод тестирования в txt файл.
3. Разработать появление, исчезновение формы
4. В приложении были использованы следующие объекты (**TextBox**, **RadioButton, CheckButton, ListBox, ComboBox, Hscrollbar**)

Не функциональные требования:

1. Создать простой, удобный и понятный интерфейс для пользователя.
2. Вывод результатов тестирования в txt файл.
   1. **Логическая модель задачи**

На рисунке 1 представлена блок-схема.

**Untitled Diagram (5)**

Рисунок 1 Логическая модель задачи

# **3 Руководство по использованию программы**

## **3.1Руководство пользователя**

### **3.1.1 Назначение программы**

Программный продукт разработан для учителей, учеников, в целях ознакомления с темой курсового проекта «Интеллектуальные мехатронные системы». Программа содержит в себе электронное пособие и тестовую часть по теме курсового проекта.

### **3.1.2 Условия выполнения программы**

1. Изучить теоретический материал по теме «Интеллектуальные мехатронные системы».
2. Разработать электронное учебное пособие, а также тестовую программу по теме курсовой работы.

### **3.1.3 Входные данные**

Поскольку назначением программы не является расчёт каких-либо событий в программе отсутствуют математические, физические и какие-либо другие формулы. Входными данными в данном случае являются материалы из пособия «Интеллектуальные мехатронные системы»

### **3.1.4 Выполнение программы**

Для того чтобы запустить программу пользователю нужно перейти в папку куда была скачана программа, открыть папку, которая получилась после разархивации, и запустить приложение WindowsFormsApplication1.exe щелкнув по нему двойным кликом мыши.

При запуске программы перед пользователем запустится главное меню программы с опциями выбора:

# **Приложение Б**

С помощью элемента управления **Label** реализуется текст вопроса, отвечая на который, в зависимости от поставленного вопроса пользователь должен внести или дописать ответ, который обрабатывает в компоненте **TextBox**.

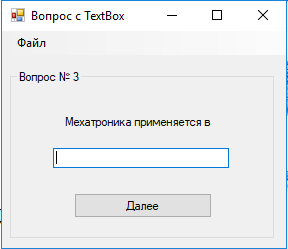


Рисунок 2 Компонент TextBox

Компонент **RadioButton** позволяет пользователю выбрать правильный вариант ответа из множества предложенных.

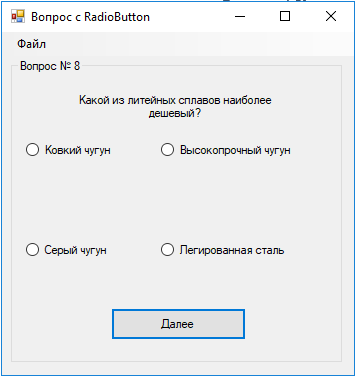


Рисунок 3 Компонент RadioButton

Компонент **CheckBox** или флажок предназначен для установки одного из двух значений: отмечен и не отмечен, также данный элемент управления подразумевает выбор нескольких вариантов ответа из предложенного множества.

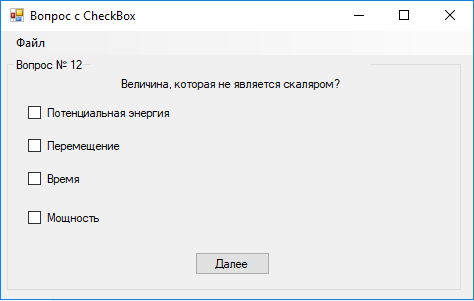


Рисунок 4 Компонент ChekBox

Элемент **Button** позволяет соотнести ответы с вариантами вопроса. Ответы на вопросы находятся в элементе **ListBox,** по нажатию одной из кнопок, выбранный вариант из списка переходит в один из нескольких элементов **TextBox**.

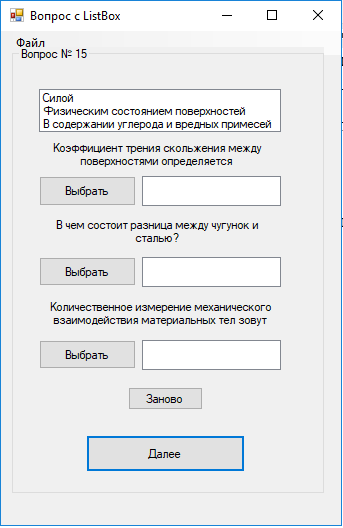


Рисунок 5 Компонент ButtonListBox

Элемент **ScrollBar** – это ползунок, при переносе которого генерируется значение, данное значение является вариантом ответа на вопрос.

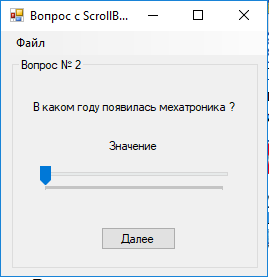


Рисунок 6 Компонент **ScrollBar**

**ComboBox** – это выпадающий список, совмещающий в себе функциональность компонентов **ListBox** и **TextBox**. Этот элемент управления позволяет пользователю выбрать готовые значения из списка.

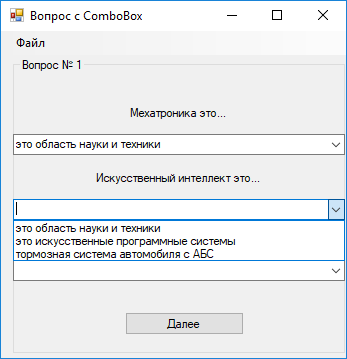


Рисунок 7 Компонент **ComboBox**

**CheckedListBox -** эта форма элемента управления представляет собой симбиоз компонентов **ListBox** и **CheckBox**. Для выбора ответа необходимо поставить галочку рядом с выбранным вариантом.

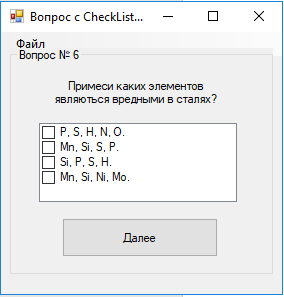
****

Рисунок 8 Компонент **CheckedListBox**

**CheckBox** в **ListBox–**данный элемент управления позволяет добавить элемент **CheckBox** в список **ListBox**. Если **CheckBox** выбран, то он автоматически заносится в список, иначе он удаляется из списка.

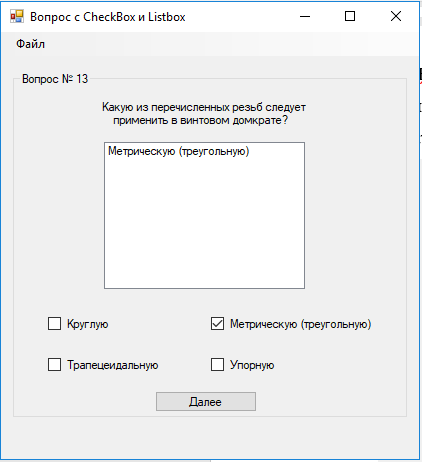
****

Рисунок 9 Компонент **CheckBox** в **ListBox**

### **3.1.5 Сообщения**

Программа предусматривает взаимодействие с пользователем посредством появления формы **MessageBox**.

Данная форма появляется в тот момент, когда пользователь решает перейти на следующий этап без заполнения полей или выбора предложенных вариантов ответа.

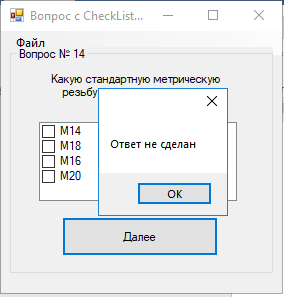


Рисунок 10 Диалоговое окно **MessageBox**

### **3.1.6 Выходные данные**

Выходной информацией являются, результаты тестирования, полученные входе выполнения тестовых заданий, выведенные в табличном виде.

# **Заключение**

Подводя итоги проделанной работы над курсовым проектом «Интеллектуальные мехатронные системы» следует отметить, что все поставленные цели и задачи были успешно реализованы.

1. Изучен теоретический материал по теме «Интеллектуальные мехатронные системы».
2. Разработано электронное учебное пособие, а также тестовая программа по теме курсовой работы.
3. Осуществлен вывод результата тестирования в элементе управления **dataGridView**.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Сырямкин В.И. «Интеллектуальные робототехнические и мехатронные системы».
2. Введение в мехатронику, Грабченко А.И., Клепиков В.Б., Доброскок В.Л., 2014.

Интернет ресурсы:

1. <https://www.twirpx.com/file/84330/>
2. <https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%85%D0%B0%D1%82%D1%80%D0%BE%D0%BD%D0%B8%D0%BA%D0%B0>
3. https://megaobuchalka.ru/1/17920.html

# **Приложение А листинг программы**

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

Questions.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

class Question

{

//массив правельных ответов

public static string[] trueAns = new string[16]

{ "это область науки и техники,это искусственные программные системы,тормозная система автомобиля с АБС", "1930", "медицинском и спортивном оборудование", "Авиационных и космических техниких направлениях",

"Одной формулой", "P, S, H, N, O.", "Литьем,Равновесие тел под действием сил,электроскопическйи кристалл", "Серый чугун",

"Персональный компьютер,совокупность нескольких мехатронных модулей и узлов,это функционально и конструктивно самостоятельное изделие для реализации движений", "1969", "транспондер", "Потенциальная энергия",

"Метрическую (треугольную)", "М16", "Физическим состоянием поверхностей,В содержании углерода и вредных примесей,Силой", "Мартенсит" };

//массив ответов пользователя

public static string[] polzAns = new string[17];

//заполение вопроса с ComboBox

public static void comboboxq(Label a1, Label a2, Label a3, ComboBox c1, ComboBox c2, ComboBox c3, int index)

{

if (index <= 8)

{

a1.Text = "Мехатроника это...";

a2.Text = "Искусственный интеллект это...";

a3.Text = "что явл мехатрнной системой ?";

c1.Items.Add("это область науки и техники");

c1.Items.Add("это искусственные программные системы");

c1.Items.Add("тормозная система автомобиля с АБС");

c2.Items.Add("это область науки и техники");

c2.Items.Add("это искусственные программные системы");

c2.Items.Add("тормозная система автомобиля с АБС");

c3.Items.Add("это область науки и техники");

c3.Items.Add("это искусственные программные системы");

c3.Items.Add("тормозная система автомобиля с АБС");

}

else

{

a1.Text = "что явл мехатрнной системой ?";

a2.Text = "Мехатронная система это...";

a3.Text = "Мехатронный модуль это...";

c1.Items.Add("Персональный компьютер");

c1.Items.Add("совокупность нескольких мехатронных модулей и узлов");

c1.Items.Add("это функционально и конструктивно самостоятельное изделие для реализации движений");

c2.Items.Add("Персональный компьютер");

c2.Items.Add("совокупность нескольких мехатронных модулей и узлов");

c2.Items.Add("это функционально и конструктивно самостоятельное изделие для реализации движений");

c3.Items.Add("Персональный компьютер");

c3.Items.Add("совокупность нескольких мехатронных модулей и узлов");

c3.Items.Add("это функционально и конструктивно самостоятельное изделие для реализации движений");

}

}

//заполение вопроса с TrackBar

public static void trackq(Label a1, int index)

{

if (index <= 8)

{

a1.Text = "В каком году появилась мехатроника ?";

}

else

{

a1.Text = "Когда появился термин механтроника? ";

}

}

public static void textboxq(Label a1, int index)

{

if (index >= 8)

{

a1.Text = "Количественное измерение механического взаимодействия материальных тел зовут";

}

else

{

a1.Text = "Мехатроника применяется в ";

}

}

//заполение вопроса с CheckBox

public static void checkboxq(Label a1, CheckBox q1, CheckBox q2, CheckBox q3, CheckBox q4, int index)

{

if (index <= 8)

{

a1.Text = "Мехатроника применяется в";

q1.Text = "Сплаве";

q2.Text = "Авиационных и космических техниких направлениях";

q3.Text = "Уравновешенность системы";

q4.Text = "Ни одно из перечисленных";

}

else

{

a1.Text = "Величина, которая не является скаляром?";

q1.Text = "Потенциальная энергия";

q2.Text = "Перемещение";

q3.Text = "Время";

q4.Text = "Мощность";

}

}

//заполение вопроса с CheckBox и ListBox

public static void checklistboxq(Label a1, CheckBox q1, CheckBox q2, CheckBox q3, CheckBox q4, int index)

{

if (index <= 8)

{

a1.Text = "Дифференциальное уравнение вращательного движения тела можно записать";

q1.Text = "Трех формулах";

q2.Text = "Имеет однозначное выражение";

q3.Text = "Одной формулой";

q4.Text = "Двух формулах";

}

else

{

a1.Text = " Какую из перечисленных резьб следует применить в винтовом домкрате? ";

q1.Text = "Круглую";

q2.Text = "Метрическую (треугольную)";

q3.Text = "Трапецеидальную";

q4.Text = "Упорную";

}

}

//заполение вопроса с CheckedListBox

public static void checkedlistboxq(Label a1, CheckedListBox q1, int index)

{

if (index <= 8)

{

a1.Text = "Примеси каких элементов являються вредными в сталях?";

q1.Items.Clear();

q1.Items.Add("P, S, H, N, O.");

q1.Items.Add("Mn, Si, S, P. ");

q1.Items.Add("Si, P, S, H. ");

q1.Items.Add("Mn, Si, Ni, Mo. ");

}

else

{

a1.Text = "Какую стандартную метрическую резьбу нужно назначить для соединения?";

q1.Items.Clear();

q1.Items.Add("М14");

q1.Items.Add("М18");

q1.Items.Add("М16");

q1.Items.Add("М20");

}

}

//заполение вопроса с ListBox

public static void listboxq(Label a1, Label a2, Label a3, ListBox q1, int index)

{

if (index <= 8)

{

a1.Text = "Каким способом изготавливается большинство чугунных изделий?";

a2.Text = "Статика - это раздел теоретической механики, которая изучает";

a3.Text = "Укажите, какой подшипник может воспринимать только осевую нагрузку?";

q1.Items.Clear();

q1.Items.Add("электроскопическйи кристалл");

q1.Items.Add("Равновесие тел под действием сил");

q1.Items.Add("Литьем");

}

else

{

a1.Text = "Коэффициент трения скольжения между поверхностями определяется";

a2.Text = "В чем состоит разница между чугунок и сталью?";

a3.Text = "Количественное измерение механического взаимодействия материальных тел зовут";

q1.Items.Clear();

q1.Items.Add("Силой");

q1.Items.Add("Физическим состоянием поверхностей");

q1.Items.Add("В содержании углерода и вредных примесей");

}

}

public static void radiobuttonq(Label a1, RadioButton q1, RadioButton q2, RadioButton q3, RadioButton q4, int index)

{

//заполение вопроса с RaddioButton

if (index <= 9)

{

a1.Text = "Какой из литейных сплавов наиболее дешевый?";

q1.Text = "Ковкий чугун";

q2.Text = "Серый чугун";

q3.Text = "Высокопрочный чугун";

q4.Text = "Легированная сталь";

}

else

{

a1.Text = "Какая структура образуется в стали после закалки?";

q1.Text = "Мартенсит";

q2.Text = "Цементит";

q3.Text = "Феррит";

q4.Text = "Перлит";

}

}

}

}

Form1

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 WindowsFormsApplication1

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

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

{

//увеличиваем индекс переключения форм на 1

Form1.i++;

//добавляем menuStrip

Staff.Mainmenu(menuStrip1);

//добавления текста на кнопку переключения формы и номера вопроса

Staff.other(button1, groupBox1, Form1.i);

//заполенение формы вопросами и вариантами ответов

Question.comboboxq(label2,label3,label4,comboBox1,comboBox2,comboBox3,Form1.i);

}

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

{

//обработка ошибки при которой пользователь не отвечает на все вопросы

if ((comboBox1.SelectedIndex < 0) || (comboBox2.SelectedIndex < 0) || (comboBox3.SelectedIndex < 0))

{

MessageBox.Show("Не все ответы выбраны");

}

else

{

//добавление ответа в массив ответов пользователя

Question.polzAns[Form1.i - 2] = comboBox1.Text + "," + comboBox2.Text + "," + comboBox3.Text;

//переключенение формы

Staff.Accept(Form1.i,this);

}

}

Form2

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 WindowsFormsApplication1

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

//демонстрация выбранного значения

label3.Text = Convert.ToString("Выбранное значение :" + trackBar1.Value);

}

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

{

//увеличиваем индекс переключения форм на 1

Form1.i++;

//добавляем menuStrip

Staff.Mainmenu(menuStrip1);

//добавления текста на кнопку переключения формы и номера вопроса

Staff.other(button1, groupBox1, Form1.i);

//заполенение формы вопросами и вариантами ответов

Question.trackq(label1,Form1.i);

}

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

{

//добавления в массив ответа пользователя

Question.polzAns[Form1.i-2] = Convert.ToString(trackBar1.Value);

//переключения формы

Staff.Accept(Form1.i, this);

}

Form3

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 WindowsFormsApplication1

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

//обработка введения цифр

char ch = e.KeyChar;

if (Char.IsDigit(ch))

{

e.Handled = true;

MessageBox.Show("Цифры к вводу не допустимы", "Ошибка");

}

}

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

{

//увеличиваем индекс переключения форм на 1

Form1.i++;

//добавляем menuStrip

Staff.Mainmenu(menuStrip1);

//добавления текста на кнопку переключения формы и номера вопроса

Staff.other(button1, groupBox1, Form1.i);

//заполенение формы вопросами и вариантами ответов

Question.textboxq(label1, Form1.i);

}

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

{

//обработка ошибки при которой пользователь не отвечает

if (textBox1.Text == "")

{

MessageBox.Show("Введите ответ!");

}

else

{

//добавление в массив ответа пользователя

Question.polzAns[Form1.i-2] = textBox1.Text.ToLower();

//переключение формы

Staff.Accept(Form1.i, this); } }

Form4

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 WindowsFormsApplication1

{

public partial class Form5 : Form

{

//переменная содержащие количестов выбранных ответов

int chek;

public Form5()

{

InitializeComponent();

}

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

{

//увеличиваем индекс переключения форм на 1

Form1.i++;

//добавляем menuStrip

Staff.Mainmenu(menuStrip1);

//добавления текста на кнопку переключения формы и номера вопроса

Staff.other(button1, groupBox1, Form1.i);

//заполенение формы вопросами и вариантами ответов

Question.checkboxq(label2, checkBox1, checkBox2, checkBox3, checkBox4, Form1.i);

}

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

{

//обработка ошибки при которой пользователь не выбрал ответ

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

{

MessageBox.Show("Ответ не выбран");

}

else if (chek == 1)

{

//добавление в массив ответов пользователя

if (checkBox1.Checked == true)

Question.polzAns[Form1.i-2] = checkBox1.Text;

if (checkBox2.Checked == true)

Question.polzAns[Form1.i-2] = checkBox2.Text;

if (checkBox3.Checked == true)

Question.polzAns[Form1.i-2] = checkBox3.Text;

if (checkBox4.Checked == true)

Question.polzAns[Form1.i-2] = checkBox4.Text;

//перключение формы

Staff.Accept(Form1.i,this);

}

else

MessageBox.Show("Можно только 1");

}

//проверка количества выбранных ответов

private void checkBox2\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox2.Checked == true)

chek++;

else

chek--;

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked == true)

chek++;

else

chek--;

}

private void checkBox3\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox3.Checked == true)

chek++;

else

chek--;

}

private void checkBox4\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox4.Checked == true)

chek++;

else

chek--;

}

}

}

Form5

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 WindowsFormsApplication1

{

public partial class Form6 : Form

{

int chek;

public Form6()

{

InitializeComponent();

}

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

{

//увеличиваем индекс переключения форм на 1

Form1.i++;

//добавляем menuStrip

Staff.Mainmenu(menuStrip1);

//добавления текста на кнопку переключения формы и номера вопроса

Staff.other(button1, groupBox1, Form1.i);

//заполенение формы вопросами и вариантами ответов

Question.checklistboxq(label1, checkBox1, checkBox2, checkBox3, checkBox4, Form1.i);

}

//добавление выбранного ответа в ListBox

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked == true)

{

listBox1.Items.Add(checkBox1.Text);

chek++;

}

else

{

listBox1.Items.Clear();

chek--;

}

}

private void checkBox2\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox2.Checked == true)

{

listBox1.Items.Add(checkBox2.Text);

chek++;

}

else

{

chek--;

listBox1.Items.Clear();

}

}

private void checkBox3\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox3.Checked == true)

{

chek++;

listBox1.Items.Add(checkBox3.Text);

}

else

{

chek--;

listBox1.Items.Clear();

}

}

private void checkBox4\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox4.Checked == true)

{

chek++;

listBox1.Items.Add(checkBox4.Text);

}

else

{

chek--;

listBox1.Items.Clear();

}

}

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

{

//обработка ошибки пользователя при которой он не ответил на все вопросы

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

{

MessageBox.Show("Выберете ответ!");

}

else if (chek>1)

{

MessageBox.Show("Можно только 1");

checkBox1.Checked = false;

checkBox2.Checked = false;

checkBox3.Checked = false;

checkBox4.Checked = false;

listBox1.Items.Clear();

}

else

{

//добавление в массив ответов пользователя

if (checkBox1.Checked == true)

Question.polzAns[Form1.i-2] = checkBox1.Text;

if (checkBox2.Checked == true)

Question.polzAns[Form1.i-2] = checkBox2.Text;

if (checkBox3.Checked == true)

Question.polzAns[Form1.i-2] = checkBox3.Text;

if (checkBox4.Checked == true)

Question.polzAns[Form1.i-2] = checkBox4.Text;

//переключение формы

Staff.Accept(Form1.i, this);

}

}

}

}

Form6

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 WindowsFormsApplication1

{

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

}

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

{

if (checkedListBox1.CheckedItems.Count == 1)

{

//доавление в массив ответов пользователя

if (checkedListBox1.GetSelected(0) == true)

Question.polzAns[Form1.i-2] = Convert.ToString(checkedListBox1.Items[0]);

else if (checkedListBox1.GetSelected(1) == true)

Question.polzAns[Form1.i-2] = Convert.ToString(checkedListBox1.Items[1]);

else if (checkedListBox1.GetSelected(2) == true)

Question.polzAns[Form1.i-2] = Convert.ToString(checkedListBox1.Items[2]);

else Question.polzAns[Form1.i-2] = Convert.ToString(checkedListBox1.Items[3]);

Staff.Accept(Form1.i, this);

}

else

//проверка количства ответов

if (checkedListBox1.CheckedItems.Count > 1)

MessageBox.Show("Можно только 1");

else

MessageBox.Show("Ответ не сделан");

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

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

{

//увеличиваем индекс переключения форм на 1

Form1.i++;

//добавляем menuStrip

Staff.Mainmenu(menuStrip1);

//добавления текста на кнопку переключения формы и номера вопроса

Staff.other(button1, groupBox1, Form1.i);

//заполенение формы вопросами и вариантами ответов

Question.checkedlistboxq(label1, checkedListBox1, Form1.i);

}

}

}

Form7

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 WindowsFormsApplication1

{

public partial class Form8 : Form

{

public Form8()

{

InitializeComponent();

}

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

{

//увеличиваем индекс переключения форм на 1

Form1.i++;

//добавляем menuStrip

Staff.Mainmenu(menuStrip1);

//добавления текста на кнопку переключения формы и номера вопроса

Staff.other(button1, groupBox1, Form1.i);

//заполенение формы вопросами и вариантами ответов

Question.listboxq(label1, label2, label3, listBox1, Form1.i);

}

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

{

//доавление в массив ответов пользователя

Question.polzAns[Form1.i - 2] = listBox2.Items[0] + "," + listBox3.Items[0] + "," + listBox4.Items[0];

//перключение формы

Staff.Accept(Form1.i, this);

}

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

{

//добавление выбранного ответа в листбокс

listBox2.Items.Clear();

listBox2.Items.Insert(0, Convert.ToString(listBox1.SelectedItem));

listBox1.Items.RemoveAt(listBox1.SelectedIndex);

button2.Enabled = false;

}

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

{

//добавление выбранного ответа в листбокс

listBox3.Items.Clear();

listBox3.Items.Insert(0, Convert.ToString(listBox1.SelectedItem));

listBox1.Items.RemoveAt(listBox1.SelectedIndex);

button3.Enabled = false;

}

private void button4\_Click(object sender, EventArgs e)

{

//добавление выбранного ответа в листбокс

listBox4.Items.Clear();

listBox4.Items.Insert(0, Convert.ToString(listBox1.SelectedItem));

listBox1.Items.RemoveAt(listBox1.SelectedIndex);

button4.Enabled = false;

}

private void button5\_Click(object sender, EventArgs e)

{

//перезаполнение формы

button2.Enabled = true;

button3.Enabled = true;

button4.Enabled = true;

listBox1.Items.Clear();

listBox2.Items.Clear();

listBox3.Items.Clear();

listBox4.Items.Clear();

Question.listboxq(label1, label2, label3, listBox1, Form1.i);

}

}}

Form8

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 WindowsFormsApplication1

{

public partial class Form9 : Form

{

public Form9()

{

InitializeComponent();

}

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

{

//обработка ошибки пользователя при которой он не ответил на все вопросы

if ((radioButton1.Checked == false) && (radioButton2.Checked == false) && (radioButton3.Checked == false) && (radioButton4.Checked == false))

MessageBox.Show("Выбирите ответ!");

else

{

//добавление в массив ответов пользователя

if (radioButton1.Checked == true)

Question.polzAns[Form1.i - 2] = radioButton1.Text;

if (radioButton2.Checked == true)

Question.polzAns[Form1.i - 2] = radioButton2.Text;

if (radioButton3.Checked == true)

Question.polzAns[Form1.i - 2] = radioButton3.Text;

if (radioButton4.Checked == true)

Question.polzAns[Form1.i - 2] = radioButton4.Text;

Staff.Accept(Form1.i, this);

}

}

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

{

//увеличиваем индекс переключения форм на 1

Form1.i++;

//добавляем menuStrip

Staff.Mainmenu(menuStrip1);

//добавления текста на кнопку переключения формы и номера вопроса

Staff.other(button1, groupBox1, Form1.i);

//заполенение формы вопросами и вариантами ответов

Question.radiobuttonq(label1, radioButton1, radioButton2, radioButton3, radioButton4, Form1.i);

}

}

}

Start

using System;

using System.IO;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace WindowsFormsApplication1

{

public partial class Form1 : Form

{

//создание глобальной переменной индекса форомы

public static int i = 1;

public Form1()

{

InitializeComponent();

}

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

{

//переход на первый вопрос теста

Form l = new Form2();

l.Show();

button1.Enabled = false;

}

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

{

//добавление menuStrip

Staff.Mainmenu(menuStrip1);

//откртыие формы с титульным листом

Form tit = new titul();

tit.Show();

}

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

{

//открытие документа с теорией

/\* var application = new Microsoft.Office.Interop.Word.Application();

var document = new Microsoft.Office.Interop.Word.Document();

var docPath = Directory.GetCurrentDirectory() + @"\..\..\..\..\";

application.Visible = true;\*/

System.Diagnostics.Process.Start(Directory.GetCurrentDirectory() + @"\..\..\..\..\text1.pdf");

}

}

}

Titul

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 WindowsFormsApplication1

{

public partial class titul : Form

{

public titul()

{

InitializeComponent();

}

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

{

this.Width = 0;

this.Height = 0;

this.timer1.Enabled = true;

}

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

{

this.Width = this.Width + 50;

this.Height = this.Height + 50;

if ((this.Width == 520) && (this.Height == 550))

{

this.timer1.Enabled = false;

}

}

private void titul\_Paint(object sender, PaintEventArgs e)

{

Image newImage = Image.FromFile("1.jpg");

e.Graphics.DrawImage(newImage, 50, 50, 400, 300);

}

private void label13\_Click(object sender, EventArgs e)

{

}

private void label12\_Click(object sender, EventArgs e)

{

}

}

}

Result

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

public partial class result : Form

{

public result()

{

InitializeComponent();

}

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

{

//добавляем menuStrip

Staff.Mainmenu(menuStrip1);

for (int i = 0; i <= 15; i++)

{

//доавление строк в таблицу

dataGridView1.Rows.Add();

}

//переменная хранящия переменную порядковый номер вопроса

int count = 1;

for (int j = 0; j <= 15; j++)

{

//заполение таблицы результатов теста

dataGridView1.Rows[j].Cells[0].Value = "№" + count ;

dataGridView1.Rows[j].Cells[2].Value = Question.trueAns[j];

dataGridView1.Rows[j].Cells[1].Value = Question.polzAns[j];

//закраска верных и не верных ответов на тест

if (Convert.ToString(dataGridView1.Rows[j].Cells[2].Value) == Convert.ToString(dataGridView1.Rows[j].Cells[1].Value))

{

dataGridView1.Rows[j].Cells[0].Style.BackColor = Color.Green;

dataGridView1.Rows[j].Cells[1].Style.BackColor = Color.Green;

dataGridView1.Rows[j].Cells[2].Style.BackColor = Color.Green;

}

else

{

dataGridView1.Rows[j].Cells[0].Style.BackColor = Color.Red;

dataGridView1.Rows[j].Cells[1].Style.BackColor = Color.Red;

dataGridView1.Rows[j].Cells[2].Style.BackColor = Color.Red;

}

count++;

}

Вывод в текстовый файл

string path = @"D:\test.txt";

using (StreamWriter sw = new StreamWriter(path, true, Encoding.Default))

{

for (int j = 0; j <= 15; j++)

{

string value = "Номер вопроса " + (j + 1) + " правельный ответ: " + Question.trueAns[j] + " ответ пользователя: " + Question.polzAns[j];

sw.WriteLine(value);

}

sw.Close();

}

}

}

}