

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

УТВЕРЖДАЮ
Зав.кафедрой,
доцент, к. ф.-м. н.

_____ С. В. Миронов

ОТЧЕТ О ПРАКТИКЕ

студентки 2 курса 211 группы факультета компьютерных наук и
информационных технологий

Никитенко Яны Валерьевны

вид практики: учебная

кафедра: информатики и программирования

курс: 2

семестр: 3

продолжительность: 18 нед., с 02.09.2024 г. по 12.01.2025 г.

Руководитель практики от университета,

доцент, к. ф.-м. н.

_____ М. И. Сафончик

Руководитель практики от организации (учреждения, предприятия),

доцент, к. ф.-м. н.

_____ М. И. Сафончик

Тема практики: «Разработка приложений Windows.Forms на языке C++ в среде Microsoft Visual Studio»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Приложение для вычисления факториала	5
2 Простые вычисления	10
3 Рекурсивные вычисления	16
4 Обработка табличных данных. Часть 1	23
5 Обработка табличных данных. Часть 2	28
6 Матричный калькулятор	34
7 Файловые диалоги и работа с файлами	45
8 Использование коллекций в Windows Forms	53
9 Приложение ТЕСТ	60
ЗАКЛЮЧЕНИЕ	67
Приложение А Примеры кода	68
Приложение Б CD-диск с отчетом о выполненной работе	81

ВВЕДЕНИЕ

Целью практики является освоение механизма построения оконного интерфейса приложений в среде Visual Studio. [?] В результате прохождения практики должны быть отработаны навыки

- создания нового проекта;
- добавления и настройки элементов управления;
- отладка корректного ввода данных для решения поставленной задачи;
- разработки алгоритма решения поставленной задачи с использованием оконного интерфейса;
- тестирования приложения;
- документирования разработанного кода.

1 Приложение для вычисления факториала

Задание. Реализовать приложение для вычисления факториала [?]

Создано окно приложения, содержащее два элемента TextBox, два элемента Label и один элемент Button [?]. Для отображения сообщений об ошибках в окно добавлен элемент ErrorProvider [?]. Вид окна представлен на рисунке 1.



Рисунок 1 – Окно приложения «Факториал» открытое в конструкторе

У элементов изменены значения некоторых атрибутов. Значения измененных атрибутов представлены в таблице 1.

Таблица 1 – Значения атрибутов элементов в приложении «Факториал»

Наименование атрибута	Значение
Для формы	
Text	Factorial
FormBorderStyle	Fixed3D
BackgroundImage	System.Drawing.Bitmap
MaximizeBox	False
Для первой надписи	
(Name)	lblInput
Text	N =
Для второй надписи	
(Name)	lblOutput
Text	N! =
Для первого текстового поля	
(Name)	txtInput
Для второго текстового поля	
(Name)	txtOutput
Для кнопки	
(Name)	btnCalculate
Text	Вычислить
Для обработчика ошибок	
(Name)	errorProvider

Была написана функция вычисления факториала:

```

1 #pragma once
2 long long fact(long long N) {
3     if (N < 0) //отрицательное число
4         return -1;
5     else if (N == 0 || N == 1) // 0! = 1
6         return 1;
7     else
8         return N * fact(N - 1); //n! = n * (n - 1)!
9 }
```

Здесь переменная N — число, для которого нужно вычислить факториал.

На нажатие кнопки «Вычислить» установлено выполнение следующего

кода:

```

1
2 private: System::Void btnCalculate_Click(System::Object^ sender, System::EventArgs^ e) {
3     ClearAll();
4     long long InputNumber;
```

```
5     bool result = Int64::TryParse(this->txtInput->Text, InputNumber); //переводим строку из TextBox в
    ↵     число
6
7     if (!result) //двели не число
8         errorProvider1->SetError(txtInput, "введено не целое число");
9     else {//число
10        if (InputNumber >= 20)
11            this->txtOutput->Text = "—лишком большое число";
12        else {
13            long long OutputNumber = fact(InputNumber); //результат
14            if (OutputNumber == -1) //отрицательное число
15                errorProvider1->SetError(txtInput, "введено отрицательное число");
16            else //все нормально
17                this->txtOutput->Text = System::Convert::ToString(OutputNumber); //записываем в поле
18
19        }
20
21    }
22
23 }
24
```

После запуска приложения на экране появляется окно (см. рисунок 2).

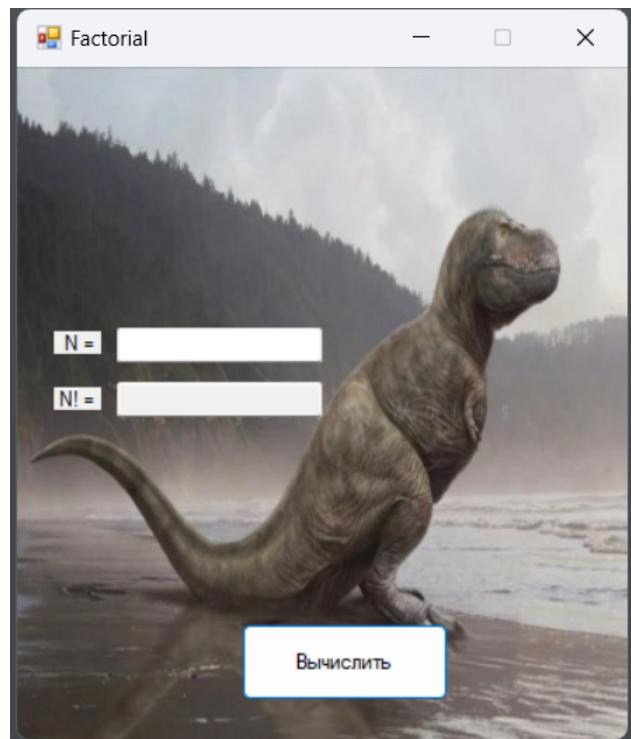


Рисунок 2 – Окно приложения «Факториал»: начальный запуск

При вводе целого числа после нажатия кнопки в поле вывода приводится результат вычисления факториала для заданного числа (см. рисунок 3).

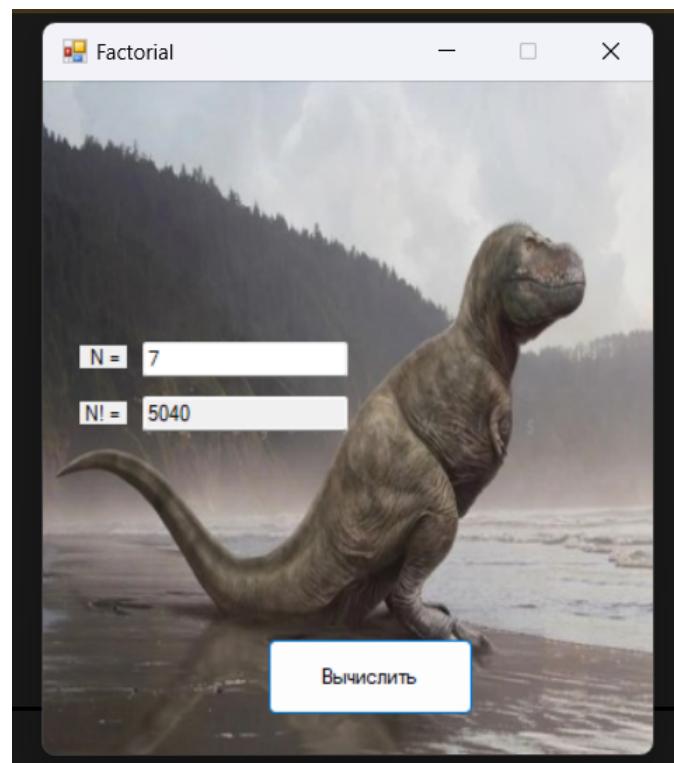


Рисунок 3 – Окно приложения «Факториал»: корректное вычисление

При вводе некорректного значения возникает сообщение об ошибке (см.

рисунок 4).



Рисунок 4 – Окно приложения «Факториал»: сообщение о некорректном вводе

Полный код программы приведен в приложении Б.

2 Простые вычисления

Задание. Разработать приложение для вычисления значения функции. Создано окно приложения, содержащее три элемента TextBox, три элемента Label и один элемент Button. Для отображения сообщений об ошибках в окно добавлены два элемента ErrorProvider. Вид окна представлен на рисунке 5.

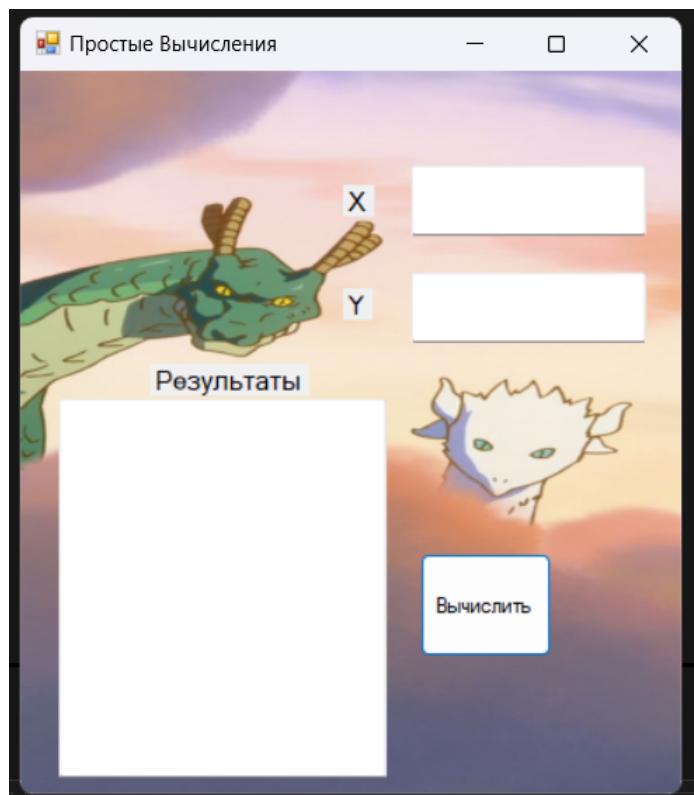


Рисунок 5 – Окно приложения «Простые вычисления №12» открытое в конструкторе

У элементов изменены значения некоторых атрибутов. Значения измененных атрибутов представлены в таблице 2.

Таблица 2 – Значения атрибутов элементов в приложении «Простые вычисления №12»

Наименование атрибута	Значение
Для формы	
Text	Простые вычисления №7
BackgroundImage	System.Drawing.Bitmap
Для первой надписи	
Text	X =
Для второй надписи	
Text	Y =
Для третьей надписи	
Text	Результаты
Для первого текстового поля	
(Name)	txtInputX
Для второго текстового поля	
(Name)	txtInputY
Для третьего текстового поля	
(Name)	txtOutput
Для кнопки	
(Name)	Calculate
Text	Вычислить

На нажатие кнопки «Вычислить» установлено выполнение следующего кода:

```

1
2 private: System::Void Calculate_Click(System::Object^ sender, System::EventArgs^ e) {
3     ClearAll();
4
5     double InputNumberX, InputNumberY;
6     bool hasError = false; // Флаг для ошибки
7
8     // Проверка
9     bool resultX = Double::TryParse(this->txtInputX->Text, InputNumberX);
10    if (!resultX) { // Если не число
11        errorProvider1->SetError(txtInputX, "введено не целое число");
12        hasError = true; // Установка флага
13    }
14    else if (InputNumberX < 0) { // Проверка на отрицательное число
15        errorProvider1->SetError(txtInputX, "введено отрицательное число");
16        hasError = true; // Установка флага
17    }
18
19    // Проверяем второе число

```

```

20 bool resultY = Double::TryParse(this->txtInputY->Text, InputNumberY);
21 if (!resultY) { // Если не число
22     errorProvider2->SetError(txtInputY, "введено не целое число");
23     hasError = true; // // Установка флага
24 }
25 else if (InputNumberY < 0) { // // Проверка на отрицательное число
26     errorProvider2->SetError(txtInputY, "введено отрицательное число");
27     hasError = true; // // Установка флага
28 }
29
30 // Если есть ошибки, то гг :0
31 if (hasError) {
32     return;
33 }
34
35
36 // Результат
37 double OutputNumber = Simple(InputNumberX, InputNumberY);
38 this->txtOutput->Text = System::Convert::ToString(OutputNumber); // Записываем в поле
39
40 }
41

```

После запуска приложения на экране появляется окно (см. рисунок 6).

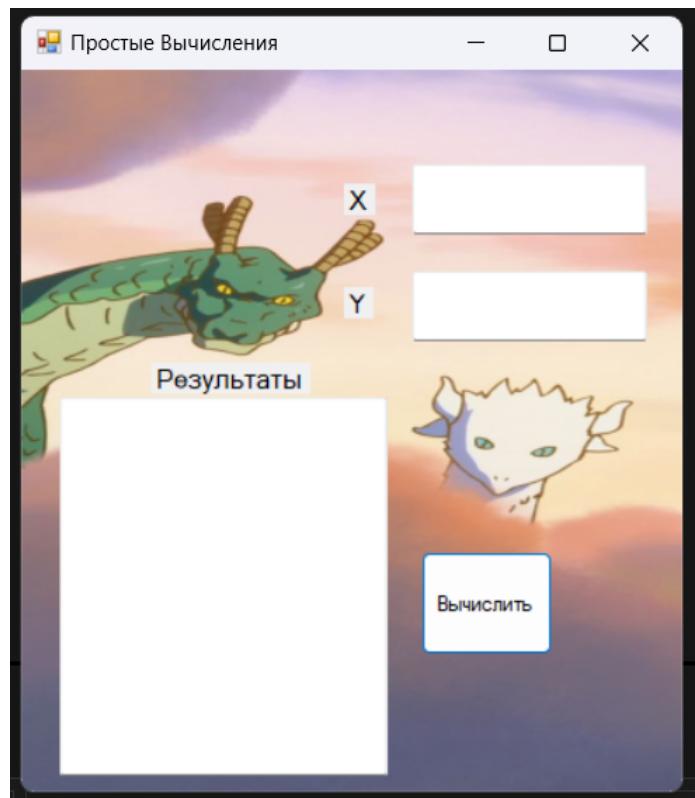


Рисунок 6 – Окно приложения «Простые вычисления №12»: начальный запуск

При вводе целых чисел после нажатия кнопки в поле вывода приводится результат вычисления функции для заданных чисел (см. рисунок 7).

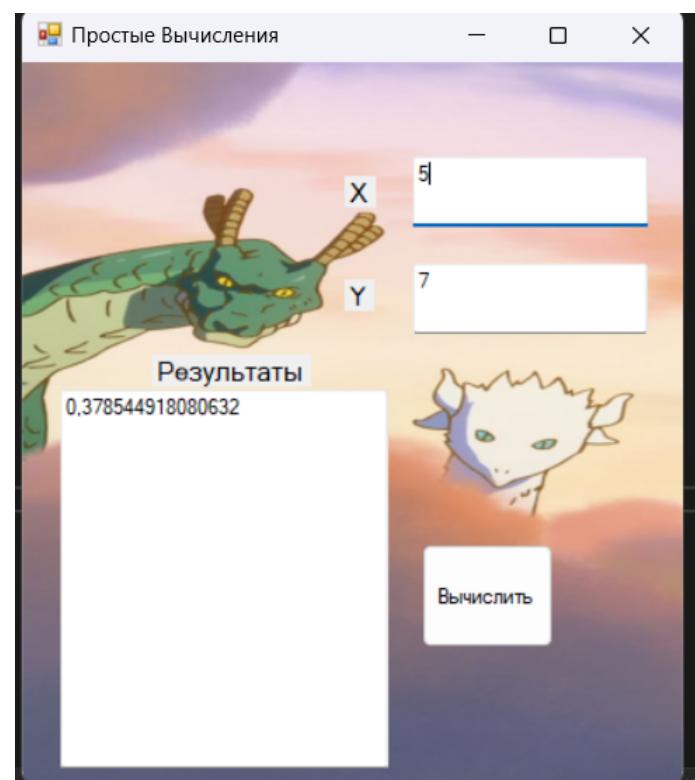


Рисунок 7 – Окно приложения «Простые вычисления №12»: корректное вычисление

При вводе некорректного значения возникает сообщение об ошибке (см. рисунки 8 и 9).

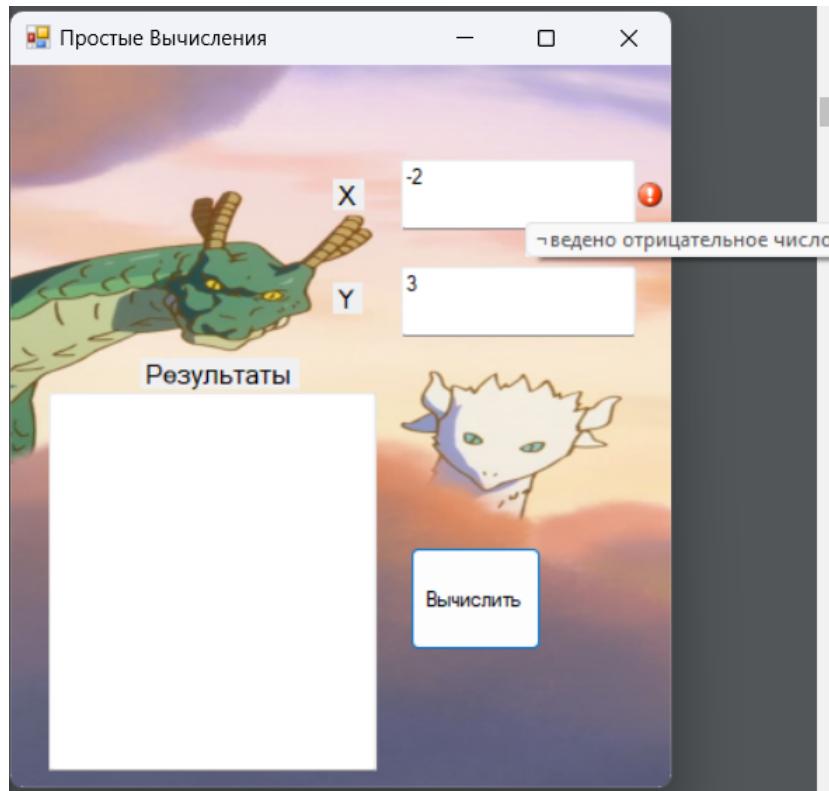


Рисунок 8 – Окно приложения «Простые вычисления №12»: сообщение о некорректном вводе
«Введено отрицательное число»)

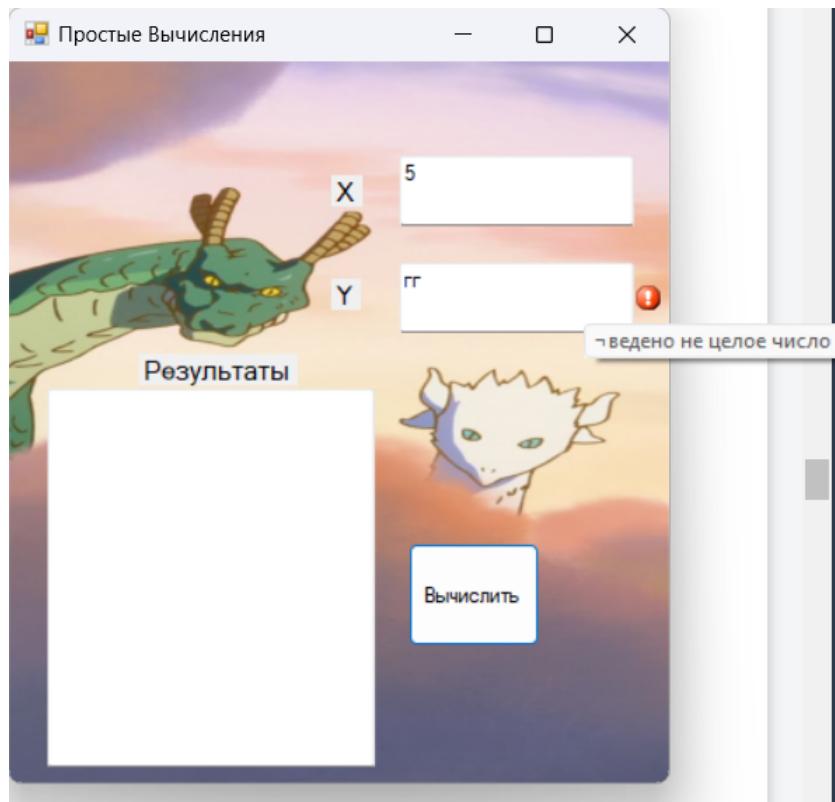


Рисунок 9 – Окно приложения «Простые вычисления №12»: сообщение о некорректном вводе
«Введено не число»)

Полный код программы приведен в приложении **Б**.

3 Рекурсивные вычисления

Задание. Разработать приложение для вычисления суммы цифр числа. Создано окно приложения, содержащее два элемента TextBox, два элемента Label и один элемент Button. Для отображения сообщений об ошибках в окно добавлен элемент ErrorProvider. Вид окна представлен на рисунке 10.



Рисунок 10 – Окно приложения «RecurFunc3» открытое в конструкторе

элементов изменены значения некоторых атрибутов. Значения измененных атрибутов представлены в таблице 3.

Таблица 3 – Значения атрибутов элементов в приложении «Рекурсия»

Наименование атрибута	Значение
Для формы	
Text	Рекурсия
FormBorderStyle	Fixed3D
Для первой надписи	
(Name)	lblInput
Text	N =
Для второй надписи	
(Name)	lblOutput
Text	Результат
Для первого текстового поля	
(Name)	txtInput
Для второго текстового поля	
(Name)	txtOutput
Для кнопки	
(Name)	btnCalc
Text	Вычислить

На нажатие кнопки «Вычислить» установлено выполнение следующего кода:

```

1
2 private: System::Void ButtonResult_Click(System::Object^ sender, System::EventArgs^ e) {
3     ClearAll();
4     long long InputNumber;
5     bool result = Int64::TryParse(this->InputForN->Text, InputNumber); //перевод строки из TextBox в
6     // число
7
8     if (!result) { // Введено не число
9         errorProvider1->SetError(InputForN, "Введено не целое число");
10        return; // Выход из функции, чтобы избежать дальнейших проверок
11    }
12
13    if (InputNumber < 0) { // Проверка на отрицательное число
14        errorProvider1->SetError(InputForN, "Введено отрицательное число");
15        return; // Выход из функции
16    }
17
18    if (InputNumber == 0) { // Проверка на отрицательное число
19        errorProvider1->SetError(InputForN, "Введен ноль");
20        return; // Выходим из функции
21    }

```

```
22 if (InputNumber >= 127) {  
23     errorProvider1->SetError(InputForN, "Переполнение стека");  
24     return;  
25 }  
26  
27  
28 long long OutputNumber = Recur(InputNumber);  
29  
30 // Запись в поле  
31 this->OutPutForN->Text = System::Convert::ToString(OutputNumber);  
32 }  
33
```

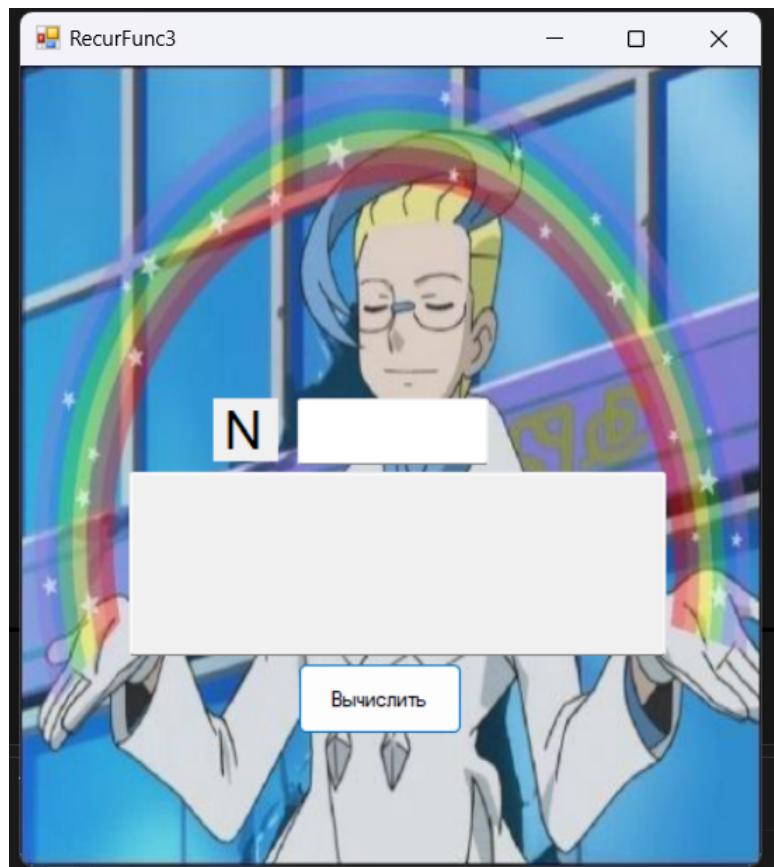


Рисунок 11 – Окно приложения «RecurFunc3»: начальный запуск



Рисунок 12 – Окно приложения «RecurFunc3»: корректное вычисление

При вводе слишком большого значения для вычислений, появляется сообщение о том что происходит переполнение стека **13**.

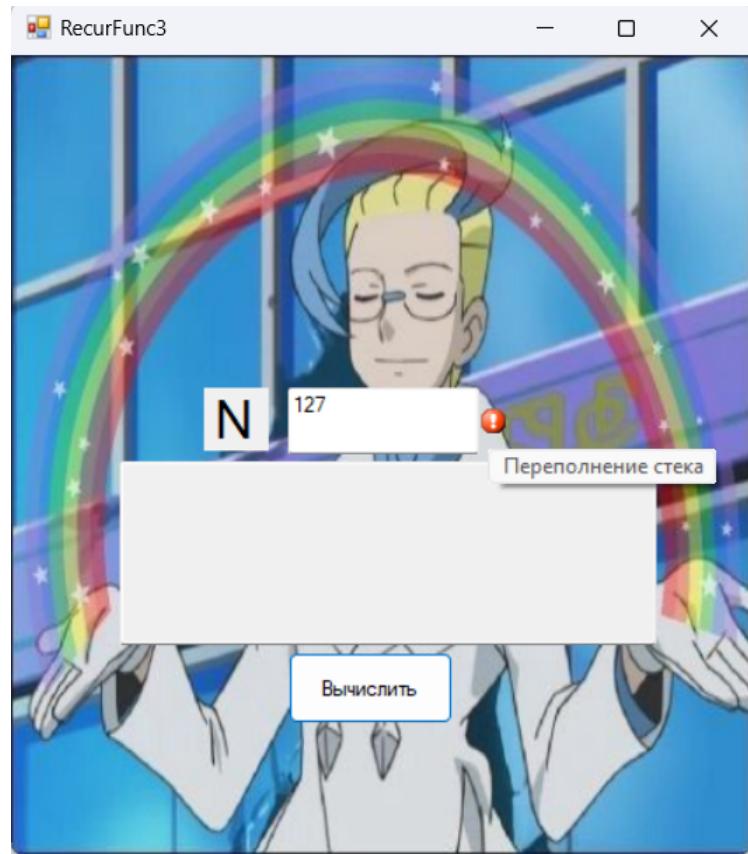


Рисунок 13 – Окно приложения «RecurFunc3»: некорректное вычисление (введение слишком большого значения для вычислений)

При вводе некорректного значения возникает сообщение об ошибке (см. рисунки 14 и 15).

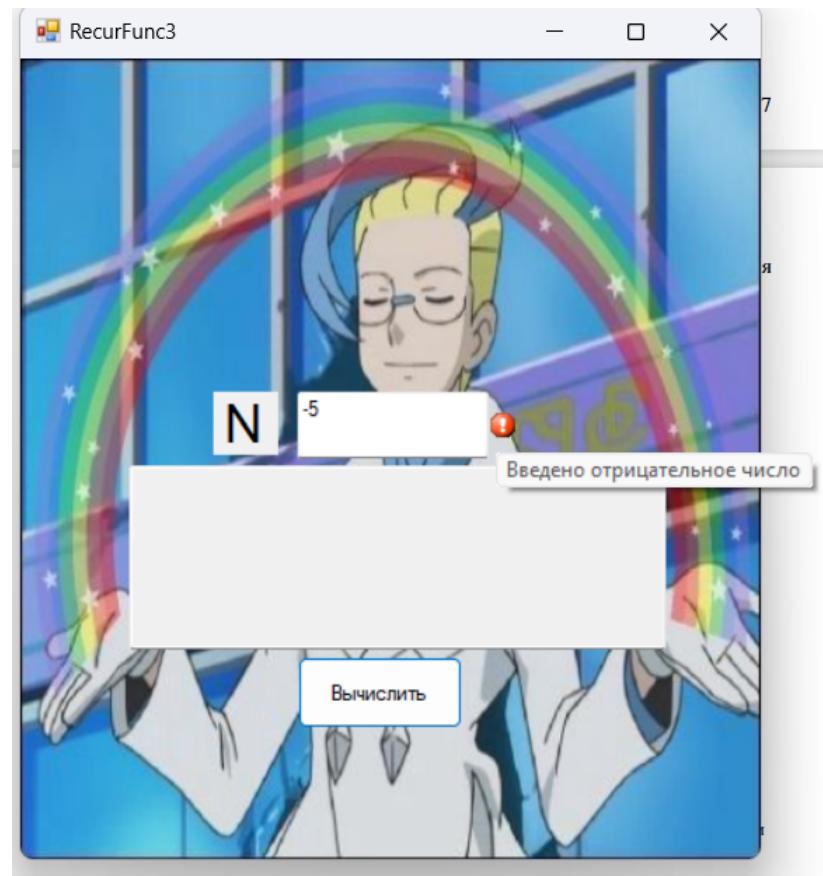


Рисунок 14 – Окно приложения «RecurFunc3»: некорректное вычисление (введение отрицательного числа)

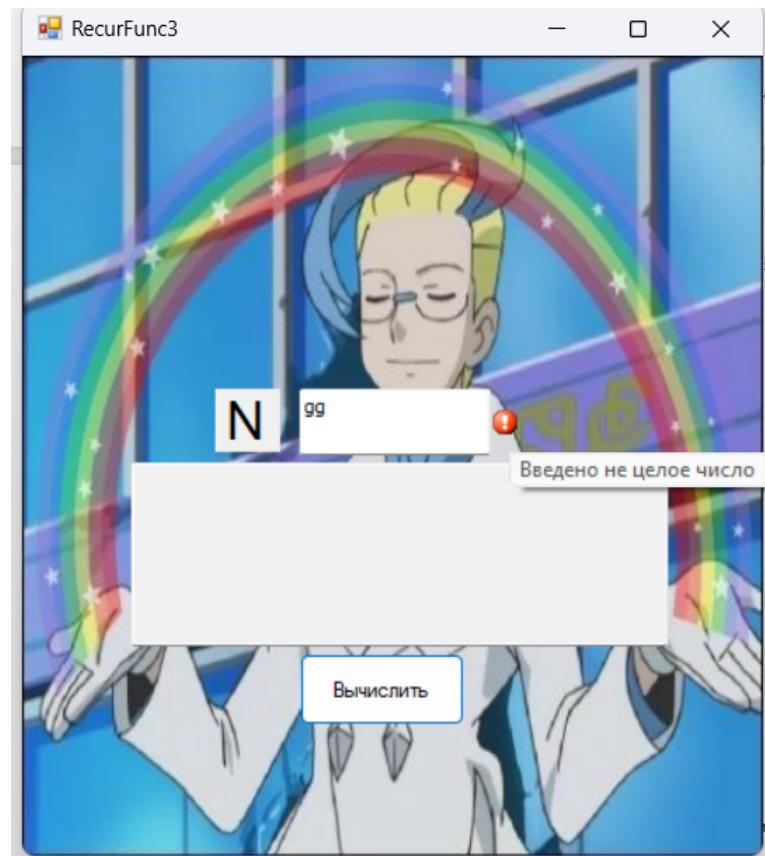


Рисунок 15 – Окно приложения «RecurFunc3»: некорректное вычисление (введение не целого числа)

Полный код программы приведен в приложении **Б**.

4 Обработка табличных данных. Часть 1

Задание. Разработать приложение для работы с одномерным массивом.

Создано окно приложения, содержащее четыре элемента TextBox, пять элементов Label, четыре элемента Button и один элемент DataGridView [?]. Для отображения сообщений об ошибках в окно добавлены четырьес элемента ErrorProvider. Вид окна представлен на рисунке 16.

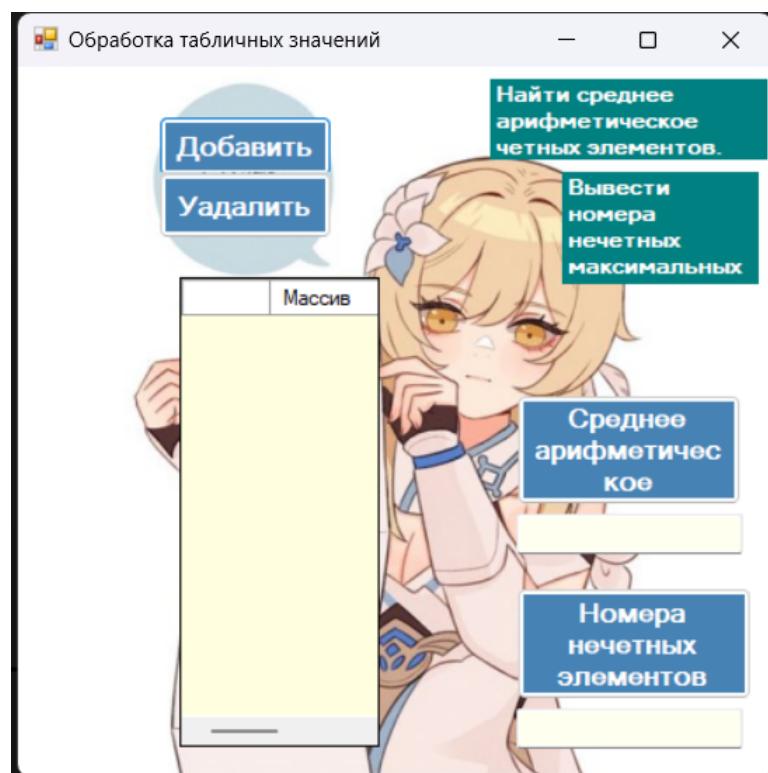


Рисунок 16 – Окно приложения «Обработка табличных значений»:открытое в конструкторе

У элементов изменены значения некоторых атрибутов. Значения измененных атрибутов представлены в таблице 4.

Таблица 4 – Значения атрибутов элементов в приложении «Работа с таблицей»

Наименование атрибута	Значение
Для формы	
Text	Обработка табличных данных
Для первой надписи	
Text	Найти среднее арифметическое четных элементов,
Для второй надписи	
Text	Вывести номера нечетных элементов,
Для первого текстового поля	
(Name)	SrResult
Для второго текстового поля	
(Name)	NomeraResult
Для первой кнопки	
(Name)	Add
Text	Добавить
Для второй кнопки	
(Name)	Delete
Text	Удалить
Для третьей кнопки	
(Name)	SredneeArifm
Text	Среднее арифмитическое
Для четвертой кнопки	
(Name)	NechotElem
Text	Номера нечетных элементов
Для таблицы	
(TextColumn)	Массив

На нажатие кнопки «Добавить» установлено выполнение следующего кода:

```

1 this->grdMassiv->Rows->Add(1);
2 if (this->grdMassiv->Rows->Count > 0) {
3     errorProvider3->SetError(grdMassiv, ""); // Очищаем сообщение об ошибке
4 }
```

На нажатие кнопки «Удалить» установлено выполнение следующего кода:

```

1 // Проверка на наличие строк в grdMassiv
2 if (this->grdMassiv->Rows->Count == 0) {
3     errorProvider3->SetError(grdMassiv, "Тут не строк чтоб их удалить"); // Установка ошибки, если нет
4     ← строк
5     return; // Выход из метода
6 }
```

```

6
7
8
9 // Удаление
10 if (!this->grdMassiv->CurrentRow->IsNewRow) {
11     int i = this->grdMassiv->CurrentRow->Index;
12     this->grdMassiv->Rows->Remove(this->grdMassiv->Rows[i]);
13 }

```

Примеры остальных кодов приведены в приложении А.

После запуска приложения на экране появляется окно (см. рисунок 17).

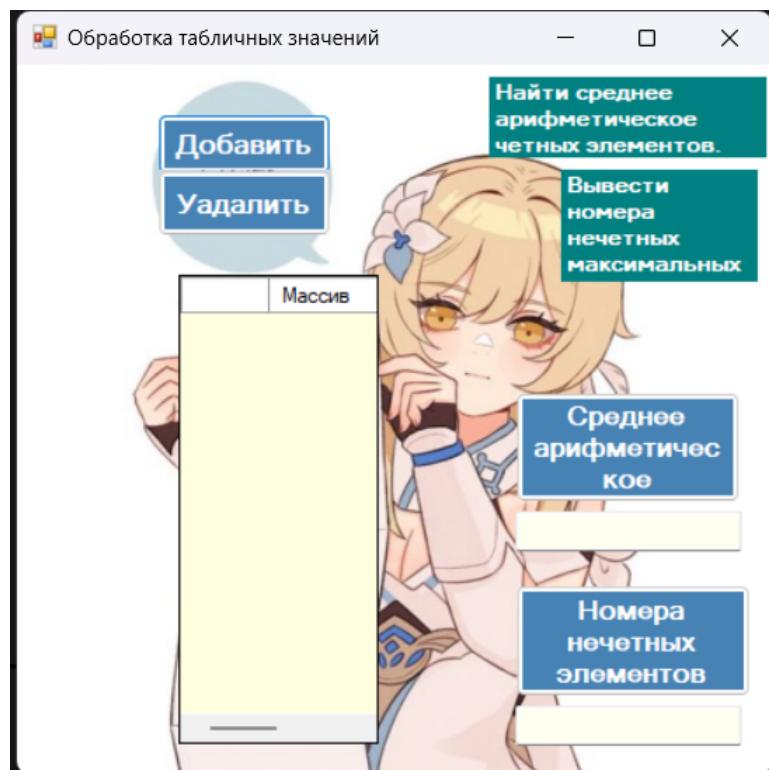


Рисунок 17 – Окно приложения «Обработка табличных значений»: начальный запуск

При вводе целых чисел в таблицу и нажатия кнопки «Среднее арифметическое» в соответствующем поле вывода приводится результат вычисления среднего четных элементов. При вводе целых чисел в таблицу после нажатия кнопки «Номера нечетных элементов» в соответствующем поле вывода приводится результат вычисления номеров максимальных нечетных элементов (см. рисунок 18)

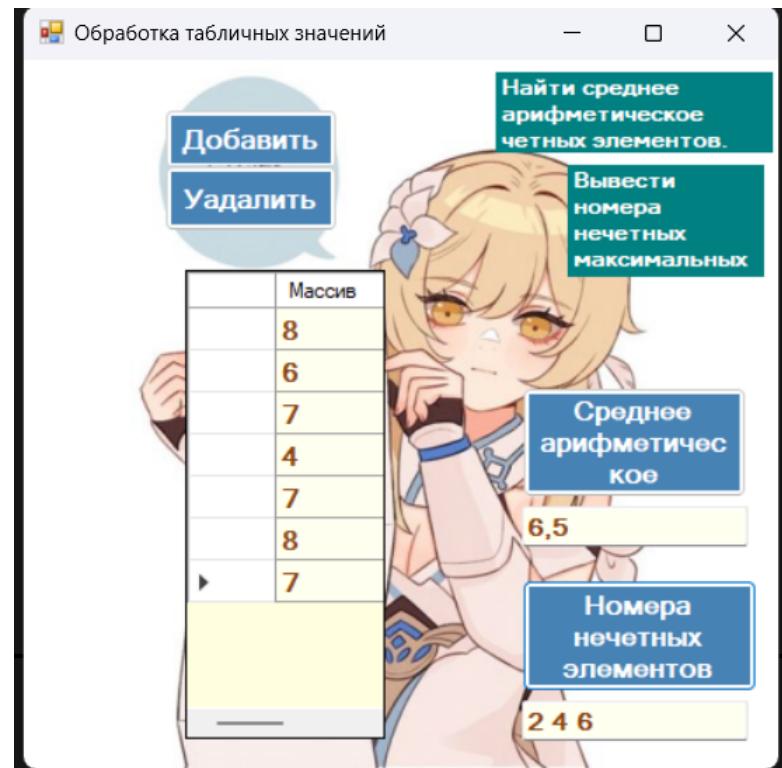


Рисунок 18 – Окно приложения «Обработка табличных значений»: корректное вычисление

При вводе некорректного значения возникает сообщение об ошибке (см. рисунки 19 и 20).

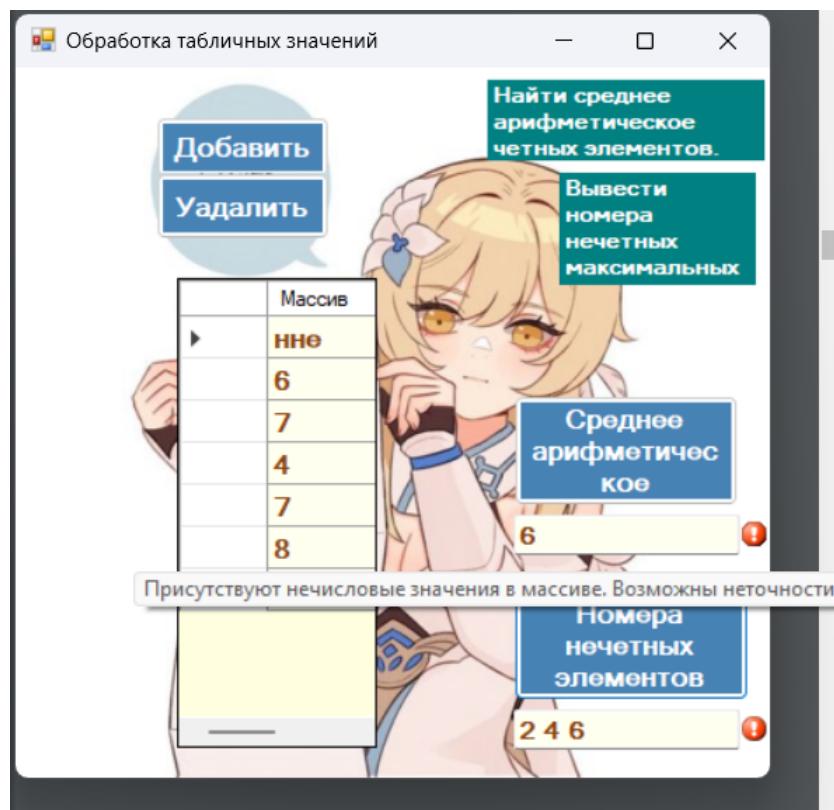


Рисунок 19 – Окно приложения «Обработка табличных значений»: введение не числа



Рисунок 20 – Окно приложения «Обработка табличных значений»: отсутствие среднего арифметического и номеров нечетных элементов

Полный код программы приведен в приложении Б.

5 Обработка табличных данных. Часть 2

Задание. Разработать приложение для работы с двумерным массивом. Создано окно приложения, два элемента Label, пять элементов Button и два элемента DataGridView, один элемент PictureBox. Для отображения сообщений об ошибках в окно добавлены два элемента ErrorProvider. Вид окна представлен на рисунке 21.

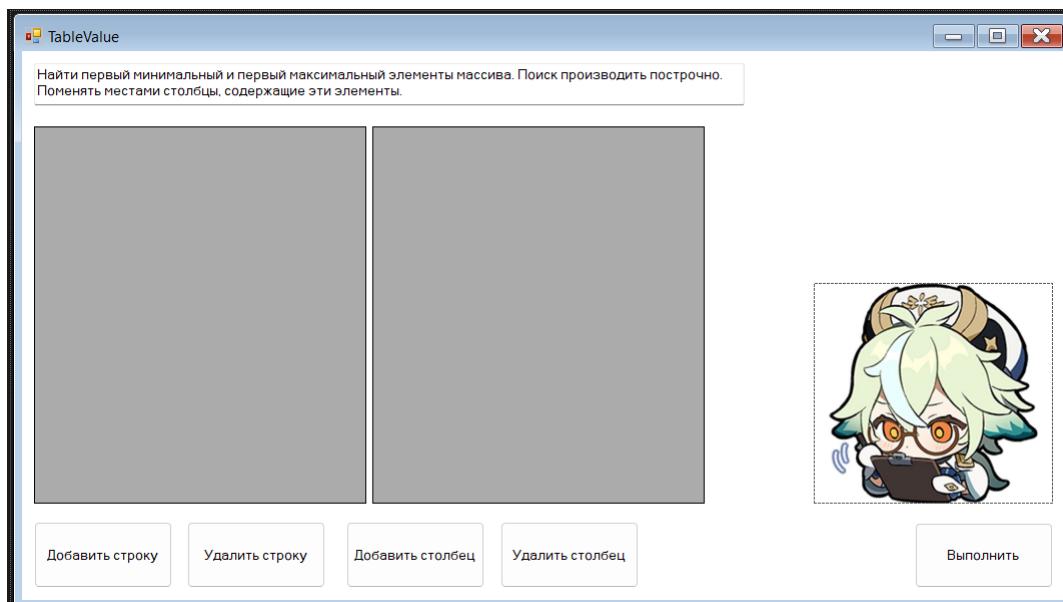


Рисунок 21 – Окно приложения «TableValue» открытое в конструкторе)

У элементов изменены значения некоторых атрибутов. Значения измененных атрибутов представлены в таблице 5

Таблица 5 – Значения атрибутов элементов в приложении «TableValue»

Наименование атрибута	Значение
Для формы	
Text	TableValue
Для первой надписи	
(Name)	Task
Text	Найти первый минимальный и первый максимальный элементы массива. Поиск производить построчно. Поменять местами столбцы, содержащие эти элементы.
Для первой кнопки	
(Name)	AddString
Text	Добавить строку
Для второй кнопки	
(Name)	DeleteString
Text	Удалить строку
Для третьей кнопки	
(Name)	AddColumn
Text	Добавить столбец
Для четвертой кнопки	
(Name)	DeleteColumn
Text	Удалить столбец
Для пятой кнопки	
(Name)	Execute
Text	Выполнить
Для первой таблицы	
(Name)	DataMassivInPut
Для второй таблицы	
(Name)	DataMassivOutPut

На нажатие кнопки «Добавить строку» установлено выполнение следующего кода:

```

1 // Добавление строк
2 private: System::Void AddString_Click(System::Object^ sender, System::EventArgs^ e) {
3
4 // Проверка на наличие столбцов
5 if (this->DataMassivInPut->Columns->Count == 0) {
6     errorProvider1->SetError(DataMassivInPut, "Столбцы должны быть добавлены перед добавлением
    ↳ строк.");
7     return;
8 }
```

```

9
10 this->DataMassivInPut->Rows->Add(1);
11 this->DataMassivOutPut->Rows->Add(1);
12 if (this->DataMassivInPut->Rows->Count > 0 && this->DataMassivOutPut->Rows->Count > 0) {
13     errorProvider1->SetError(DataMassivInPut, "");
14 }
15
16 }

```

На нажатие кнопки «Удалить строку» установлено выполнение следующего кода:

```

1 // Удаление строк
2 private: System::Void DeleteString_Click(System::Object^ sender, System::EventArgs^ e) {
3
4
5 // Проверка на наличие строк
6 if (this->DataMassivInPut->Rows->Count == 0) {
7     errorProvider1->SetError(DataMassivInPut, "Что бы удалить строку, нужно в начале добавить ее ");
8     ↵ // Установка ошибки, если нет строк
9     return;
10 }
11
12 // Удаление
13 if (!this->DataMassivInPut->CurrentRow->IsNewRow &&
14     ↵ !this->DataMassivOutPut->CurrentRow->IsNewRow) {
15     int i = this->DataMassivInPut->CurrentRow->Index;
16     this->DataMassivInPut->Rows->Remove(this->DataMassivInPut->Rows[i]);
17
18     int o = this->DataMassivOutPut->CurrentRow->Index;
19     this->DataMassivOutPut->Rows->Remove(this->DataMassivOutPut->Rows[o]);
20 }

```

На нажатие кнопки «Добавить столбец» установлено выполнение следующего кода:

```

1 // Добавление столбцов
2 private: System::Void AddColumn_Click(System::Object^ sender, System::EventArgs^ e) {
3
4     this->DataMassivInPut->Columns->Add("COLUMN", " ");
5     this->DataMassivOutPut->Columns->Add("COLUMN", " ");
6     if (this->DataMassivInPut->Columns->Count > 0 && this->DataMassivOutPut->Columns->Count > 0) {
7         errorProvider1->SetError(DataMassivInPut, "");
8     }
9     for (int i = 0; i < this->DataMassivInPut->Columns->Count; i++) {

```

```

10    this->DataMassivInPut->Columns[i]->Width =
    ↵      System::Convert::ToInt32(this->DataMassivInPut->Width / (1.25 *
    ↵      this->DataMassivInPut->Columns->Count));
11  }
12  for (int i = 0; i < this->DataMassivOutPut->Columns->Count; i++) {
13    this->DataMassivOutPut->Columns[i]->Width =
    ↵      System::Convert::ToInt32(this->DataMassivOutPut->Width / (1.25 *
    ↵      this->DataMassivOutPut->Columns->Count));
14  }
15 }
```

На нажатие кнопки «Удалить столбец» установлено выполнение следующего кода:

```

1 // Удаление столбцов
2 private: System::Void DeleteColumn_Click(System::Object^ sender, System::EventArgs^ e) {
3
4 // Проверка на наличие строк
5 if (this->DataMassivInPut->Columns->Count == 0) {
6   errorProvider1->SetError(DataMassivInPut, "Что бы удалить столбец, нужно в начале добавить его
    ↵  ");
7   return; // Выход из метода
8 }
9
10
11 // Проверяем, есть ли столбцы в DataGridView
12 if (this->DataMassivInPut->Columns->Count > 0) {
13   // Получение индекса последнего столбца
14   int i = this->DataMassivInPut->Columns->Count - 1; // Индекс последнего столбца
15   int o = this->DataMassivOutPut->Columns->Count - 1;
16
17   // Удаление
18   this->DataMassivInPut->Columns->RemoveAt(i);
19   this->DataMassivOutPut->Columns->RemoveAt(o);
20 }
21
```

Примеры остальных кодов приведены в приложении А.

После запуска приложения на экране появляется окно (см. рисунок 22).



Рисунок 22 – Окно приложения «TableValue» открытое в конструкторе

При вводе целых чисел в первую таблицу после нажатия кнопки «Выполнить» во второй таблице приводится результат замены столбцов содержащих первый минимальный и первый максимальный элементы массива (см. рисунок 23).

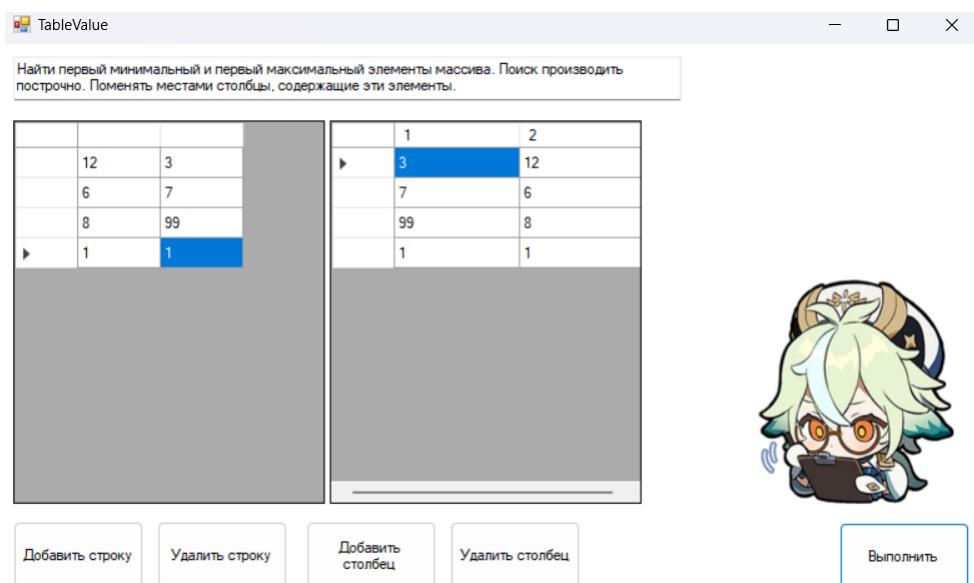


Рисунок 23 – Окно приложения «TableValue»: корректное вычисление

При вводе некорректного значения возникает сообщение об ошибке (см. рисунки 24 и 25).

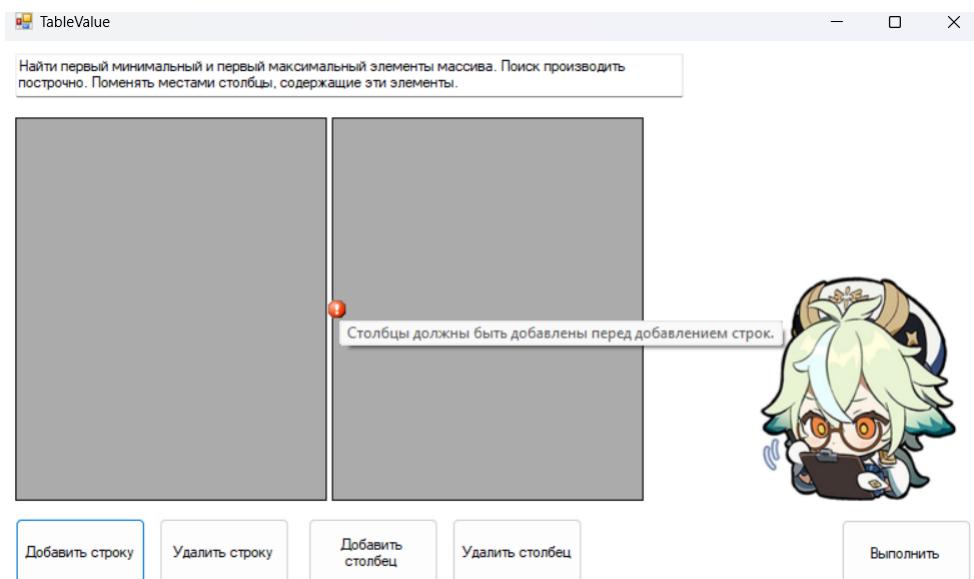


Рисунок 24 – Окно приложения «TableValue» некорректный ввод (Столбцы должны быть добавлены перед добавлением строк.)

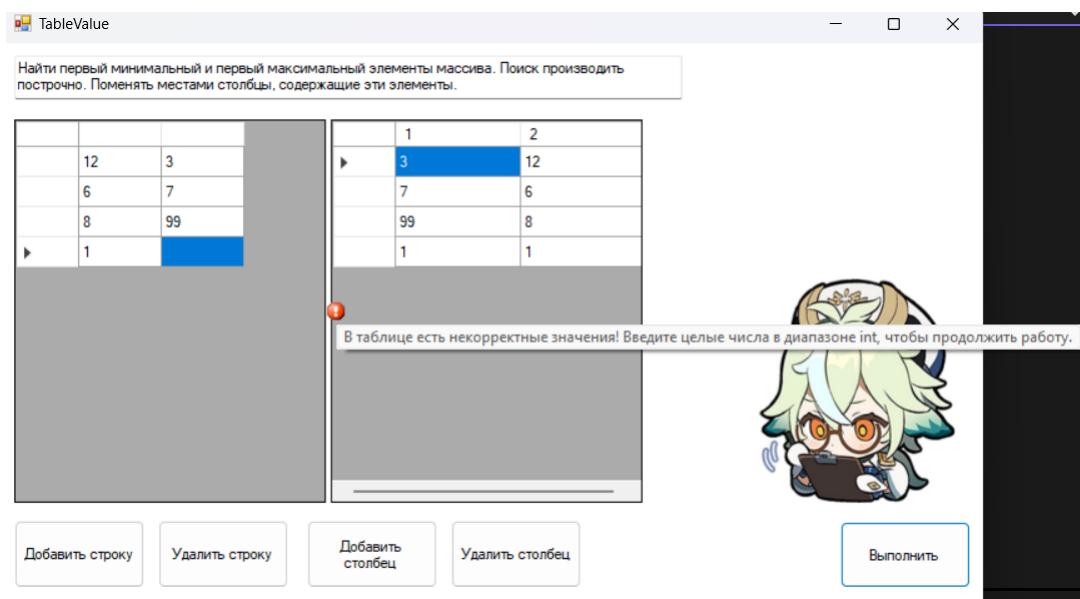


Рисунок 25 – Окно приложения «TableValue» некорректный ввод (В таблице есть некорректные значения! Введите целые числа в диапазоне int, чтобы продолжить работу.)

Полный код программы приведен в приложении **Б**.

6 Матричный калькулятор

Задание. Разработать приложение «Матричный калькулятор» [?].

Создано окно приложения, содержащее три элемента TextBox, семь элементов Label, семнадцать элементов Button, пять элементов DataGridView, один элемент PictureBox, один элемент GroupBox и четырнадцать элементов RadioButton. Для отображения сообщений об ошибках в окно добавлены восемь элементов ErrorProvider. Вид окна представлен на рисунке 26.

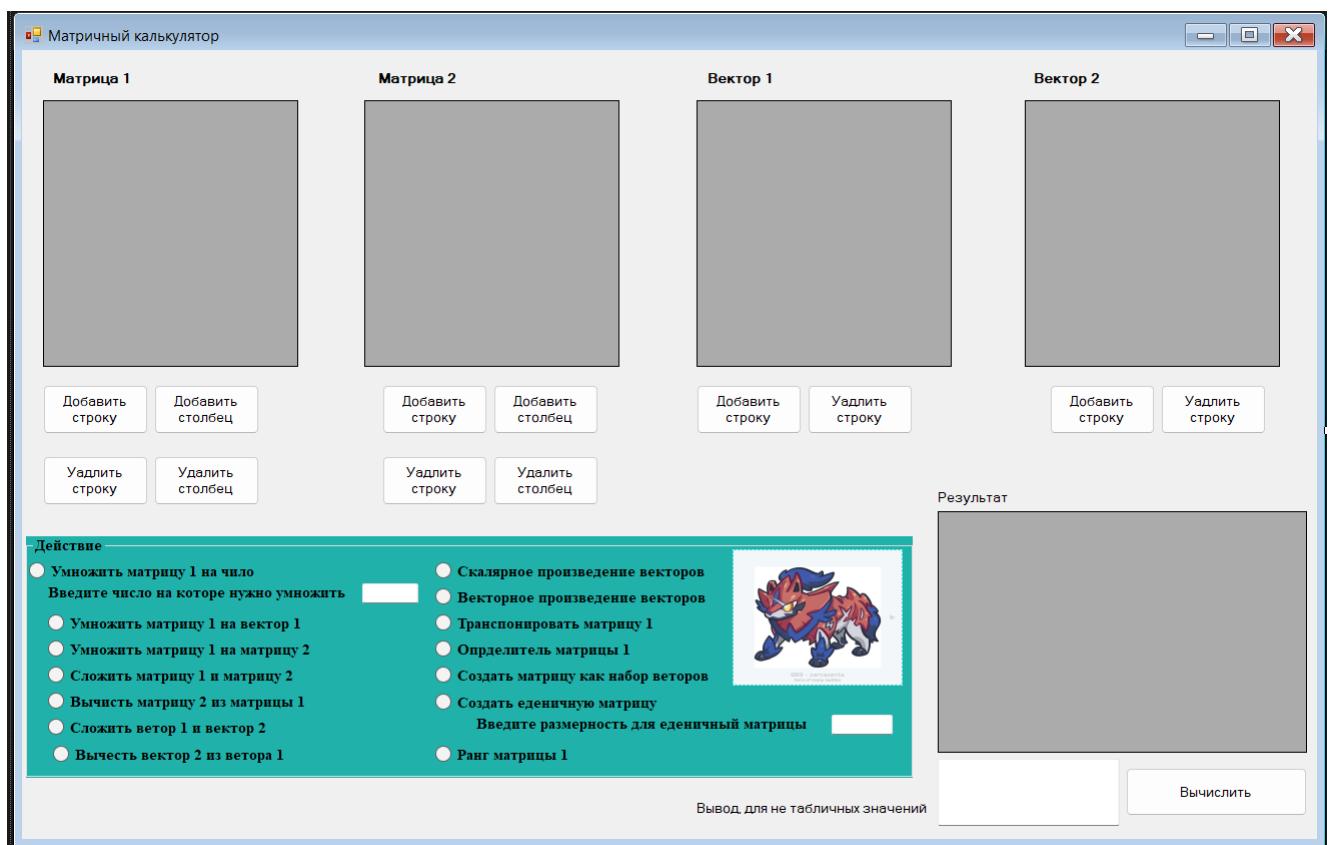


Рисунок 26 – Окно приложения «Матричный калькулятор» открытое в конструкторе

У элементов изменены значения некоторых атрибутов. Значения измененных атрибутов представлены в таблицах 6, 7, 8 и 9.

Таблица 6 – Значения атрибутов элементов в приложении «Матричный калькулятор»

Наименование атрибута	Значение
Для формы	
Text	Матричный калькулятор
Для первой надписи	
Text	Матрица 1
Для второй надписи	
Text	Матрица 2

Таблица 7 – Значения атрибутов элементов в приложении «Матричный калькулятор»

Наименование атрибута	Значение
Для третьей надписи	
Text	Вектор 1
Для четвертой надписи	
Text	Вектор 2
Для пятой надписи	
Text	Введите размерерность для единичной матрицы:
Для шестой надписи	
Text	Результат
Для седьмой надписи	
Text	Вывод, для не табличных значений
Для первого текстового поля	
(Name)	InPutNumber
Для второго текстового поля	
(Name)	MatrixSize
Для третьего текстового поля	
(Name)	ResultText
Для первой кнопки	
(Name)	AddString
Text	Добавить строку
Для второй кнопки	
(Name)	DeleteString
Text	Удалить строку
Для третьей кнопки	
(Name)	AddColumn
Text	Добавить столбец
Для четвертой кнопки	
(Name)	DeleteColumn
Text	Удалить столбец
Для пятой кнопки	
(Name)	AddString2
Text	Добавить строку
Для шестой кнопки	
(Name)	DeleteString2
Text	Удалить строку
Для седьмой кнопки	
(Name)	AddColumn2
Text	Добавить столбец
Для восьмой кнопки	
(Name)	DeleteColumn2
Text	Удалить столбец

Таблица 8 – Значения атрибутов элементов в приложении «Матричный калькулятор»

Наименование атрибута	Значение
Для девятой кнопки	
(Name)	AddStringV
Text	Добавить строку
Для десятой кнопки	
(Name)	DeleteStringV
Text	Удалить строку
Для одиннадцатой кнопки	
(Name)	AddStringV2
Text	Добавить строку
Для двенадцатой кнопки	
(Name)	DeleteStringV2
Text	Удалить строку
Для тринадцатой кнопки	
(Name)	btnCalc
Text	Execute
Для первой таблицы	
(Name)	Matrix1
Для второй таблицы	
(Name)	Matrix2
Для третьей таблицы	
(Name)	Vector1
Для четвертой таблицы	
(Name)	Vector2
Для пятой таблицы	
(Name)	Result
Для группового окна	
Text	Действия

Таблица 9 – Значения атрибутов элементов в приложении «Матричный калькулятор»

Наименование атрибута	Значение
Для первого переключателя	
(Name)	multX
Text	Умножить первую матрицу на число
Для второго переключателя	
(Name)	multVect
Text	Умножить первую матрицу на первый вектор
Для третьего переключателя	
(Name)	multMatr
Text	Умножить первую матрицу на вторую матрицу
Для четвертого переключателя	
(Name)	plusMatr
Text	Сложить две матрицы
Для пятого переключателя	
(Name)	minusMatr
Text	Вычесть из первой матрицы вторую матрицу
Для шестого переключателя	
(Name)	plusVect
Text	Сложить два вектора
Для седьмого переключателя	
(Name)	minusVect
Text	Вычесть из первого вектора второй вектор
Для восьмого переключателя	
(Name)	skalMultVect
Text	Скалярное произведение двух векторов
Для девятого переключателя	
(Name)	vectMultVect
Text	Векторное произведение двух векторов
Для десятого переключателя	
(Name)	transMatr
Text	Транспонирование первой матрицы
Для одиннадцатого переключателя	
(Name)	OpredRangMatr
Text	Определитель
Для двенадцатого переключателя	
(Name)	createMatrVect
Text	Создать матрицу как набор векторов
Для тринадцатого переключателя	
(Name)	edinMatr
Text	Создать единичную матрицу

На нажатие кнопки «Добавить строку» установлено выполнение следующего кода:

```
1 private: System::Void AddString_Click(System::Object^ sender, System::EventArgs^ e) {  
2  
3     // Проверка на наличие столбцов  
4     if (this->Matrix1->Columns->Count == 0) {  
5         errorM1->SetError(Matrix1, "Столбцы должны быть добавлены перед добавлением строк.");  
6         return;  
7     }  
8     clearMatr1();  
9     this->Matrix1->Rows->Add(1);  
10  
11    if (this->Matrix1->Rows->Count > 0) {  
12        errorM1->SetError(Matrix1, "");  
13    }  
14  
15 }
```

На нажатие кнопки «Удалить строку» установлено выполнение следующего кода:

```
1 private: System::Void DeleteString_Click(System::Object^ sender, System::EventArgs^ e) {  
2  
3  
4     // Проверка на наличие строк  
5     if (this->Matrix1->Rows->Count == 0) {  
6         errorM1->SetError(Matrix1, "Что бы удалить строку, нужно в начале добавить ее"); // Установка  
    ↳ ошибки, если нет строк  
7         return;  
8     }  
9     clearMatr1();  
10    // Удаление  
11    if (!this->Matrix1->CurrentRow->IsNewRow) {  
12        int i = this->Matrix1->CurrentRow->Index;  
13        this->Matrix1->Rows->Remove(this->Matrix1->Rows[i]);  
14  
15    }  
16  
17 }
```

На нажатие кнопки «Добавить столбец» установлено выполнение следующего кода:

```
1 private: System::Void AddColumn_Click(System::Object^ sender, System::EventArgs^ e) {  
2     this->Matrix1->Columns->Add("COLUMN", " ");  
3  
4     if (this->Matrix1->Columns->Count > 0) {
```

```

5     errorM1->SetError(Matrix1, "");
6 }
7 for (int i = 0; i < this->Matrix1->Columns->Count; i++) {
8     this->Matrix1->Columns[i]->Width = System::Convert::ToInt32(this->Matrix1->Width / (1.25 *
9         → this->Matrix1->Columns->Count));
10    }
11
12 }

```

На нажатие кнопки «Удалить столбец» установлено выполнение следующего кода:

```

1 private: System::Void DeleteColumn_Click(System::Object^ sender, System::EventArgs^ e) {
2     // Проверка на наличие строк
3     if (this->Matrix1->Columns->Count == 0) {
4         errorM1->SetError(Matrix1, " Что бы удалить столбец, нужно в начале добавить его "); // Установка
         → ошибки, если нет строк
5         return; // Выход из метода
6     }
7     clearMatr1();
8
9     // Проверяем, есть ли столбцы в DataGridView
10    if (this->Matrix1->Columns->Count > 0) {
11        // Получение индекса последнего столбца
12        int i = this->Matrix1->Columns->Count - 1; // Индекс последнего столбца
13        // Удаление
14        this->Matrix1->Columns->RemoveAt(i);
15
16    }
17
18 }

```

Примеры остальных кодов приведены в приложении А.

После запуска приложения на экране появляется окно (см.рисунок 27).

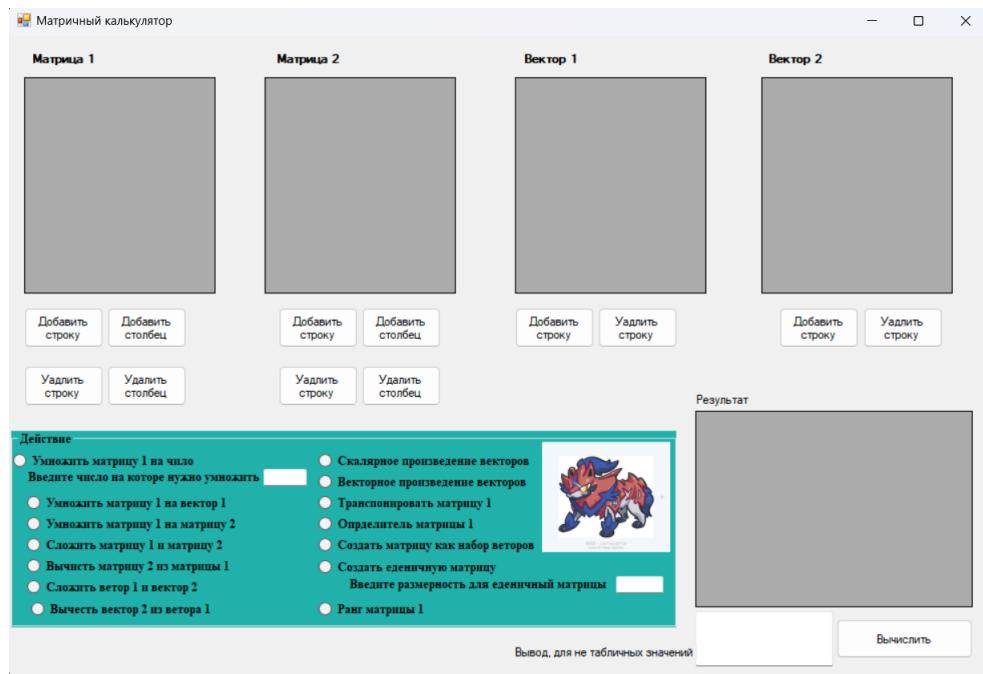


Рисунок 27 – Окно приложения «Матричный калькулятор»: начальный запуск

При нажатии на кнопку «Добавить столбец» под первой таблицей в ней добавляется пустой столбец (см. рисунок 28).

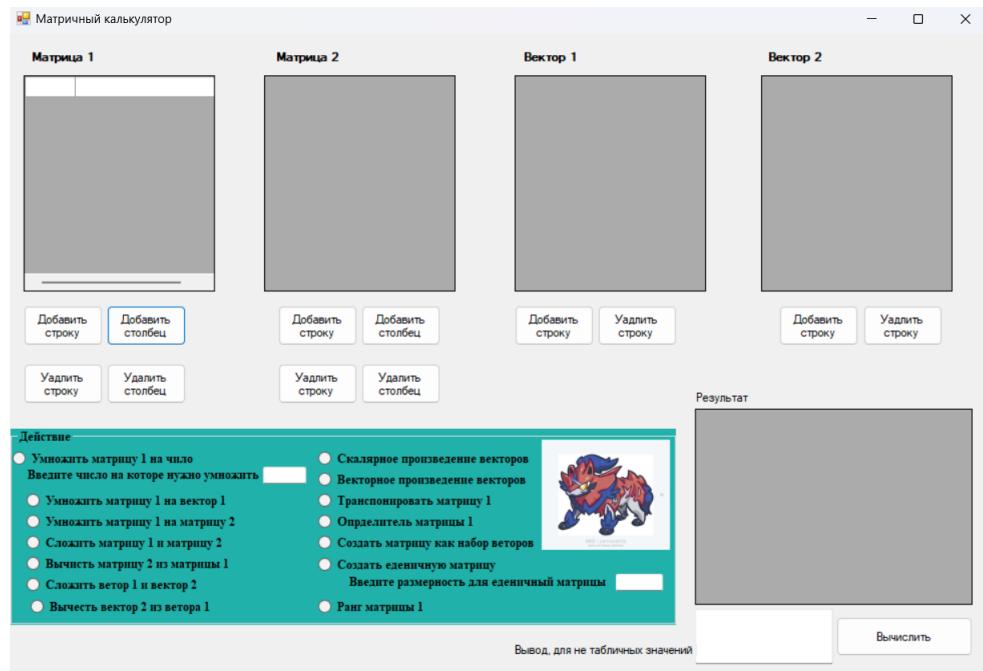


Рисунок 28 – Окно приложения «Матричный калькулятор»: корректное добавление столбца

При нажатии на кнопку «Удалить столбец» под первой таблицей из нее удаляется выбранный столбец (см. рисунок 29).

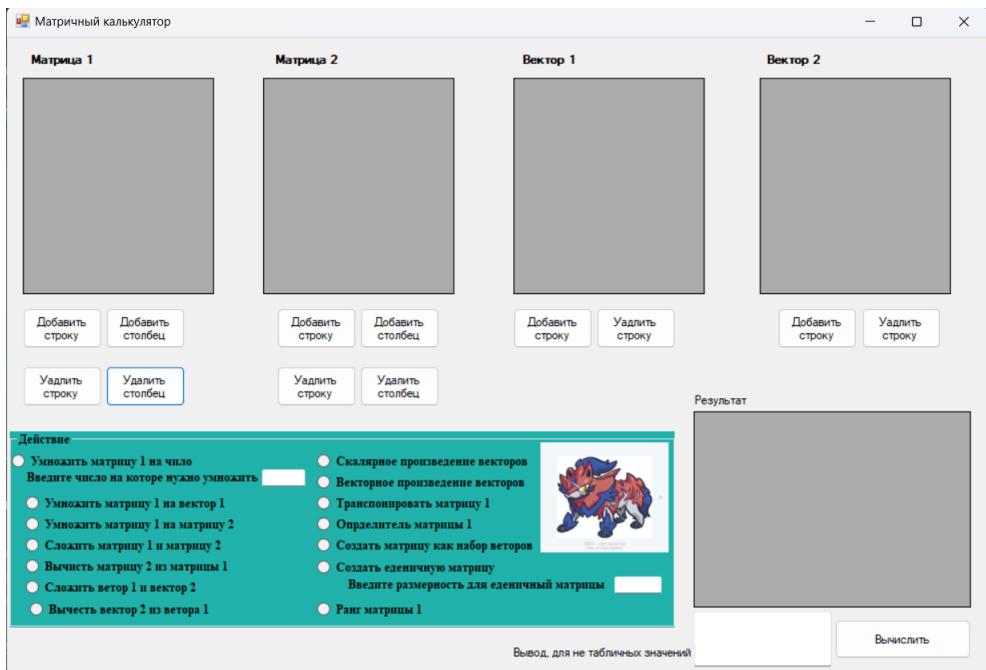


Рисунок 29 – Окно приложения «Матричный калькулятор»: корректное удаление столбца

При выборе переключателя «Скалярное произведение векторов» и нажатия на кнопку «Вычислить» в третье текстовое поле записывается результат (см. рисунок 30).

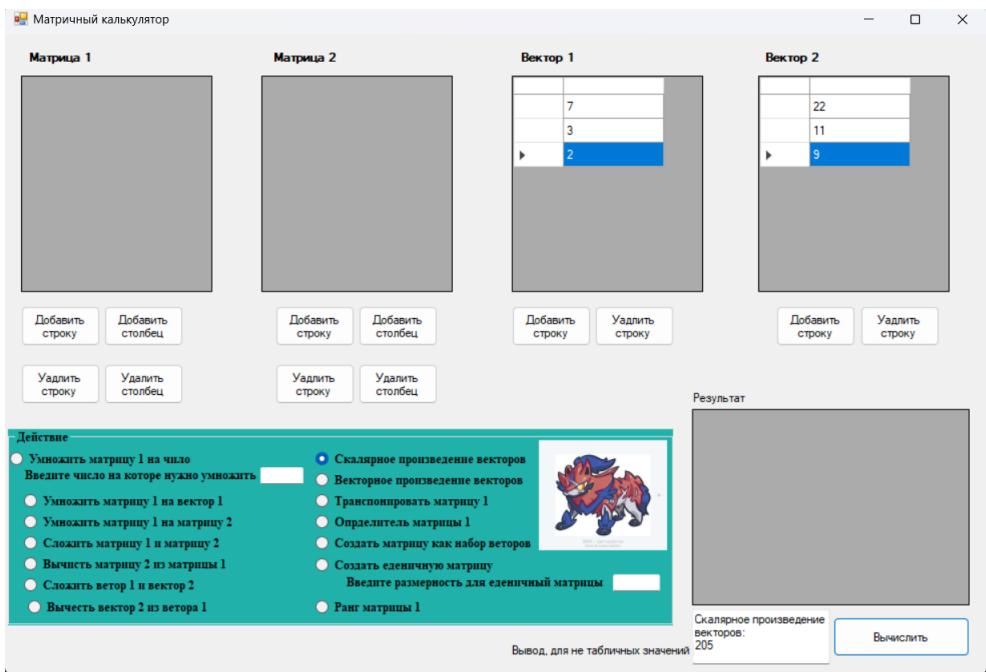


Рисунок 30 – Окно приложения «Матричный калькулятор»: корректное вычисление скалярного произведения векторов

При выборе переключателя «Создать единичную матрицу» и вводе во второе текстовое поле целого числа после нажатия на кнопку «Вычислить» в

пятую таблицу записывается результат создания единичной матрицы с размерностью заданного числа (см. рисунок 31).

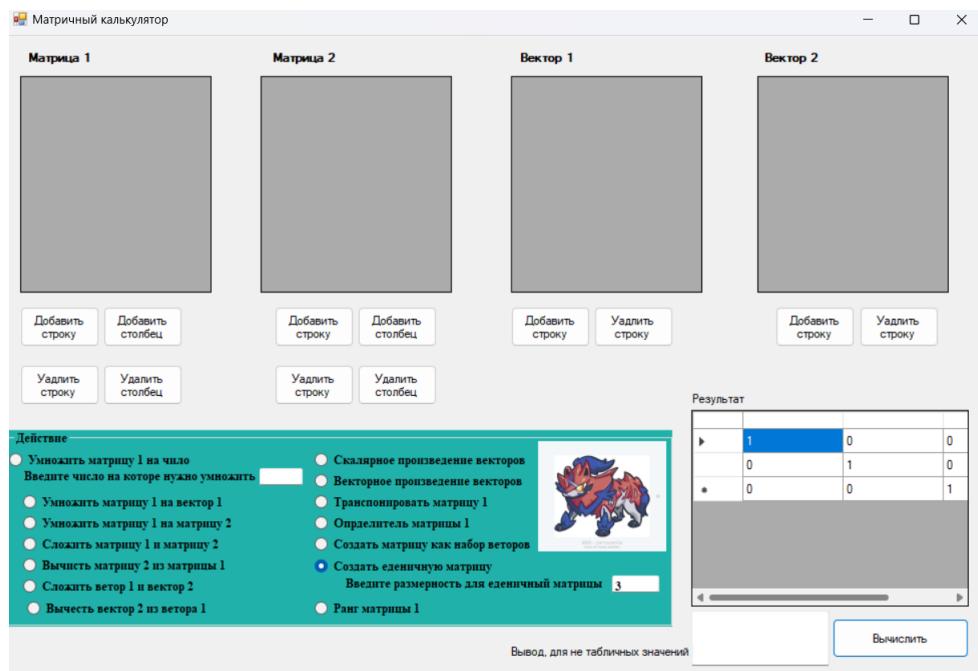


Рисунок 31 – Окно приложения «Матричный калькулятор»: корректное создание единичной матрицы

При выборе переключателя «Умножить первую матрицу на число» и нажатия кнопке «Вычислить» значения матрица изменяются и результат записывается в пятую таблицу (см. рисунок 32).

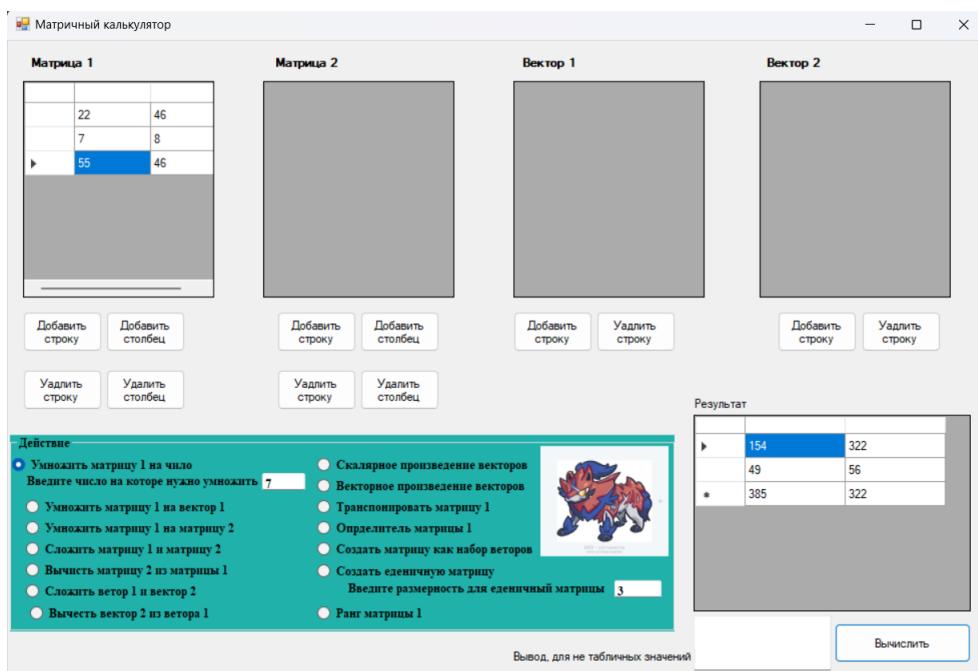


Рисунок 32 – Окно приложения «Матричный калькулятор»: корректное умножение на число

При выборе переключателя «Определитель» и создании не квадратной матрицы возникает сообщение об ошибке (см. рисунок 33).

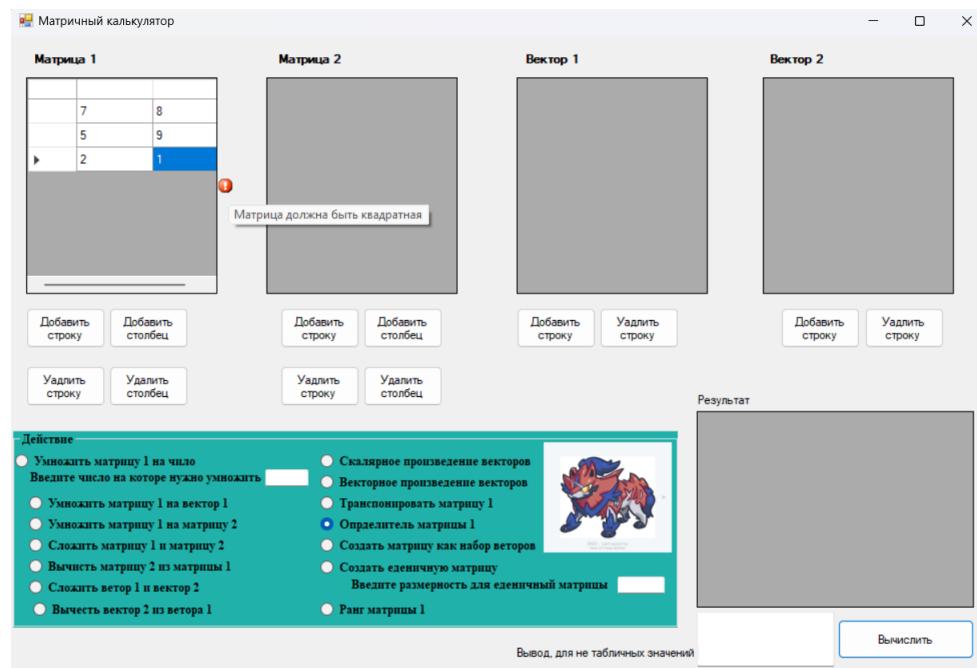


Рисунок 33 – Окно приложения «Матричный калькулятор»: сообщение о некорректном вводе («Матрица должна быть квадратная»)

При выборе переключателя «Векторное произведение векторов» и создании второго вектора другого размера возникает сообщение об ошибке (см. рисунок 34).

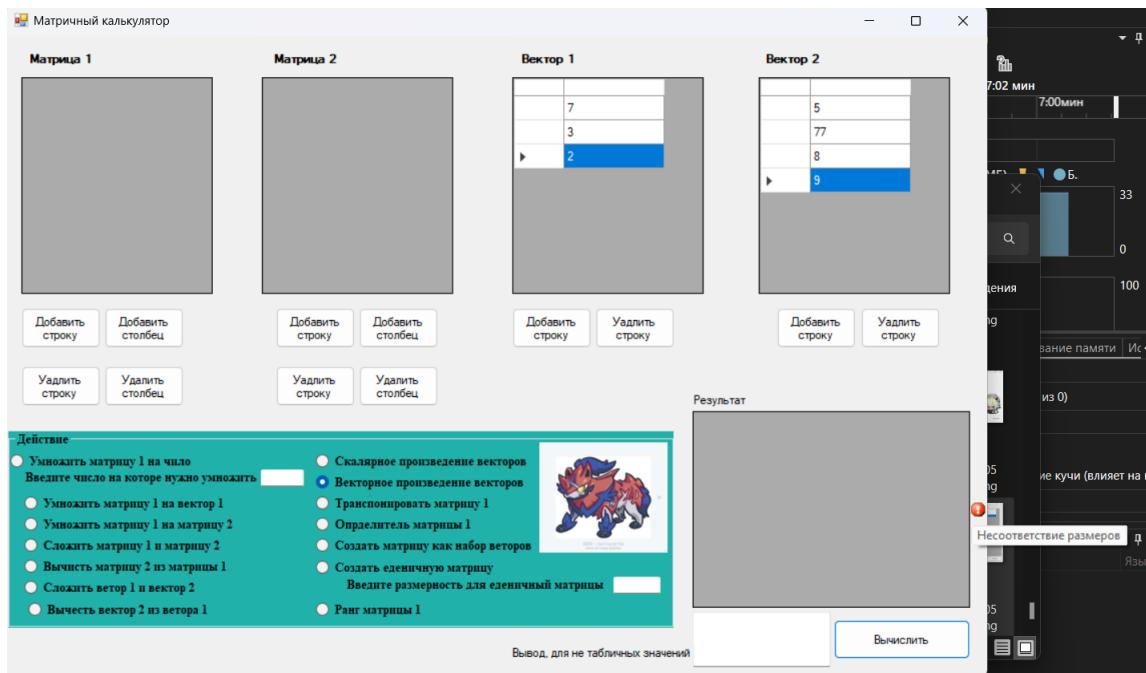


Рисунок 34 – Окно приложения «Матричный калькулятор»: сообщение о некорректном вводе («Несоответствие размеров»)

Полный код программы приведен в приложении **Б**.

7 Файловые диалоги и работа с файлами

Задание. Разработать приложение, реализующее работу с файлами.

Создано окно приложения, содержащее один элемент TextBox, один элемент Label, шесть элементов Button и два элемента DataGridView. Для отображения сообщений об ошибках в окно добавлены три элемента ErrorProvider. Для работы с файлами добавлены элементы SaveFileDialog [?, ?] и OpenFileDialog [?]. Вид окна представлен на рисунке 35.

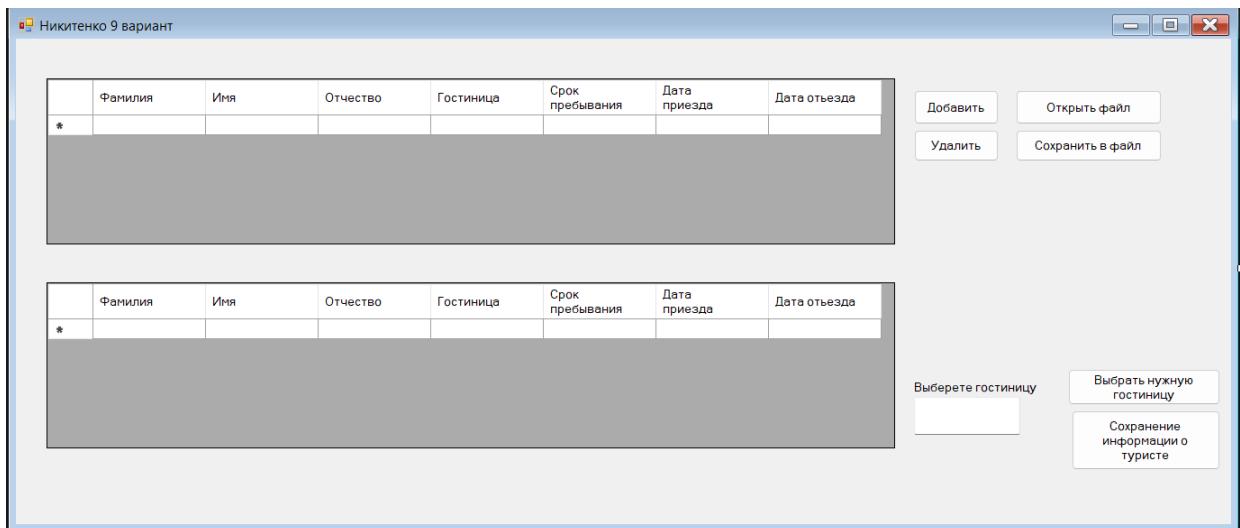


Рисунок 35 – Окно приложения «Никитенко 9 вариант» открытое в конструкторе

У элементов изменены значения некоторых атрибутов. Значения измененных атрибутов представлены в таблицах 10 и 11.

Таблица 10 – Значения атрибутов элементов в приложении «Работа с файлами»

Наименование атрибута	Значение
Для формы	
Text	Никитенко 9 вариант
Для надписи	
Text	Выберите гостиницу
Для текстового поля	
(Name)	ChoiceRace
Для первой кнопки	
(Name)	Add
Text	Добавить

Таблица 11 – Значения атрибутов элементов в приложении «Использование коллекций»

Наименование атрибута	Значение
Для второй кнопки	
(Name)	Delete
Text	Удалить
Для третьей кнопки	
(Name)	OpenFile
Text	Открыть файл
Для четвертой кнопки	
(Name)	SaveInFile
Text	Сохранить в файл
Для пятой кнопки	
(Name)	ChoiceRaceBut
Text	Выбрать нужную гостиницу
Для шестой кнопки	
(Name)	SaveInPass
Text	Сохранение информации о туристе
Для первой таблицы	
(Name)	Table1
NameColumn1	Фамилия
NameColumn2	Имя
NameColumn3	Отчество
NameColumn4	Гостиница
NameColumn5	Срок пребывания
NameColumn6	Дата приезда
NameColumn7	Дата отъезда
Для второй таблицы	
(Name)	Table2
NameColumn1	Фамилия
NameColumn2	Имя
NameColumn3	Отчество
NameColumn4	Гостиница
NameColumn5	Срок пребывания
NameColumn6	Дата приезда
NameColumn7	Дата отъезда

На нажатие кнопки «Открыть файл» установлено выполнение следующего кода:

```
1 private: System::Void OpenFile_Click(System::Object^ sender, System::EventArgs^ e) {
```

2

```

3
4 if (this-> openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK) {
5     System::IO::Stream^ myStream;
6     if ((myStream = openFileDialog1->OpenFile()) != nullptr) {
7         System::IO::StreamReader^ sw = gcnew System::IO::StreamReader(myStream);
8         String^ str = "";
9         String^ person;
10        date DoR;
11
12        int pos = 0;
13        int at = 0;
14        int i = 0;
15
16        // Очищаем старые данные из Table1
17        this->Table1->Rows->Clear();
18
19        while (sw->Peek() > -1) {
20            this->Table1->Rows->Add(1);
21
22            person = sw->ReadLine();
23            person += " ";
24            at = 0;
25            pos = person->IndexOf(" ");
26            for (int j = 0; j < 5; j++) {
27                this->Table1->Rows[i]->Cells[j]->Value = person->Substring(at, pos - at);
28                at = pos + 1;
29                pos = person->IndexOf(" ", at);
30            }
31            DoR = StringToDate(person->Substring(at, pos - at));
32            this->Table1->Rows[i]->Cells[5]->Value = DateToString(DoR);
33            at = pos + 1;
34            pos = person->IndexOf(" ", at);
35            this->Table1->Rows[i]->Cells[6]->Value = person->Substring(at, pos - at);
36
37            i++;
38        }
39    }
40    else {
41        errorProvider1->SetError(OpenFile, "Файл не открыт");
42    }
43}
44else {
45    errorProvider1->SetError(OpenFile, "Файл не открыт");
46}
47}

```

На нажатие кнопки «Сохранить в файл» установлено выполнение следующего кода:

```

1 private: System::Void SaveInFile_Click(System::Object^ sender, System::EventArgs^ e) {
2
3     errorProvider1->Clear();
4     System::IO::Stream^ MyStream;
5     if (this->saveFileDialog1->>ShowDialog() == System::Windows::Forms::DialogResult::OK) {
6         if ((MyStream = this->saveFileDialog1->OpenFile()) != nullptr) {
7             System::IO::StreamWriter^ sw = gcnew System::IO::StreamWriter(MyStream);
8             String^ str = "";
9             int n = this->Table1->RowCount;
10            for (int i = 0; i < n - 1; i++) {
11                str = "";
12                for (int j = 0; j < this->Table1->ColumnCount; j++) {
13                    str += this->Table1->Rows[i]->Cells[j]->Value + " ";
14                }
15                sw->WriteLine(str);
16            }
17            str = "";
18            for (int j = 0; j < this->Table1->ColumnCount; j++) {
19                str += this->Table1->Rows[n - 1]->Cells[j]->Value + " ";
20            }
21            sw->Write(str);
22            sw->Close();
23        }
24    }
25 }

```

Примеры остальных кодов приведены в приложении А.

После запуска приложения на экране появляется окно (см. рисунок 36).

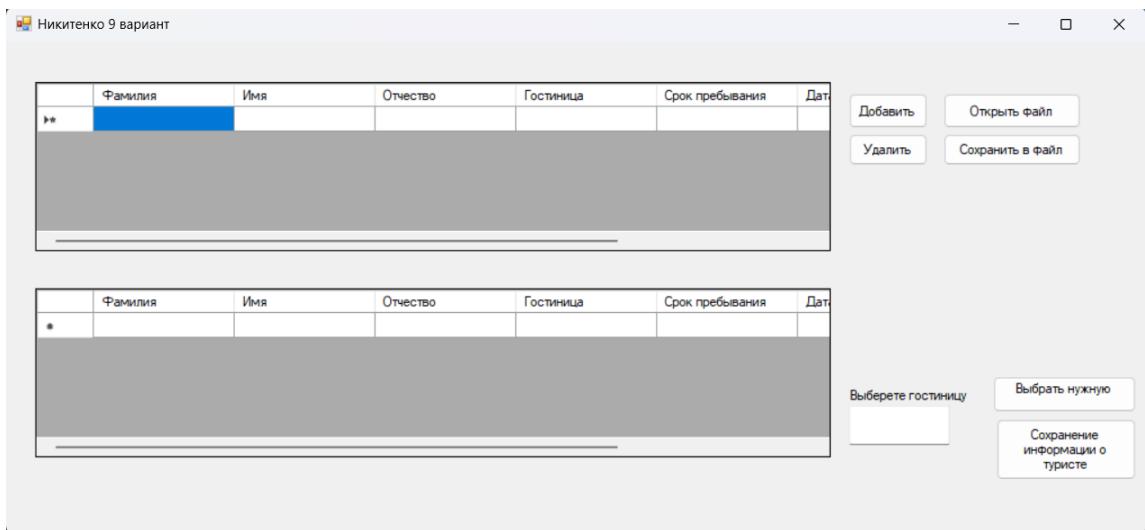


Рисунок 36 – Окно приложения «Работа с файлами»: начальный запуск

При нажатии кнопки «Добавить» в первую таблицу добавляется пустая строка (см. рисунок 37).

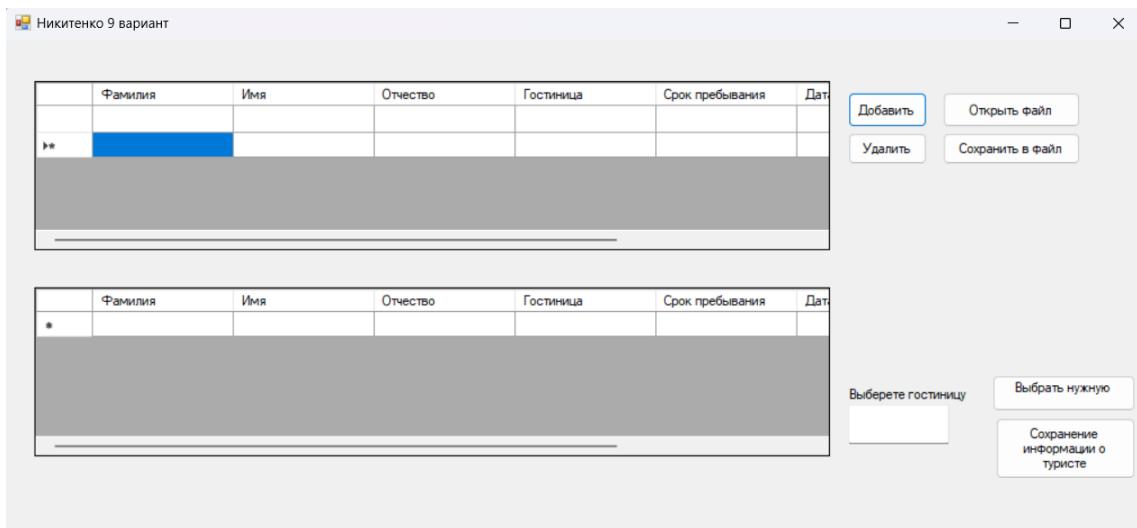


Рисунок 37 – Окно приложения «Работа с файлами»: корректное добавление строки

При нажатии кнопки «Удалить» из первой таблицы удаляется выбранная строка (см. рисунок 38).

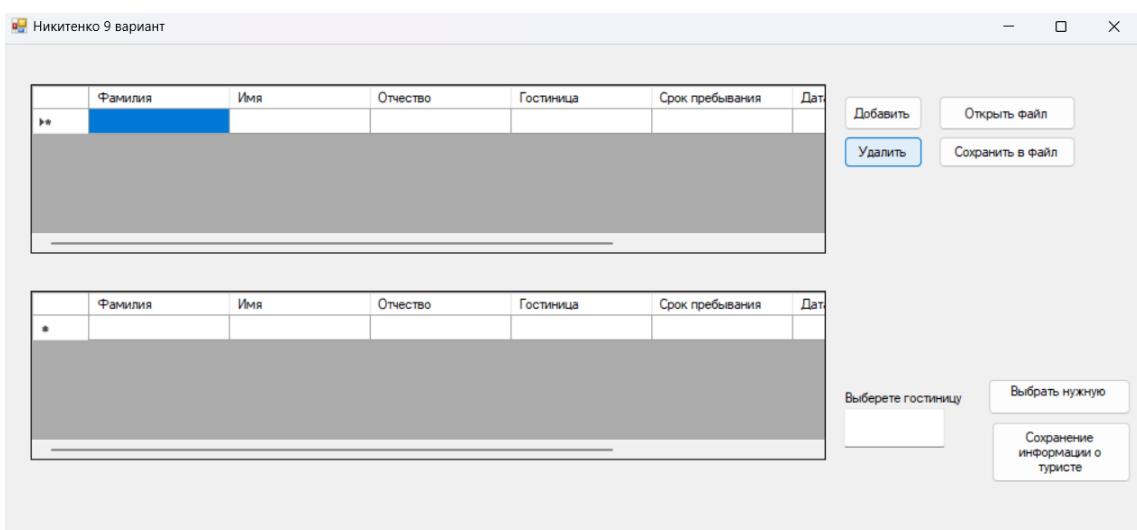


Рисунок 38 – Окно приложения «Работа с файлами»: корректное удаление строки

При нажатии кнопки «Открыть файл» вызывается окно открытия файла (см. рисунок 39).

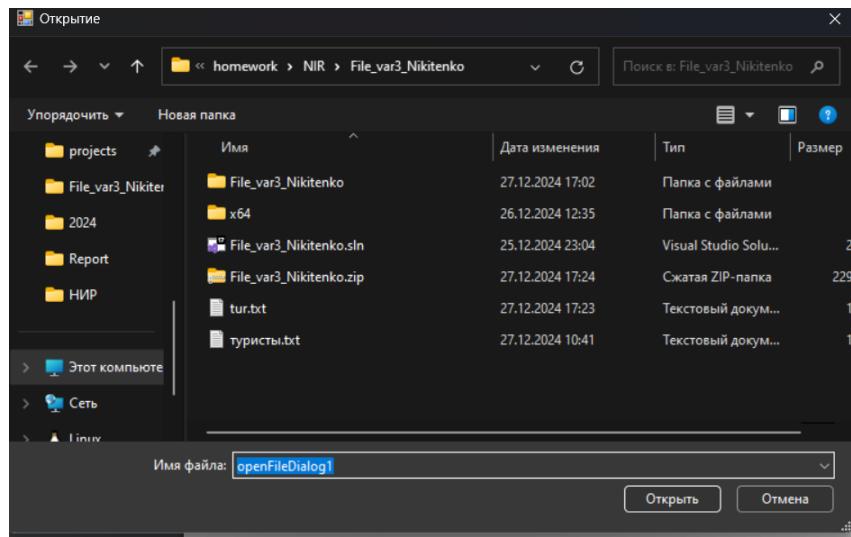


Рисунок 39 – Окно приложения «Работа с файлами»: корректное открытие файла

При нажатии кнопки «Выбрать нужную гостиницу» и вводе корректного названия гостинице в соответствующее поле во второй таблице появляются данные о туристе в этой гостинице (см. рисунок 40).

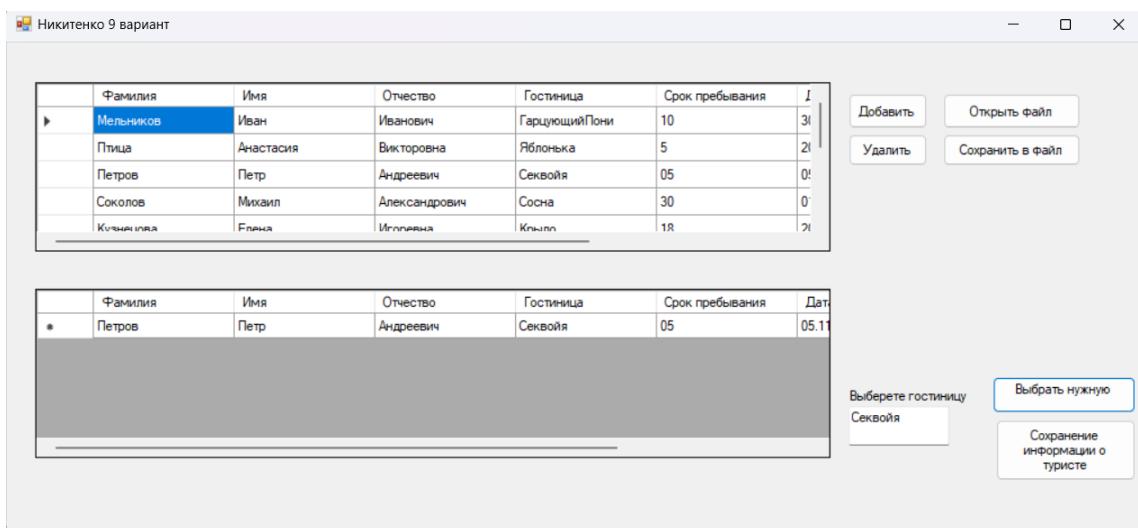


Рисунок 40 – Окно приложения «Работа с файлами»: корректный выбор гостиницы и вывод данных

При нажатии кнопки «Сохранение информации о туристе» открывается окно для сохранения файла с выбранными туристами (см. рисунок 41).

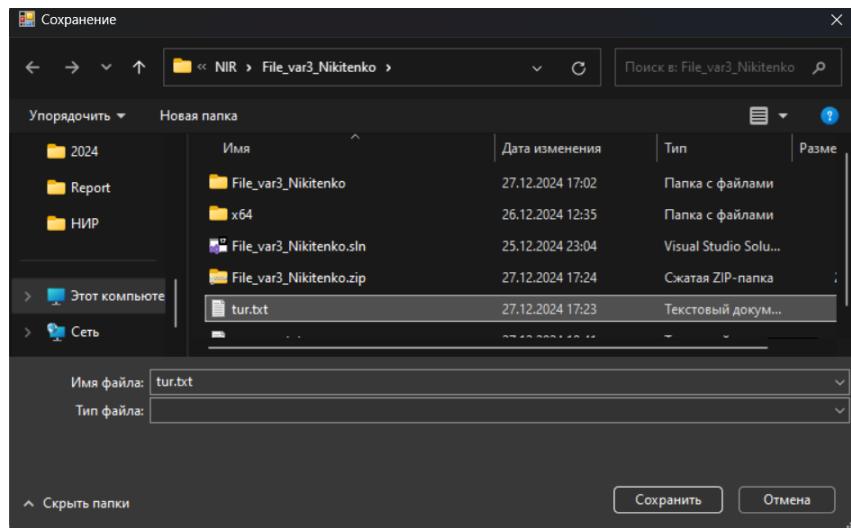


Рисунок 41 – Окно приложения «Работа с файлами»: корректное сохранение файла с выбранными туристами

При нажатии кнопки «Сохранить в файл» открывается окно для сохранения файла со всеми туристами (см. рисунок 42).

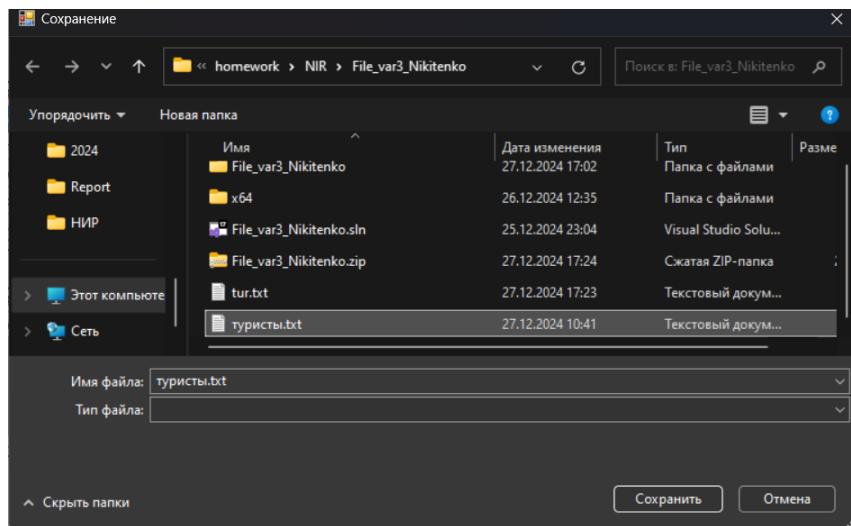


Рисунок 42 – Окно приложения «Работа с файлами»: корректное сохранение файла со всеми туристами

При нажатии кнопки «Открыть файл» и закрытии окна без выбранного файла возникает сообщение об ошибке (см. рисунок 43).

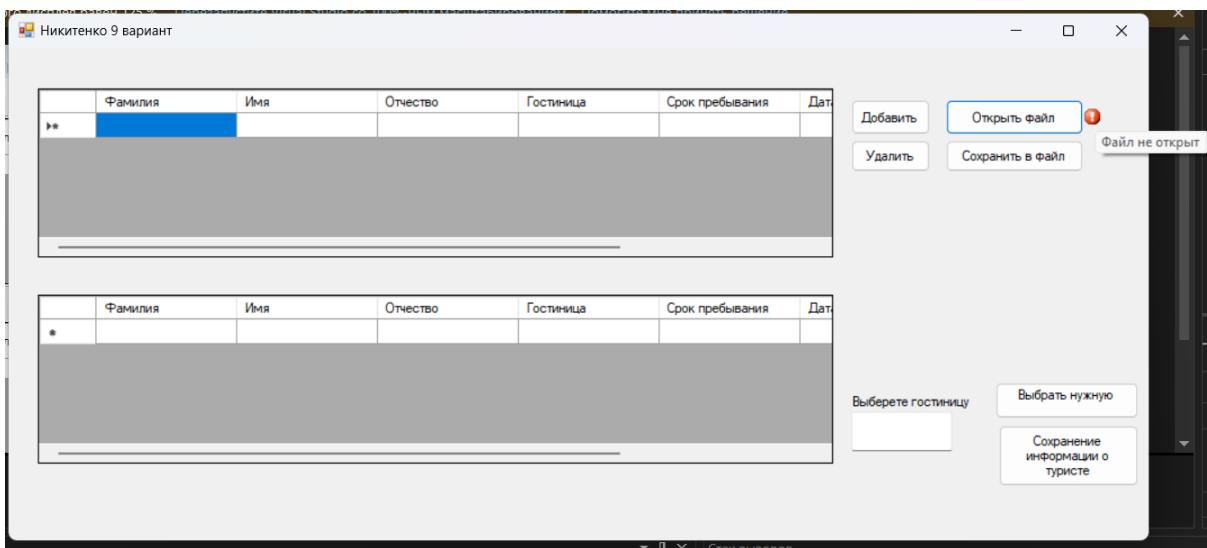


Рисунок 43 – Окно приложения «Работа с файлами»: сообщение о некорректном открытии файла («Файл не открыт»)

При нажатии кнопки «Выбрать нужную гостиницу» и пустом поле с названием гостинцы возникает сообщение об ошибке (см. рисунок 44).

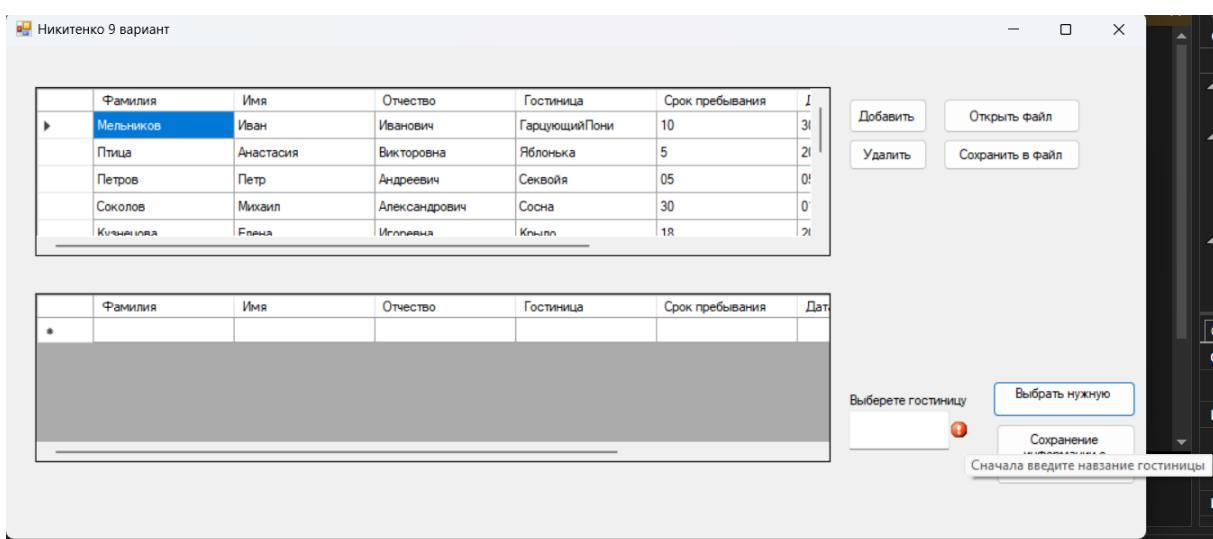


Рисунок 44 – Окно приложения «Работа с файлами»: сообщение о некорректном выборе рейса («Сначала введите название гостиницы»)

Полный код программы приведен в приложении Б.

8 Использование коллекций в Windows Forms

Задание. Разработать приложение, реализующее работу с очередью [?].

Создано окно приложения, содержащее пять элементов TextBox, четыре элемента Label и шесть элементов Button. Для отображения сообщений об ошибках в окно добавлены три элемента ErrorProvider. Вид окна представлен на рисунке 45.

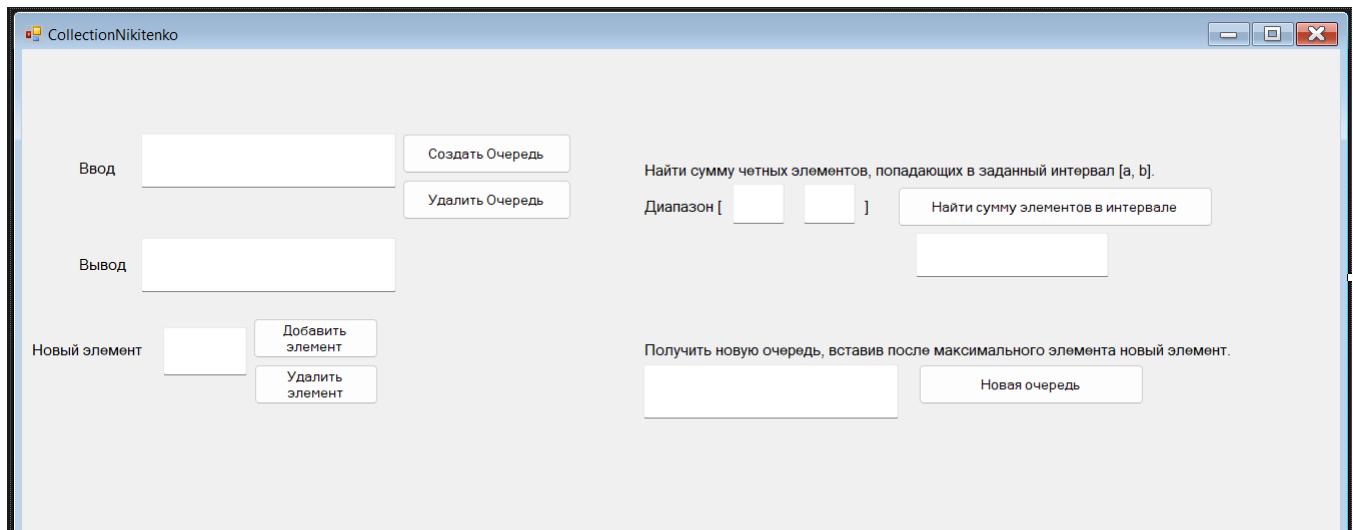


Рисунок 45 – Окно приложения «CollectionNikitenko» открытое в конструкторе

У элементов изменены значения некоторых атрибутов. Значения измененных атрибутов представлены в таблицах 12 и 13.

Таблица 12 – Значения атрибутов элементов в приложении «Использование коллекций»

Наименование атрибута	Значение
Для формы	
Text	CollectionNikitenko
Для первой надписи	
Text	Ввод
Для второй надписи	
Text	Выход
Для третьей надписи	
Text	Новый элемент

Таблица 13 – Значения атрибутов элементов в приложении «Использование коллекций»

Наименование атрибута	Значение
Для четвертой надписи	
Text	Получить новую очередь, вставив после максимального элемента новый элемент.
Для пятой надписи	
Text	Диапазон.
Для шестой надписи	
Text	Найти сумму четных элементов, попадающих в заданный интервал [a, b].
Для первого текстового поля	
(Name)	InPutQu
Для второго текстового поля	
(Name)	OutPutQu
Для третьего текстового поля	
(Name)	NewEIn
Для четвертого текстового поля	
(Name)	a
Для пятого текстового поля	
(Name)	b
Для шестого текстового поля	
(Name)	ChotElem
Для седьмого текстового поля	
(Name)	NewQOut
Для первой кнопки	
(Name)	CreateQ
Text	Создать очередь
Для второй кнопки	
(Name)	DeleteQu
Text	Удалить очередь
Для третьей кнопки	
(Name)	AddEl
Text	Добавить элемент
Для четвертой кнопки	
(Name)	DeleteEl
Text	Удалить элемент
Для пятой кнопки	
(Name)	Interval
Text	Найти сумму элементов в интервале
Для шестой кнопки	
(Name)	NewQ
Text	Новая очередь

На нажатие кнопки «Создать очередь» установлено выполнение следующего кода:

```
1 private: System::Void CreateQ_Click(System::Object^ sender, System::EventArgs^ e) {
2
3     q.Clear();
4     errorProvider1->Clear();
5
6     System::String^ str = System::Convert::ToString(this -> InPutQu -> Text);
7     str += " ";
8     int X, at, pos = 0;
9     at = str -> IndexOf(" ");
10    while (at != -1) {
11        System::String ^ str1 = str->Substring(pos,at-pos);
12        pos = at + 1;
13        bool res = Int32::TryParse(str1, X);
14
15        if (!X) {
16            errorProvider1->SetError(InPutQu, "В строке есть не числа");
17            return;
18        }
19
20        q.Enqueue(X);
21        at = str -> IndexOf(" ", pos);
22
23    }
24    System::Collections::Generic::Queue<int> buf;
25    System::String ^ str2 = "";
26    while (q.Count) {
27        buf.Enqueue(q.Peek());
28        str2 += System::Convert::ToString(q.Peek()) + " ";
29        q.Dequeue();
30    }
31    while (buf.Count) {
32        q.Enqueue(buf.Peek());
33        buf.Dequeue();
34
35    }
36    this -> OutPutQu -> Text = str2;
37 }
```

На нажатие кнопки «Найти сумму элементов в интервале» установлено выполнение следующего кода:

```
1 private: System::Void Interval_Click(System::Object^ sender, System::EventArgs^ e) {
2
3     ChotElem->Text = String::Empty; //очистка поля
4     int InterA;
```

```

5 int interB;
6 bool resultA = Int32::TryParse(this->a->Text, InterA);
7 bool resultB = Int32::TryParse(this->b->Text, interB);
8
9 if (!resultA && !resultB) { //если а и b не числа
10    errora->SetError(a, "Введено не целое число");
11    errorb->SetError(b, "Введено не целое число");
12}
13
14 else if (!resultB) { //если b не число
15    errorb->SetError(b, "Введено не целое число");
16    errora->SetError(a, String::Empty);
17}
18 else if (!resultA) { //если а не число
19    errora->SetError(a, "Введено не целое число");
20    errorb->SetError(b, String::Empty);
21}
22 else {
23    if (InterA > interB) { //если а > b
24        errora->SetError(a, "Неправильно введен диапазон");
25        errorb->SetError(b, String::Empty);
26    }
27    // Получение элементов из OutPutQu
28    array<String> elements = OutPutQu->Text->Split(' ');
29
30    int sum = 0;
31    int count1 = 0;
32
33    for each (String element in elements) {
34        int number;
35        if (Int32::TryParse(element->Trim(), number)) {
36            if (number >= InterA && number <= interB && number % 2 == 0) { // проверка на четность и
37                // диапазон
38                sum += number;
39                count1++;
40            }
41        }
42
43    if (count1 == 0) { // если нет таких элементов
44        ChotElem->Text = "Таких элементов нет";
45    }
46    else { // элементы есть
47        ChotElem->Text = System::Convert::ToString(sum); // вывод суммы четных элементов
48    }
49}
50}

```

Примеры остальных кодов приведены в приложении А.

После запуска приложения на экране появляется окно (см. рисунок 46).

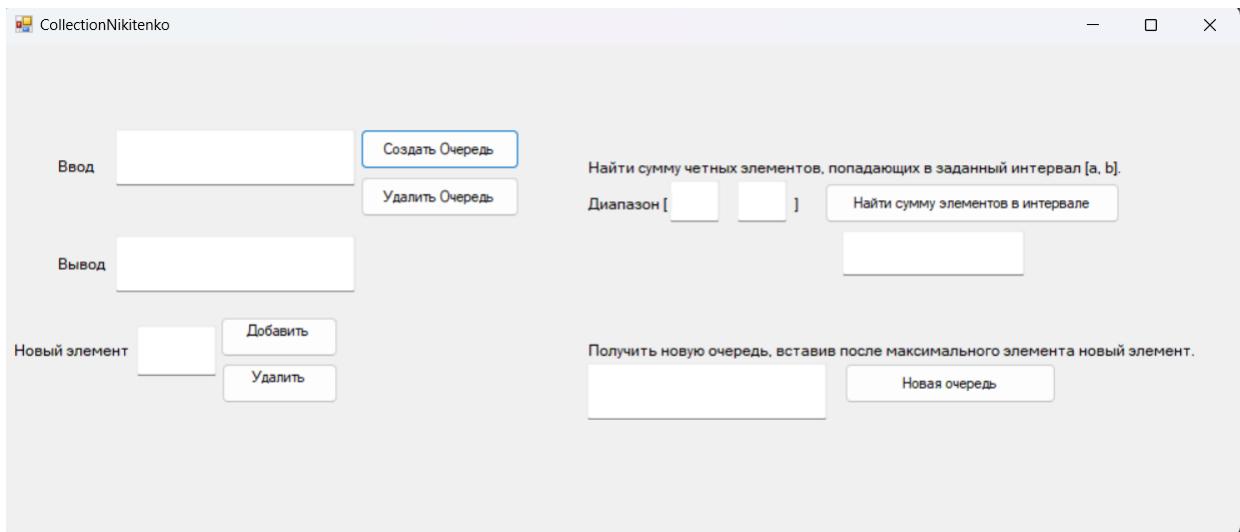


Рисунок 46 – Окно приложения «CollectionNikitenko»: начальный запуск

При вводе целых чисел в поле для ввода и нажатии кнопок «Создать очередь», «Новая очередь», «Найти сумму элементов в интервале» в поле вывода очереди записываются введенные числа, в поле для новой очереди записывается очередь, потом находится максимальный элемент и после него вставляется новый элемент и после введения диапазона записывается сумма элементов попавших в этот диапазон (см. рисунок 47).

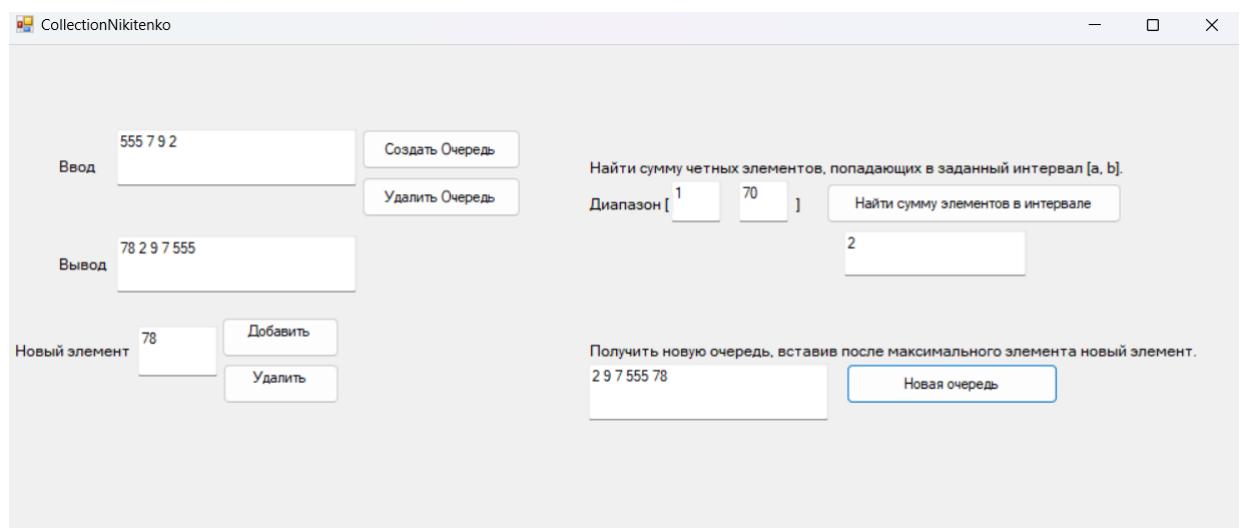


Рисунок 47 – Окно приложения «CollectionNikitenko»: корректное создание очереди и выполнение указанных действий

При нажатии кнопки «Удалить очередь» выполняется удаление очереди (см. рисунок 48).

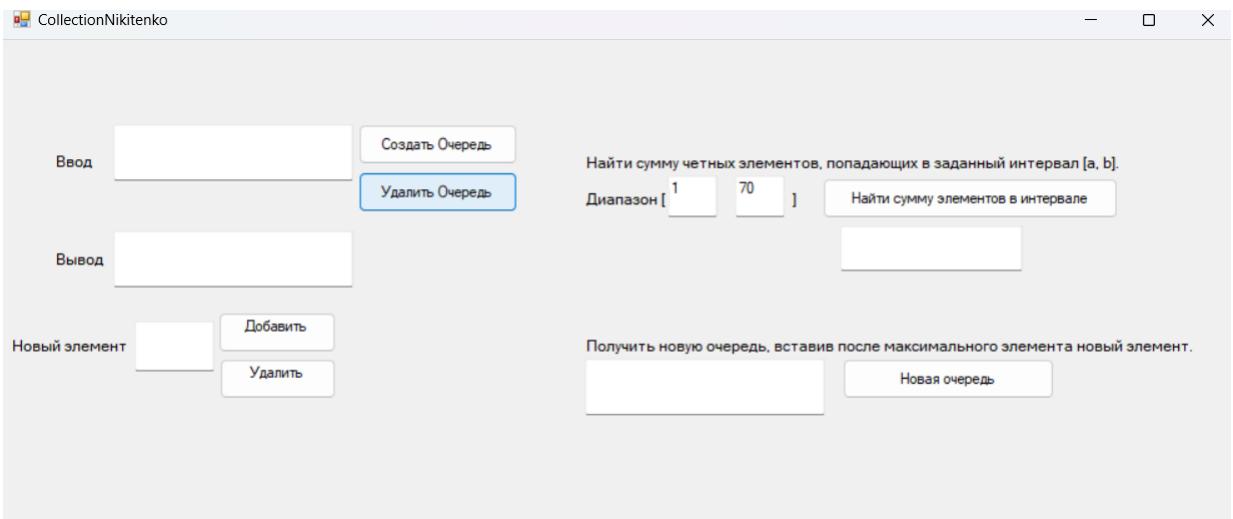


Рисунок 48 – Окно приложения «Использование коллекций»: корректное удаление очереди

При нажатии на кнопку «Добавить элемент» и вводе целого числа в соответствующее поле в поле вывода очереди записывается новый элемент (см. рисунок 49).

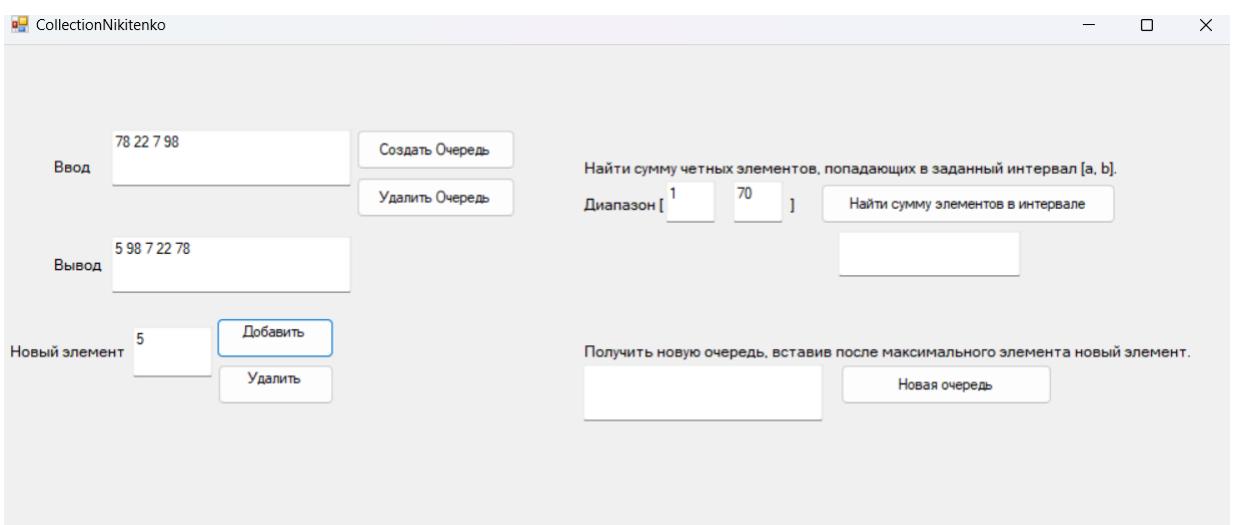


Рисунок 49 – Окно приложения «Использование коллекций»: корректное добавление нового элемента

При нажатии кнопки «Добавить элемент» и отсутствии целого числа в соответствующем поле выводится сообщение об ошибке (см. рисунок 50).

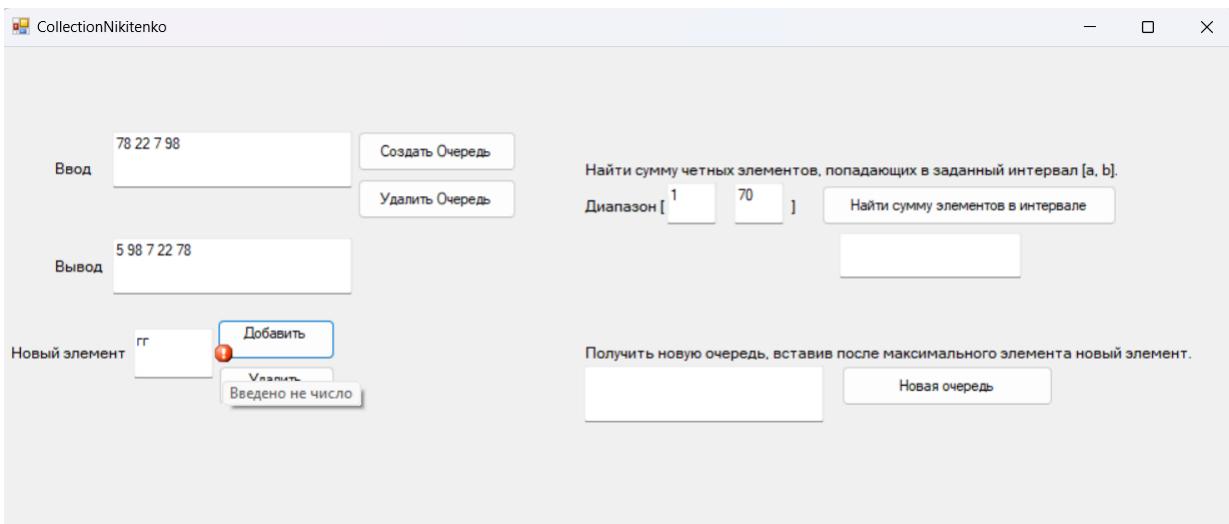


Рисунок 50 – Окно приложения «Использование коллекций»: сообщение о некорректном вводе («Введено не число»)

При нажатии кнопки «Найти сумму элементов в интервале» и отсутствии четных элементов в данном интервале выводиться сообщение о том что таких элементов нет (см. рисунок 51).

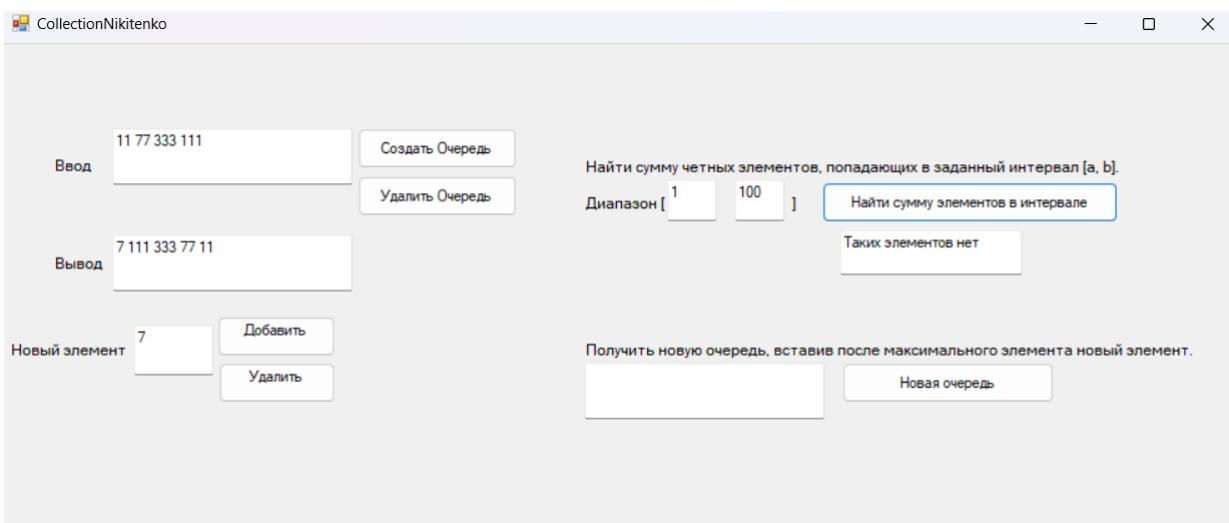


Рисунок 51 – Окно приложения «Использование коллекций»: сообщение о об отсутствии элементов («Таких элементов нет»)

Полный код программы приведен в приложении Б.

9 Приложение ТЕСТ

Задание. Разработать приложение для тестирования.

Создано окно приложения, содержащее один элемент TextBox, шесть элементов Label, четыре элемента Button, три элемента CheckBox и три элемента RadioButton. Вид окна представлен на рисунке 52.

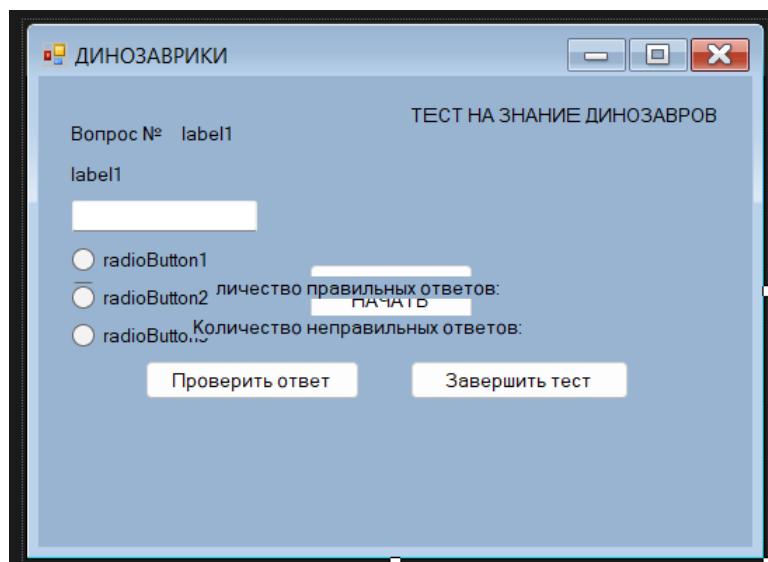


Рисунок 52 – Окно приложения «Тест» открытое в конструкторе

У элементов изменены значения некоторых атрибутов. Значения измененных атрибутов представлены в таблицах 14 и 15.

Таблица 14 – Значения атрибутов элементов в приложении «Тест»

Наименование атрибута	Значение
Для формы	
Text	ДИНОЗАВРИКИ
Для текстового поля	
Visible	False
Для первого переключателя	
Visible	False
Для второго переключателя	
Visible	False
Для третьего переключателя	
Visible	False
Для первой надписи	
(Name)	lblNum1
Text	Вопрос №
Visible	False

Таблица 15 – Значения атрибутов элементов в приложении «Использование коллекций»

Наименование атрибута	Значение
Для первого флажка	
Visible	False
Для второго флажка	
Visible	False
Для третьего флажка	
Visible	False
Для второй надписи	
(Name)	lblNum2
Visible	False
Для третьей надписи	
(Name)	lblQuestion
Visible	False
Для четвертой надписи	
(Name)	lblAnswer
Visible	False
Для пятой надписи	
(Name)	lblTrue
Text	Количество правильных ответов:
Visible	False
Для шестой надписи	
(Name)	lblFalse
Text	Количество неправильных ответов:
Visible	False
Для первой кнопки	
(Name)	btnStart
Text	НАЧАТЬ
Для второй кнопки	
(Name)	btnCheck
Text	Проверить ответ
Visible	False
Для третьей кнопки	
(Name)	btnNext
Text	Следующий вопрос
Visible	False
Для четвертой кнопки	
(Name)	btnRes
Text	Завершить тест
Visible	False

На нажатие кнопки «НАЧАТЬ» установлено выполнение следующего кода:

```
1 private: System::Void btnStart_Click(System::Object^ sender, System::EventArgs^ e) {  
2     btnStart->Visible = false;  
3     lblNum1->Visible = true;  
4     lblNum2->Visible = true;  
5     lblQuestion->Visible = true;  
6     btnCheck->Visible = true;  
7  
8     btnNext_Click(sender, e); // Переход к выполнению кода, заданного для кнопки "след. вопрос"  
9 }
```

На нажатие кнопки «Проверить ответ» установлено выполнение следующего кода:

```
1 private: System::Void btnCheck_Click(System::Object^ sender, System::EventArgs^ e) {  
2     switch (curElem)  
3     {  
4         case 1:  
5             if (radioButton3->Checked) CheckAnswer(1);  
6             else CheckAnswer(0);  
7             break;  
8  
9         case 2:  
10            if (radioButton1->Checked) CheckAnswer(1);  
11            else CheckAnswer(0);  
12            break;  
13  
14        case 3:  
15            if (radioButton1->Checked) CheckAnswer(1);  
16            else CheckAnswer(0);  
17            break;  
18  
19        case 4:  
20            if (radioButton2->Checked) CheckAnswer(1);  
21            else CheckAnswer(0);  
22            break;  
23  
24        case 5:  
25            if (checkBox1->Checked && checkBox2->Checked && !checkBox3->Checked) CheckAnswer(1);  
26            else CheckAnswer(0);  
27            break;  
28  
29        case 6:  
30            if (radioButton1->Checked) CheckAnswer(1);  
31            else CheckAnswer(0);  
32            break;
```

```

33
34 case 7:
35 if (checkBox1->Checked && checkBox2->Checked && checkBox3->Checked) CheckAnswer(1);
36 else CheckAnswer(0);
37 break;
38
39 case 8:
40 if (textBox1->Text=="эпидексиптерикс" || textBox1->Text=="Эпидексиптерикс") CheckAnswer(1);
41 else CheckAnswer(0);
42 break;
43
44 case 9:
45 if (radioButton3->Checked) CheckAnswer(1);
46 else CheckAnswer(0);
47 break;
48
49 case 10:
50 if (checkBox1->Checked && !checkBox2->Checked && !checkBox3->Checked) CheckAnswer(1);
51 else CheckAnswer(0);
52 break;
53
54 default:
55 break;
56 }
57 ///
58
59 // устанавливается запрет на выбор вариантов/ввод ответа
60 textBox1->ReadOnly = true;
61 radioButton1->Enabled = false;
62 radioButton2->Enabled = false;
63 radioButton3->Enabled = false;
64 checkBox1->Enabled = false;
65 checkBox2->Enabled = false;
66 checkBox3->Enabled = false;
67
68 btnCheck->Visible = false;
69
70 if (num < 10) { // если вопросы не закончились, то появляется кнопка след. вопроса
71 btnNext->Visible = true;
72 }
73 else { // если вопросы закончились, то появляется кнопка для завершения тестирования
74 btnRes->Visible = true;
75 }
76 }

```

На нажатие кнопки «Завершить тест» установлено выполнение следующего кода:

```
1 private: System::Void btnRes_Click(System::Object^ sender, System::EventArgs^ e) {
```

```

2 //скрытие всех ненужных элементов формы
3 lblNum1->Visible = false;
4 lblNum2->Visible = false;
5 lblQuestion->Visible = false;
6 btnCheck->Visible = false;
7 lblTrue->Visible = true;
8 lblFalse->Visible = true;
9 btnRes->Visible = false;
10 lblAnswer->Visible = false;
11 textBox1->Visible = false;
12 radioButton1->Visible = false;
13 radioButton2->Visible = false;
14 radioButton3->Visible = false;
15 checkBox1->Visible = false;
16 checkBox2->Visible = false;
17 checkBox3->Visible = false;
18
19 // вывод правильных и неправильных ответов
20 lblTrue->Text += System::Convert::ToString(countTrue);
21 lblFalse->Text += System::Convert::ToString(countFalse);
22 //
23 }

```

Примеры остальных кодов приведены в приложении А.

После запуска приложения на экране появляется окно (см. рисунок 53).

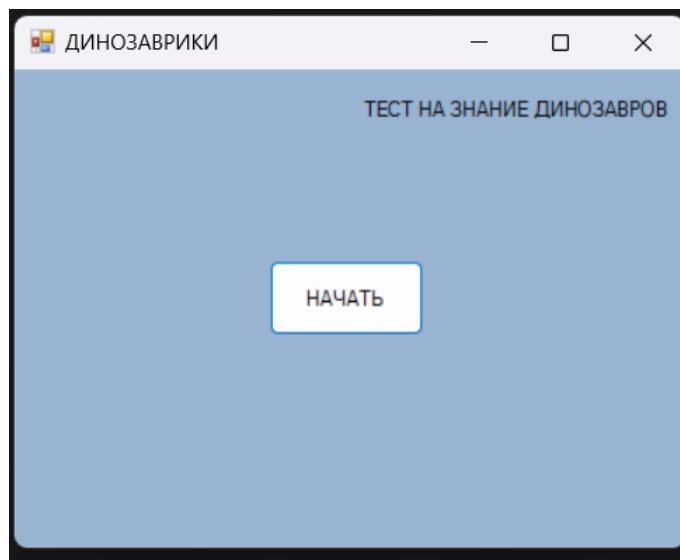


Рисунок 53 – Окно приложения «Тест»: начальный запуск

При нажатии на кнопку «НАЧАТЬ» некоторые элементы формы становятся видимыми и в случайном порядке появляются вопросы и варианты ответа на них (см. рисунок 54).

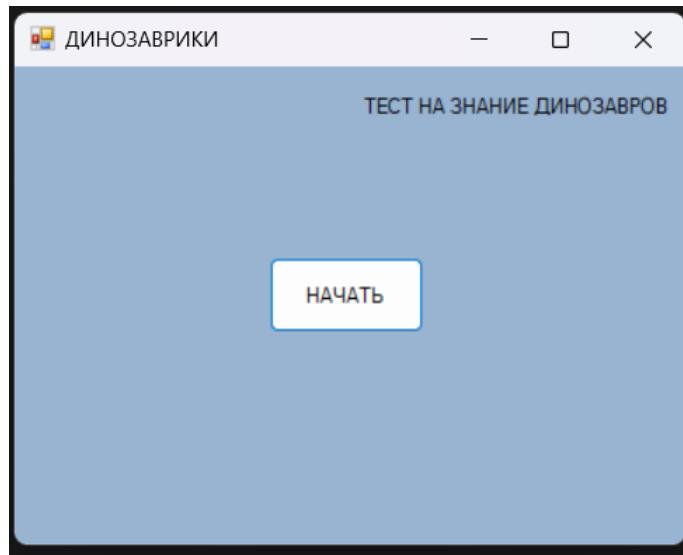


Рисунок 54 – Окно приложения «Тест»: начало тестирования

При выборе ответа и нажатии на кнопку «Проверить ответ» появляется сообщение о правильности ответа, кнопка «Следующий вопрос» и запрещается редактировать ответ (см. рисунок 55).

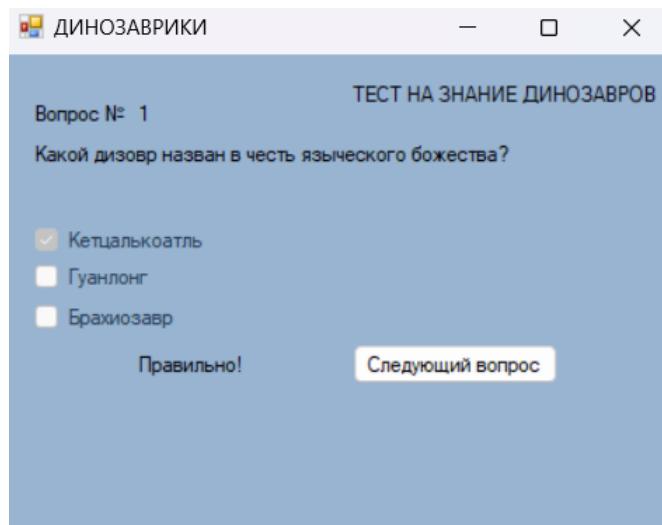


Рисунок 55 – Окно приложения «Тест»: ответы на вопросы

Когда вопросы заканчиваются, появляется кнопка «Завершить тест» (см. рисунок 56).

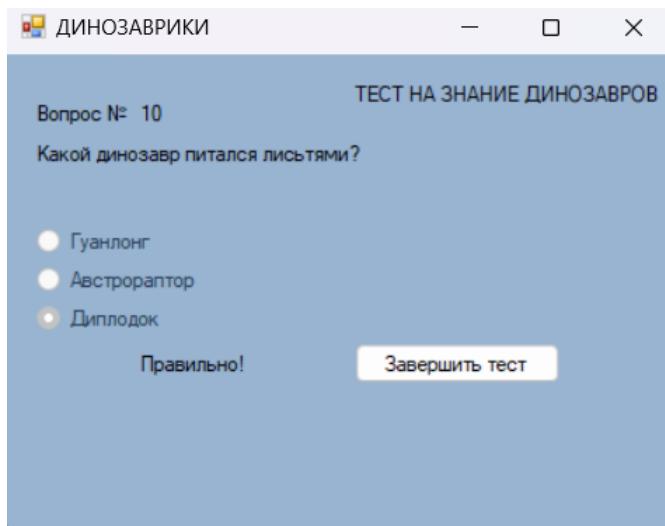


Рисунок 56 – Окно приложения «Тест»: завершение теста

При нажатии кнопки «Завершить тест» появляется информация с результатами прохождения теста (см. рисунок 57).

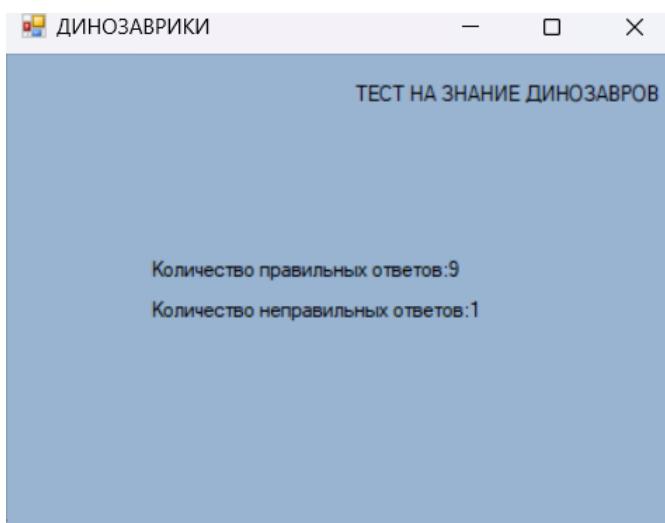


Рисунок 57 – Окно приложения «Тест»: результаты тестирования

Полный код программы приведен в приложении Б.

ЗАКЛЮЧЕНИЕ

В ходе практики было реализовано несколько приложений на языке C++ в среде Microsoft Visual Studio с целью закрепления навыков построения оконного интерфейса. На практике разработаны приложения, содержащие такие элементы интерфейса как TextBox, Label, Button, DataGridView, OpenFileDialog, SaveFileDialog, ErrorProvider.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Создание приложения Windows Forms с помощью .NET Framework (C++) [Электронный ресурс]. — URL: [https://learn.microsoft.com/ru-ru/previous-versions/ms235634\(v=vs.100\)?redirectedfrom=MSDN](https://learn.microsoft.com/ru-ru/previous-versions/ms235634(v=vs.100)?redirectedfrom=MSDN) (Дата обращения 30.12.2024).
- 2 Рекурсивные функции [Электронный ресурс]. — URL: <https://learn.microsoft.com/ru-ru/cpp/c-language/recursive-functions?view=msvc-170> (Дата обращения 30.12.2024).
- 3 Button Class [Электронный ресурс]. — URL: <https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.button?view=netframework-4.0> (Дата обращения 30.12.2024). Загл. с экрана. Яз. англ.
- 4 ErrorProvider Class [Электронный ресурс]. — URL: <https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.errorprovider?view=windowsdesktop-8.0> (Дата обращения 30.12.2024). Загл. с экрана. Яз. англ.
- 5 DataGridView Класс [Электронный ресурс]. — URL: <https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.forms.datagridview?view=netframework-4.0> (Дата обращения 04.01.2025). Загл. с экрана. Яз. англ.
- 6 Пошаговое руководство. Умножение матриц [Электронный ресурс]. — URL: <https://learn.microsoft.com/ru-ru/cpp/parallel/amp/walkthrough-matrix-multiplication?view=msvc-170> (Дата обращения 04.01.2025). Загл. с экрана. Яз. англ.
- 7 SaveFileDialog Класс [Электронный ресурс]. — URL: <https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.forms.savefigledialog?view=netframework-4.0> (Дата обращения 04.01.2025). Загл. с экрана. Яз. англ.
- 8 Павловская, С/С++. Программирование на языке высокого уровня / Павловская. — Санкт - Перербург: Питер, 2010.
- 9 OpenFileDialog Класс [Электронный ресурс]. — URL: <https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.forms.openfiledialog?view=netframework-4.0> (Дата обращения 04.01.2025). Загл. с экрана. Яз. англ.
- 10 Страуструп, Программирование. Принципы и практика использования C++ / Страуструп. — Москва: ЛитРес, 2022.

ПРИЛОЖЕНИЕ А

Примеры кода

Обработка табличных данных. Часть 1

На нажатие кнопки «Среднее арифметическое» установлено выполнение следующего кода:

```
1 double sum = 0; // Переменная для хранения суммы значений
2 int AmountOfNumbers = 0; // счетчик
3 bool hasNonNumeric = false; // Флаг для проверки не числовых значений
4
5
6
7 // Вычисление суммы, счетчик и нахождения четных элементов
8 for (int i = 0; i < this->grdMassiv->Rows->Count; i++) {
9
10
11
12 // Пропуск пустых ячеек
13 // Проверка на наличие значения в ячейке
14 if (grdMassiv->Rows[i]->Cells[0]->Value == nullptr) {
15     continue; // Пропустить пустую клетку
16 }
17
18 // Получаем значение ячейки
19 String^ cellValue = grdMassiv->Rows[i]->Cells[0]->Value->ToString();
20
21 // Проверка на пустую клетку
22 if (String::IsNull(cellValue)) {
23     continue; // Пропустить пустую клетку
24 }
25
26 //
27
28 double num;
29 if (Double::TryParse(grdMassiv->Rows[i]->Cells[0]->Value->ToString(), num)) {
30     if (static_cast<int>(num) % 2 == 0 && num != 0) { // Нахождение четных элементов
31         sum += num; // Суммирование элементов
32         AmountOfNumbers++; // Счетчик
33     }
34 }
35 else {
36     hasNonNumeric = true; // Установка флага если, значение не числовое
37 }
38 }
39 //
40 // Вычисление среднего арифметического,
```

```

42 if (AmountOfNumbers > 0) {
43     double average = sum / AmountOfNumbers;
44     this->SrResult->Text = System::Convert::ToString(average); // Отображение результата
45 }
46 else {
47     this->SrResult->Text = "Увы их нет :0"; // Если будет все плохо, 0
48 }
49 }
50
51 // Проверка на наличие нечисловых значений
52 if (hasNonNumeric) {
53     errorProvider1->SetError(SrResult, "Присутствуют нечисловые значения в массиве. Возможны
      ↪ неточности");
54 }
55 else {
56     errorProvider1->SetError(SrResult, ""); // Очищение сообщения об ошибки, если все норм
57 }
58 //

```

На нажатие кнопки «Номера нечетных элементов» установлено выполнение следующего кода:

```

1 double summ = 0;
2 int AmountofNumberss = 0;
3 bool hasNonNumeric = false; // Флаг для проверки не числовых значений
4
5 // Стока для вывода индекса(ну почему бы и нет)
6 System::String^ oddIndices = "";
7 int oddCount = 0; // Счетчик нечетных чисел, дабы могла работать проверка на наличие нечетных чисел
8
9
10 for (int i = 0; i < this->grdMassiv->Rows->Count; i++) {
11
12
13 // Пропуск пустых ячеек
14
15 // Проверка на наличие значения в ячейке
16 if (grdMassiv->Rows[i]->Cells[0]->Value == nullptr) {
17     continue; // Пропустить пустую клетку
18 }
19
20 // Получаем значение ячейки
21 String^ cellValue = grdMassiv->Rows[i]->Cells[0]->Value->ToString();
22
23 // Проверка на пустую клетку
24 if (String::IsNullOrEmpty(cellValue)) {
25     continue; // Пропустить пустую клетку
26 }

```

```

27
28     // 
29
30
31     double numm;
32
33     if (Double::TryParse(grdMassiv->Rows[i]->Cells[0]->Value->ToString(), numm)) { // Проверка, можно
34         ← ли преобразовать значение в число
35
36         if (static_cast<int>(numm) % 2 != 0 && numm != 0) { // Проверка, является ли число нечетным
37             oddIndices += i.ToString() + " "; // Индекс добавляется к строке с пробелом
38             oddCount++; // Увеличиваем счетчик нечетных чисел
39         }
40     } else {
41         hasNonNumeric = true; // Установка флага если, значение не числовое
42     }
43 }
44
45 // Проверка наличия нечетных чисел
46 if (oddCount > 0) {
47     this->NomeraResult->Text = oddIndices; // Вывод индексов
48 }
49 else {
50     this->NomeraResult->Text = "Увы их нет :0"; // Сообщение о том, что нечетных элементов нет, мда
51     ← треш
52 }
53
54 // Проверка на наличие нечисловых значений
55 if (hasNonNumeric) {
56     errorProvider1->SetError(NomeraResult, "Присутствуют нечисловые значения в массиве. Возможны
57     ← неточности");
58 } else {
59     errorProvider1->SetError(NomeraResult, ""); // Очищение сообщения об ошибки, если все норм
60 }
61 //
```

Обработка табличных данных. Часть 2

На нажатие кнопки «Выполнить» установлено выполнение следующего кода:

```

1 private: System::Void Execute_Click(System::Object^ sender, System::EventArgs^ e) {
2
3     // Обработка неккоректных данных в таблице
4     if (this->DataMassivInPut->RowCount == 0) {
5         errorProvider1->SetError(DataMassivInPut, " Пустая таблица. Нужно создать столбцы и строки ");
```

```

6     return;
7 }
8 int number;
9 bool flag = true;
10
11 // Определение минимальные и максимальные значения для int, да весьма грубо но работает
12 const int MinV = -2147483648;
13 const int MaxV = 2147483647;
14 //
15
16 for (int i = 0; i < DataMassivInPut->RowCount; i++) {
17     for (int j = 0; j < DataMassivInPut->ColumnCount; j++) {
18         // Попытка преобразовать значение в строке в число
19         bool res =
20             ↳ Int32::TryParse(System::Convert::ToString(this->DataMassivInPut->Rows[i]->Cells[j]->Value),
21             ↳ number);
22
23         // Если преобразование не удалось или число выходит за пределы int
24         if (!res || number < MinV || number > MaxV) {
25             errorProvider1->SetError(DataMassivInPut, "В таблице есть некорректные значения! Введите
26             ↳ целые числа в диапазоне int, чтобы продолжить работу.");
27             this->DataMassivInPut->Rows[i]->Cells[j]->Value = "";
28             flag = false;
29         }
30     }
31 }
32
33 if (flag == false) {
34     return;
35 }
36 //
37
38 int minRowIndex = -1, maxRowIndex = -1;
39 int minColIndex = -1, maxColIndex = -1;
40
41 // Поиск первого минимального и максимального элемента
42 for (int col = 0; col < DataMassivInPut->ColumnCount; col++) {
43     for (int row = 0; row < DataMassivInPut->RowCount; row++) {
44         if (DataMassivInPut->Rows[row]->Cells[col]->Value != nullptr &&
45             DataMassivInPut->Rows[row]->Cells[col]->Value->ToString() != "") {
46
47             int value;
48             if (Int32::TryParse(DataMassivInPut->Rows[row]->Cells[col]->Value->ToString(), value)) {
49                 // Поиск минимального элемента
50                 if (value < minValue) {

```

```

51         minValue = value;
52         minRowIndex = row;
53         minColIndex = col;
54     }
55     //
56
57     // Поиск максимального элемента
58     if (value > maxValue) {
59         maxValue = value;
60         maxRowIndex = row;
61         maxColIndex = col;
62     }
63     //
64 }
65 }
66 }
67 }
68 //
69
70 // Копирование данных из DataMassivInPut в DataMassivOutPut
71 DataMassivOutPut->Columns->Clear();
72 for (int i = 0; i < DataMassivInPut->ColumnCount; i++) {
73     DataMassivOutPut->Columns->Add("Column" + i, "" + (i + 1));
74 }
75
76 for (int row = 0; row < DataMassivInPut->RowCount; row++) {
77     array<Object^>^ rowData = gcnew array<Object^>(DataMassivInPut->ColumnCount);
78     for (int col = 0; col < DataMassivInPut->ColumnCount; col++) {
79         rowData[col] = DataMassivInPut->Rows[row]->Cells[col]->Value;
80     }
81     DataMassivOutPut->Rows->Add(rowData);
82 }
83 //
84
85 // Поиск первого минимального и максимального элементов в DataMassivOutPut
86 for (int col = 0; col < DataMassivOutPut->ColumnCount; col++) {
87     for (int row = 0; row < DataMassivOutPut->RowCount; row++) {
88         Object^ currentValue = DataMassivOutPut->Rows[row]->Cells[col]->Value;
89
90         Object^ minValue = DataMassivOutPut->Rows[0]->Cells[0]->Value;
91         Object^ maxValue = DataMassivOutPut->Rows[0]->Cells[0]->Value;
92
93         for (int col = 0; col < DataMassivOutPut->ColumnCount; col++) {
94             for (int row = 0; row < DataMassivOutPut->RowCount; row++) {
95                 Object^ currentValue = DataMassivOutPut->Rows[row]->Cells[col]->Value;
96
97                 // Проверка на минимальное значение
98                 if (currentValue != nullptr && Convert::.ToDouble(currentValue) < Convert::.ToDouble(minValue)) {

```

```

99         minValue = currentValue;
100        minRowIndex = row;
101        minColIndex = col;
102    }
103    //
104
105    // Проверка на максимальное значение
106    if (currentValue != nullptr && Convert::.ToDouble(currentValue) > Convert::.ToDouble(maxValue)) {
107        maxValue = currentValue;
108        maxRowIndex = row;
109        maxColIndex = col;
110    }
111    //
112    }
113    }
114    }
115    }
116    //
117
118    // Проверка, что минимальный и максимальный элементы найдены и что они находятся в разных
119    // столбцах
120    if (minRowIndex != -1 && maxRowIndex != -1 && minColIndex != maxColIndex) {
121        // Обмен значениями в найденных столбцах
122        for (int row = 0; row < DataMassivOutPut->RowCount; row++) {
123            auto tempValue = DataMassivOutPut->Rows[row]->Cells[minColIndex]->Value;
124            DataMassivOutPut->Rows[row]->Cells[minColIndex]->Value =
125                DataMassivOutPut->Rows[row]->Cells[maxColIndex]->Value;
126            DataMassivOutPut->Rows[row]->Cells[maxColIndex]->Value = tempValue;
127        }
128    }

```

Файловые диалоги и работа с файлами

На нажатие кнопки «Выбрать нужную гостиницу» установлено выполнение следующего кода:

```

1 private: System::Void ChoiseRaceBut_Click(System::Object^ sender, System::EventArgs^ e) {
2
3     errorProvider1->Clear();
4     errorProvider2->Clear();
5     this->Table2->RowCount = 1;
6     for (int i = 0; i < this->Table2->RowCount; i++) {
7         for (int j = 0; j < this->Table2->ColumnCount; j++) {
8             this->Table2->Rows[i]->Cells[j]->Value = nullptr;
9         }
10    }
11    if (ChoiceRace->Text == "") {

```

```

12     errorProvider1->SetError(ChoiceRace, "Сначала введите название гостиницы");
13     return;
14 }
15 int n = this->Table1->RowCount;
16 int m = this->Table1->ColumnCount;
17 for (int i = 0; i < n - 1; i++) {
18     for (int j = 0; j < m; j++) {
19         if (this->Table1->Rows[i]->Cells[j]->Value == nullptr) {
20             errorProvider1->SetError(this->Table1, "В строке " + Convert::ToString(i + 1) + " отсутствую
21             ↪ какие-то данные");
22             return;
23         }
24     }
25 }
26 for (int i = 0; i < n; i++) {
27     if (this->Table1->Rows[i]->Cells[6]->Value == ChoiceRace->Text) {
28         this->Table2->Rows->Add(1);
29     }
30 }
31 for (int i = 0; i < n; i++) {
32     if (System::Convert::ToString(this->Table1->Rows[i]->Cells[3]->Value) == ChoiceRace->Text) {
33         for (int k = 0; k < this->Table2->RowCount; k++) {
34             for (int j = 0; j < m; j++) {
35                 this->Table2->Rows[k]->Cells[j]->Value = this->Table1->Rows[i]->Cells[j]->Value;
36             }
37         }
38     }
39 }

```

На нажатие кнопки «Сохранить информацию о туристе» установлено выполнение следующего кода:

```

1 private: System::Void SaveInforPass_Click(System::Object^ sender, System::EventArgs^ e) {
2     errorProvider1->Clear();
3     errorProvider2->Clear();
4     if (ChoiceRace->Text != "") {
5         System::IO::Stream^ MyStream;
6         if (this->saveFileDialog2->ShowDialog() == System::Windows::Forms::DialogResult::OK) {
7             if ((MyStream = this->saveFileDialog2->OpenFile()) != nullptr) {
8                 System::IO::StreamWriter^ sw = gcnew System::IO::StreamWriter(MyStream);
9                 String^ str = "";
10                int n = this->Table2->RowCount;
11                for (int i = 0; i < n - 1; i++) {
12                    str = "";
13                    for (int j = 0; j < this->Table2->ColumnCount; j++) {
14                        str += this->Table2->Rows[i]->Cells[j]->Value + " ";
15                    }
16                }
17            }
18        }
19    }
20 }

```

```

16         sw->WriteLine(str);
17     }
18     str = "";
19     for (int j = 0; j < this->Table2->ColumnCount; j++) {
20         str += this->Table2->Rows[n - 1]->Cells[j]->Value + " ";
21     }
22     sw->Write(str);
23     sw->Close();
24 }
25 }
26 }
27 else {
28     errorProvider2->SetError(ChoiceRace, "Сначала выберите гостиницу");
29 }
30
31 } ///

```

Использование коллекций в Windows Forms

На нажатие кнопки «Добавить элемент» установлено выполнение следующего кода:

```

1 private: System::Void AddEl_Click(System::Object^ sender, System::EventArgs^ e) {
2
3     q.Clear();
4     errorProvider2->Clear();
5     System::String^ str = System::Convert::ToString(this->OutPutQu->Text);
6     System::String^ chis = System::Convert::ToString(this->NewElIn->Text);
7     int num;
8     bool _res = Int32::TryParse(chis, num);
9     if (chis == "") {
10         errorProvider2->SetError(AddEl, "Введите число");
11         return;
12     }
13     else if (!_res) {
14         errorProvider2->SetError(NewElIn, "Введено не число");
15         return;
16     }
17     else {
18         int x, at, pos = 0;
19         at = str->IndexOf(" "); //позиция первого пробела
20         while (at != -1) { //пока находим пробел
21             System::String^ str1 = str->Substring(pos, at - pos); //считываем число
22             pos = at + 1;
23             bool res = Int32::TryParse(str1, x);
24             q.Enqueue(x);
25             at = str->IndexOf(" ", pos); //ищем след пробел
26         }
27         q.Enqueue(num);

```

```

28 System::String^ str2 = "";
29 System::Collections::Generic::Stack<int> buf;
30 while (q.Count) {
31     buf.Push(q.Peek());
32     q.Dequeue();
33 }
34 while (buf.Count) {
35     q.Enqueue(buf.Peek());
36     str2 += System::Convert::ToString(buf.Peek()) + " ";
37     buf.Pop();
38 }
39
40 this->OutPutQu->Text = str2;
41 }
42 }
```

На нажатие кнопки «Новая очередь» установлено выполнение следующего кода:

```

1 private: System::Void NewQ_Click(System::Object^ sender, System::EventArgs^ e) {
2
3
4     // Очистка текстового поля для новой очереди
5     NewQOut->Text = String::Empty; // Предположим, что у нас есть текстовое поле для вывода новой
     → очереди
6
7     // Получение элементов из OutPutQu
8     array<String^>^ elements = OutPutQu->Text->Split( ' ' );
9
10    // Переменные для хранения максимального значения и нового элемента
11    int maxElement = Int32::MinValue;
12    String^ newElement = NewElIn->Text->Trim(); // Получаем новый элемент из текстового поля
13
14    // Проверка, является ли новый элемент целым числом
15    int newValue;
16    if (!Int32::TryParse(newElement, newValue)) {
17        MessageBox::Show("Введите корректное целое число для нового элемента.", "Ошибка",
     →     MessageBoxButtons::OK, MessageBoxIcon::Error);
18        return; // Выход из функции, если ввод некорректен
19    }
20
21    // Поиск максимального элемента
22    for each (String ^ element in elements) {
23        int number;
24        if (Int32::TryParse(element->Trim(), number)) {
25            if (number > maxElement) {
26                maxElement = number; // Обновляем максимальный элемент
27            }
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42 }
```

```

28     }
29 }
30
31 // Создание новой очереди с вставкой нового элемента после максимальных
32 String^ newQueue = String::Empty;
33 for each (String ^ element in elements) {
34     newQueue += element->Trim() + " "; // Добавляем текущий элемент в новую очередь
35     int number;
36     if (Int32::TryParse(element->Trim(), number) && number == maxElement) {
37         newQueue += newElement + " "; // Вставляем новый элемент после максимального
38     }
39 }
40
41
42 // Удаление первого элемента (если он есть)
43 int firstCommaIndex = newQueue->IndexOf(' ');
44 if (firstCommaIndex != -1) {
45     newQueue = newQueue->Substring(firstCommaIndex + 1); // Удаляем все символы до первого пробела
46 }
47
48 // Вывод новой очереди в текстовое поле
49 NewQOut->Text = newQueue;
50
51 }

```

Приложение «ТЕСТ»

На нажатие кнопки «Следующий вопрос» установлено выполнение следующего кода:

```

1 private: System::Void btnNext_Click(System::Object^ sender, System::EventArgs^ e) {
2     //скрытие ненужных элементов формы
3     btnNext->Visible = false;
4     lblAnswer->Visible = false;
5     textBox1->Visible = false;
6     radioButton1->Visible = false;
7     radioButton2->Visible = false;
8     radioButton3->Visible = false;
9     checkBox1->Visible = false;
10    checkBox2->Visible = false;
11    checkBox3->Visible = false;
12    btnCheck->Visible = true;
13
14    //снятие выбора
15    radioButton1->Checked = false;
16    radioButton2->Checked = false;
17    radioButton3->Checked = false;
18    checkBox1->Checked = false;
19    checkBox2->Checked = false;

```

```

20 checkBox3->Checked = false;
21
22 //разрешение выбора вариантов и очистка поля
23 radioButton1->Enabled = true;
24 radioButton2->Enabled = true;
25 radioButton3->Enabled = true;
26 checkBox1->Enabled = true;
27 checkBox2->Enabled = true;
28 checkBox3->Enabled = true;
29 textBox1->Text = "";
30
31 curElem = 0; //переменная для номера текущего вопроса
32 num += 1; //кол-во вопросов
33
34 lblNum2->Text = num.ToString();
35 Random^ rand = gcnew Random();
36 while (curElem == 0) { //рандомный выбор номера вопроса из массива
37     curElem = Arr[rand->Next(0, Arr->Length)];
38 }
39 Arr[curElem - 1] = 0;
40
41 //описание каждого вопроса и вариантов ответа
42 switch (curElem)
43 {
44     case 1:
45         lblQuestion->Text = "Как сокращенно называют теранозавра рекса?";
46         radioButton1->Visible = true;
47         radioButton2->Visible = true;
48         radioButton3->Visible = true;
49         radioButton1->Text = "Король";
50         radioButton2->Text = "Аллик";
51         radioButton3->Text = "Т-рекс";
52         break;
53
54     case 2:
55         lblQuestion->Text = "В каком пероиде жил аллозавр?";
56         radioButton1->Visible = true;
57         radioButton2->Visible = true;
58         radioButton3->Visible = true;
59         radioButton1->Text = "Юриский период";
60         radioButton2->Text = "Меловой период";
61         radioButton3->Text = "Триасовый период";
62         break;
63
64     case 3:
65         lblQuestion->Text = "Какого динозавра можно назвать предком птиц?";
66         radioButton1->Visible = true;
67         radioButton2->Visible = true;

```

```

68 radioButton3->Visible = true;
69 radioButton1->Text = "Архиоптерикс";
70 radioButton2->Text = "Рамфоринх";
71 radioButton3->Text = "Птеранодон";
72 break;
73
74 case 4:
75 lblQuestion->Text = "Какой раптор жил на территории штата Юта?";
76 radioButton1->Visible = true;
77 radioButton2->Visible = true;
78 radioButton3->Visible = true;
79 radioButton1->Text = "Велоцераптор";
80 radioButton2->Text = "Ютараптор";
81 radioButton3->Text = "Австрораптор";
82 break;
83
84 case 5:
85 lblQuestion->Text = "Какой(какие) динозавр(ы) был(и) покрыт(ы) перьями?";
86 checkBox1->Visible = true;
87 checkBox2->Visible = true;
88 checkBox3->Visible = true;
89 checkBox1->Text = "Теризинозавр";
90 checkBox2->Text = "Австрораптор";
91 checkBox3->Text = "Птеранодон";
92 break;
93
94 case 6:
95 lblQuestion->Text = "Овералтор питался яйцами?";
96 radioButton1->Visible = true;
97 radioButton2->Visible = true;
98 radioButton1->Text = "Да";
99 radioButton2->Text = "Нет";
100 break;
101
102 case 7:
103 lblQuestion->Text = "Выберите плавающих динозавров";
104 checkBox1->Visible = true;
105 checkBox2->Visible = true;
106 checkBox3->Visible = true;
107 checkBox1->Text = "Ихтиозавр";
108 checkBox2->Text = "Спинозавр";
109 checkBox3->Text = "Зухомим";
110 break;
111
112 case 8:
113 lblQuestion->Text = "Самый маленький динозавр";
114 textBox1->Visible = true;
115 textBox1->ReadOnly = false;

```

```
116     break;
117
118 case 9:
119     lblQuestion->Text = "Какой динозавр питался листьями?";
120     radioButton1->Visible = true;
121     radioButton2->Visible = true;
122     radioButton3->Visible = true;
123     radioButton1->Text = "Гуанлонг";
124     radioButton2->Text = "Австрораптор";
125     radioButton3->Text = "Диплодок";
126     break;
127
128 case 10:
129     lblQuestion->Text = "Какой длизовр назван в честь языческого божества?";
130     checkBox1->Visible = true;
131     checkBox2->Visible = true;
132     checkBox3->Visible = true;
133     checkBox1->Text = "Кетцалькоатль";
134     checkBox2->Text = "Гуанлонг";
135     checkBox3->Text = "Брахиозавр";
136     break;
137
138 default:
139     break;
140 }
141 }
```

ПРИЛОЖЕНИЕ Б

CD-диск с отчетом о выполненной работе

На приложенном диске можно ознакомиться со следующими файлами:

Папка Otchet — L^AT_EX- вариант отчета о практике;

Папка Factorial — задание №1;

Папка Simple — задание №2, вариант №12;

Папка Recursion — задание №3, вариант №3;

Папка Table1 — задание №4, вариант №2;

Папка Table2 — задание №5, вариант №6;

Папка Matrix — задание №6, вариант;

Папка Collections — задание №7, вариант №8;

Папка File — задание №8, вариант №9;

Папка Test — задание №9;

Otchet.pdf — отчет о практике.