Трясков СИП22-2

**«Введение»**

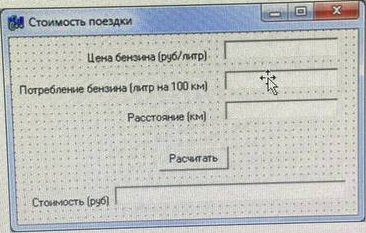
* 1. Описание цели работы:

Практическое освоение работы с компонентами. Изучение методов привязки событий к элементам пользовательского интерфейса. Развитие навыков программирования на C# с использованием среды Visual Studio.

* 1. Обзор задачи:
* Задание №1 (Общее):

Написать программу расчета стоимости поездки на автомобиле, если известна цена бензина (руб./л), потребление бензина (литр на 100км) и расстояние (км).

Компановка формы:



Фотография 1.1

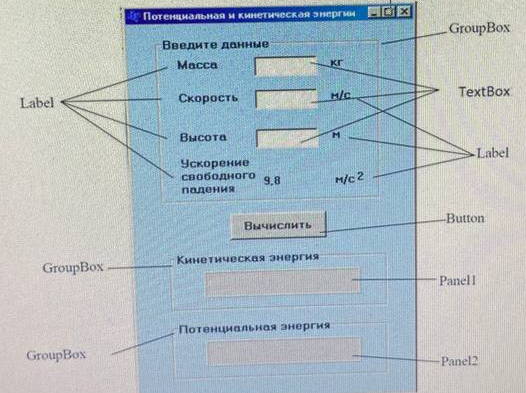
Требования:

1. Поля «Цена бензина», «Потребление бензина» и «Расстояние» должно быть с проверкой вводимых данных для предотвращения ввода нечисловых данных
2. Программа должна выдавать сообщение об ошибке, если не заполнено хотя бы одно поле из трех («Цена бензина», «Потребление бензина» и «Расстояние»)
3. Поле «Стоимость должно быть только для чтения

* Задание №2 – Индивидуальное (Вариант 4):

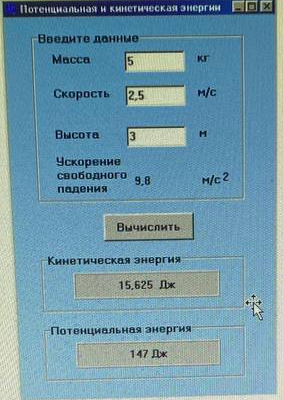
Создание приложения «Потенциальная и кинетическая энергия»

Вид приложения с используемыми компонентами:



Фотография 1.2

Вид приложения после нажатия кнопки «Вычислить»:

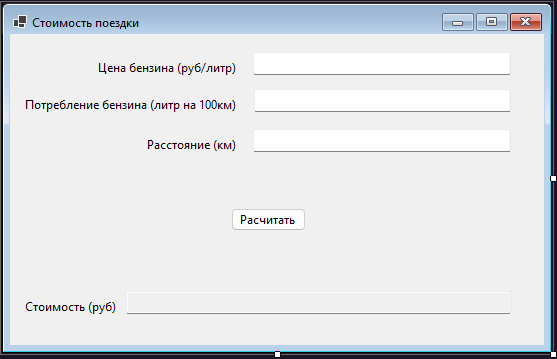


Фотография 1.3

**«Описание приложения»**

* 1. Краткое описание функциональных возможностей приложения:

1. Описание общего задания – ввод 3 данных в виде цены, потребления и расстояния. Кнопка для расчета стоимости на основе введенных данных. Поле вывода результата
2. Описание индивидуального задания – 3 поля для ввода данных в виде массы, скорости и высоты. Кнопка для вычисления кинетической и потенциальной энергии на основе введенных данных. 2 панели для вывода результатов
   1. Обзор интерфейса пользователя, включая элементы управления и их расположение:
3. Интерфейс общего задания:



## Фотография 2.1

1. Интерфейс индивидуального задания:

## 

## Рисунок 2.2

* 1. Описание используемых компонентов и их роль в создании приложения:
* Label – поля для экранизации уточняющего текста
* TextBox – поле для ввода каких либо данных в программу
* Button – интерактивный объект для выполнения каких либо событий
* GroupBox – поле для группировки объектов внутри него
* Panel – тоже поле для группировки объектов, но меньше

**«Примеры работы приложения»**

1. Скриншоты ввода и вывода данных для различных вычислений
2. Пример работы общего задания:

## 

## Фотография 3.1

1. Пример работы индивидуального задания:

## 

## Фотография 3.2

**«Тестирование»**

1. Описание процесса тестирования приложения:

* Для начала протестируем основные функции приложения, например ввод данных и их корректная обработка.
* Потом протестируем остальные функции приложения, в нашем случае обработку пустых и некорректных полей.
* По возможности, проверим приложение на непредвиденные ошибки, например вылеты.

1. Результаты тестирования, включая выявленные ошибки (если есть) и способы их устранения:

## 

## Фотография 4.1

## 

## Фотография 4.2

## 

## Фотография 4.3

## 

## Фотография 4.4

**«Выводы»**

1. Сводка основных результатов работы:

* Программа работает корректно без каких либо ошибок.

1. Оценка выполненной задачи и ее соответствия поставленным целям:

* Задача выполнена верно и полностью соответствует поставленной задаче

1. Анализ достоинств и недостатков созданного приложения:

* Приложение четко выполняет поставленные задачи.
* Низкий функционал.

1. Рекомендации по дальнейшему развитию или улучшению приложения:

* Для дальнейшего развития лучше всего развиваться в количество функций. Например, добавить еще какое либо количество формул

**«Приложения»**

6.1. Исходный код приложения:

1. Код общего задания:

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

{

int price, usage, dist, res;

try

{

price = int.Parse(textBox1.Text);

usage = int.Parse(textBox2.Text);

dist = int.Parse(textBox3.Text);

if (price <= 0 || usage <= 0 || dist <= 0)

{

textBox4.Text = ("Числа должны быть больше нуля!");

return;

}

// Рассчитываем стоимость поездки

res = price \* (dist \* usage / 100);

textBox4.Text = $"Стоимость поездки: {res} руб.";

}

catch (FormatException)

{

textBox4.Text = ("Введите корректные числовые значения!");

}

catch (Exception ex)

{

textBox4.Text = ($"Ошибка: {ex.Message}");

}

}

1. Код индивидуального задания:

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

{

float weight, speed, height;

const float \_velocity = 9.8f;

if (string.IsNullOrWhiteSpace(textBox1.Text) ||

string.IsNullOrWhiteSpace(textBox2.Text) ||

string.IsNullOrWhiteSpace(textBox3.Text))

{

label10.Text = label11.Text = "Ошибка: Заполните все поля!";

return;

}

if (!float.TryParse(textBox1.Text, out weight) ||

!float.TryParse(textBox2.Text, out speed) ||

!float.TryParse(textBox3.Text, out height))

{

label10.Text = label11.Text = "Ошибка: Введите только числа!";

return;

}

double kinetic = 0.5 \* weight \* Math.Pow(speed, 2.0f);

double potential = weight \* \_velocity \* height;

label10.Text = $"{kinetic} Дж";

label11.Text = $"{potential} Дж";

}