

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

УТВЕРЖДАЮ

Зав.кафедрой,

г. Саратов, к. ф.-м. н., доцент

\_\_\_\_\_ С. В. Миронов

**ОТЧЕТ О ПРАКТИКЕ**

студента 2 курса 251 группы факультета КНиИТ  
Соловьева Артема Сергеевича

вид практики: учебная (рассредоточенная)

кафедра: математической кибернетики и компьютерных наук

курс: 2

семестр: 3

продолжительность: 18 нед., с 02.09.24 г. по 12.01.25 г.

Руководитель практики от университета,

доцент, к. ф.-м. н.

\_\_\_\_\_

М. И. Сафрончик

Руководитель практики от организации (учреждения, предприятия),

доцент, к. ф.-м. н.

\_\_\_\_\_

М. И. Сафрончик

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

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	6
1 Вычисление факториала .....	7
1.1 Условие задания .....	7
1.2 Вид формы в конструкторе .....	7
1.3 Таблица с описанием переименованных элементов формы .....	7
1.4 Примеры правильной и неправильной работы .....	8
1.5 Примеры исходного кода .....	9
2 Простые вычисления .....	11
2.1 Условие задания .....	11
2.2 Вид формы в конструкторе .....	11
2.3 Таблица с описанием переименованных элементов формы .....	12
2.4 Примеры правильной и неправильной работы .....	12
2.5 Примеры исходного кода .....	14
3 Рекурсивные вычисления .....	15
3.1 Условие задания .....	15
3.2 Вид формы в конструкторе .....	16
3.3 Таблица с описанием переименованных элементов формы .....	16
3.4 Примеры правильной и неправильной работы .....	17
3.5 Примеры исходного кода .....	18
4 Обработка табличных данных. Часть 1 .....	20
4.1 Условие задания .....	20
4.2 Вид формы в конструкторе .....	21
4.3 Таблица с описанием переименованных элементов формы .....	22
4.4 Примеры правильной и неправильной работы .....	23
4.5 Примеры исходного кода .....	25
5 Обработка табличных данных. Часть 2 .....	27
5.1 Условие задания .....	27
5.2 Вид формы в конструкторе .....	28
5.3 Таблица с описанием переименованных элементов формы .....	28
5.4 Примеры правильной и неправильной работы .....	30
5.5 Примеры исходного кода .....	31
6 Матричный калькулятор .....	32
6.1 Условие задания .....	32

6.2	Вид формы в конструкторе .....	32
6.3	Таблица с описанием переименованных элементов формы .....	33
6.4	Примеры правильной и неправильной работы .....	34
6.5	Примеры исходного кода .....	36
7	Использование коллекций .....	37
7.1	Условие задания .....	37
7.2	Вид формы в конструкторе .....	37
7.3	Таблица с описанием переименованных элементов формы .....	38
7.4	Примеры правильной и неправильной работы .....	39
7.5	Примеры исходного кода .....	40
8	Работа с файлами .....	42
8.1	Условие задания .....	42
8.2	Вид формы в конструкторе .....	43
8.3	Таблица с описанием переименованных элементов формы .....	43
8.4	Примеры правильной и неправильной работы .....	44
8.5	Примеры исходного кода .....	46
9	Приложение ТЕСТ .....	47
9.1	Условие задания .....	47
9.2	Вид формы в конструкторе .....	47
9.3	Таблица с описанием переименованных элементов формы .....	49
9.4	Примеры правильной и неправильной работы .....	49
9.5	Примеры исходного кода .....	52
	ЗАКЛЮЧЕНИЕ .....	54
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	55
	Приложение А Фрагменты кода программы «Вычисление факториала» ...	56
	Приложение Б Фрагменты кода программы «Простые вычисления» .....	57
	Приложение В Фрагменты кода программы «Рекурсивные вычисления» ...	58
	Приложение Г Фрагменты кода программы «Обработка табличных дан- ных. Часть 1» .....	59
	Приложение Д Фрагменты кода программы «Обработка табличных дан- ных. Часть 2» .....	60
	Приложение Е Фрагменты кода программы «Матричный калькулятор» ....	62
	Приложение Ж Фрагменты кода программы «Использование коллекций в Windows Forms» .....	63

Приложение З	Фрагменты кода программы «Работа с файлами» .....	64
Приложение И	Фрагменты кода программы «Приложение тест» .....	65
Приложение К	Флешка с отчетом о выполненной работе .....	67

## ВВЕДЕНИЕ

Целью практики является освоение механизма построения оконного интерфейса приложений в среде Microsoft Visual Studio на языке C++/CLI с использованием .NET Framework и Windows Forms. В результате прохождения практики должны быть отработаны следующие навыки:

- Создание нового проекта;
- Добавление и настройка элементов управления;
- проверка пользовательского ввода данных для решения поставленной задачи, обработка ошибок ввода;
- разработка алгоритма решения поставленной задачи с использованием оконного интерфейса;
- тестирование приложения;
- документирование разработанного кода.

# 1 Вычисление факториала

## 1.1 Условие задания

Разработать приложение для вычисления факториала по приведенному примеру.

Приложение должно содержать следующие компоненты:

1. Заголовок формы должен отражать суть задания.
2. Все элементы формы должны быть внятно подписаны (кнопки подписаны, у текстового поля должно быть написано, для чего оно нужно и т. д.)
3. В коде должны быть комментарии и отступы (код должен быть легко читаем).
4. В коде программы все элементы формы должны быть переименованы (btnName - для кнопок, lblName - для ссылок, txtName - для текстового поля и т. д.) Наименования должны быть понятными.
5. Приложение должно корректно работать (выводить ответ или ошибку с соответствующим сообщением) для следующих данных: ввод буквы, ввод отрицательного числа, ввод нуля, ввод положительного числа ( $< 10$ ), ввод большого положительного числа. После вывода ошибок при вводе корректных данных поля ошибок должны очищаться.

## 1.2 Вид формы в конструкторе

Создано окно приложения, содержащее два элемента `TextBox`, два элемента `Label` и один элемент `Button`. Вид окна представлен на рисунке 1 [1].

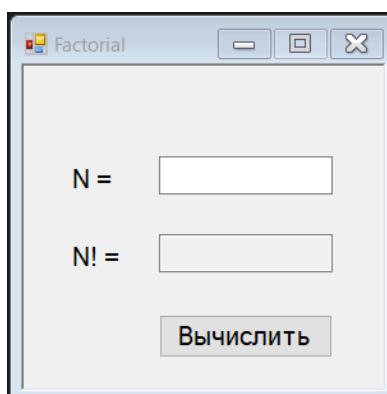


Рисунок 1 – Окно приложения «factorial» открытое в конструкторе

## 1.3 Таблица с описанием переименованных элементов формы

Все измененные элементы формы указаны в таблице 1.

Таблица 1 – Значения атрибутов элементов в приложении factorial

Описание элементов формы	Список измененных атрибутов	Новое значение атрибута
Форма MyForm	FormBorderStyle	Fixed3D
	Text	Factorial
	MaximizeBox	False
TextBox ввод числа	Name	input
TextBox вывод факториала	Name	output
	ReadOnly	True
Label у поля ввода	Name	lblin
	Text	N =
Label у поля вывода	Name	lblout
	Text	N! =
Кнопка "Вычислить"	Name	solve
	Text	Вычислить

#### 1.4 Примеры правильной и неправильной работы

После запуска программы на экране появляется окно на рисунке 2.

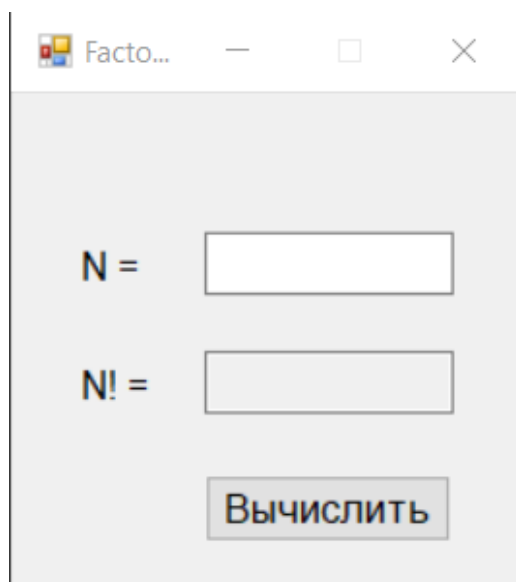


Рисунок 2 – Запуск программы

При вводе числа в поле ввода и нажатии на кнопку "Вычислить" (на рисунке 3)



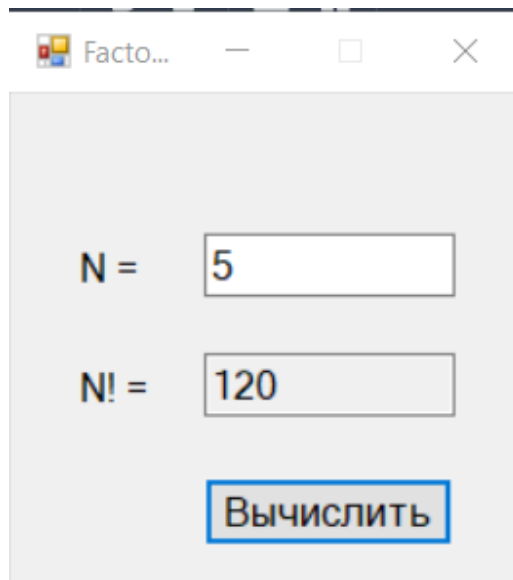


Рисунок 3 – Вычисление факториала

При попытке ввода не числа, программа выведет ошибку (на рисунке 4)

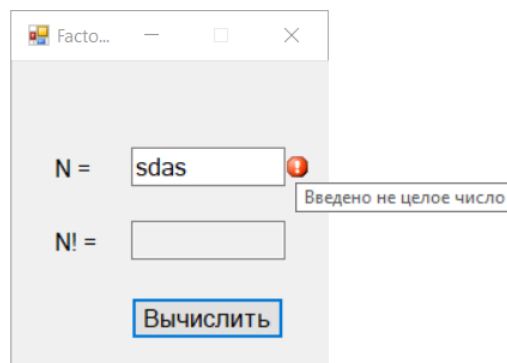


Рисунок 4 – Ошибка формата ввода

## 1.5 Примеры исходного кода

Код, выполняющийся при нажатии на кнопку "Вычислить".

```

1 private: System::Void solve_Click(System::Object^ sender, System::EventArgs^ e) {
2     this->output->Text = "";
3     errorProvider1->SetError(input, String::Empty);
4     long long InputNumber;
5     bool result = Int64::TryParse(this->input->Text, InputNumber); //переводим строку из
    ⇨   TextBox число
6     if (!result) { //двели не число
7         errorProvider1->SetError(input, "Введено не целое число");
8     }
9     else { //число
10        if (InputNumber > 20) {

```

```

11     this->output->Text = "Слишком большое число";
12 }
13 else {
14     long long OutputNumber = fact(InputNumber); //результат
15     if (OutputNumber == -1) { //отрицательное число
16         errorProvider1->SetError(input, "Введено отрицательное число");
17     }
18     else { //все нормально
19         this->output->Text = System::Convert::ToString(OutputNumber); //записываем в
        ↪ поле вывода
20     }
21 }
22 }
23 }

```

Другие фрагменты кода расположены в приложении А.

## 2 Простые вычисления

### 2.1 Условие задания

Выполнить задание. Номер варианта задает преподавателем.

Проверить работу созданного приложения на приведенных тестовых примерах. Тесты сделаны на Visual Studio 2008, поэтому обращайте внимание только на первые шесть цифр после запятой.

Приложение должно содержать следующие компоненты:

1. Заголовок формы должен отражать суть задания.
2. Все элементы формы должны быть внятно подписаны (кнопки подписаны, у текстового поля должно быть написано, для чего оно нужно и т. д.)
3. В коде должны быть комментарии и отступы (код должен быть легко читаем).
4. Должна быть проверка ошибок - ввод не числа, ввод числа, находящегося за пределами ОДЗ, ввод числа, принадлежащего ОДЗ.
5. Если надо ввести 2 значения, то в случае ввода букв в оба поля, ошибка должна быть у обоих полей; в случае ввода одной буквы - только у того поля, где буква.

**Вариант 4.** Смотреть на рисунке 5.

4. 
$$\frac{(x + y)^2 - x^3}{|x^2 - y|}$$

$x$	$y$	Результат
3	4	4.4
2	4	Деление на ноль
-2	4	Деление на ноль
$3f$	4	$x$ — не целое число
3	$4f$	$y$ — не целое число
$3f$	$4f$	$x$ — не целое число, $y$ — не целое число

Рисунок 5 – Задание 2 Вариант 4

### 2.2 Вид формы в конструкторе

Создано окно приложения, содержащее три элемента TextBox, три элемента Label, один элемент Button и один элемент ErrorProvider для обработки ошибок. Вид окна представлен на рисунке 6 [2].

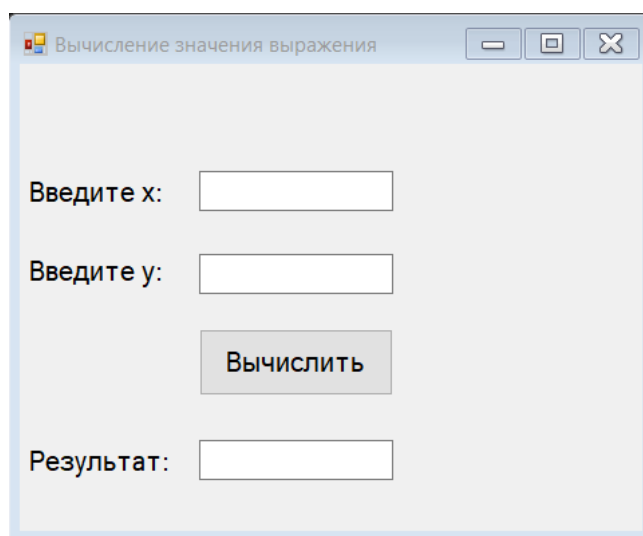


Рисунок 6 – Окно приложения «Вычисление значения выражения» открытое в конструкторе

## 2.3 Таблица с описанием переименованных элементов формы

Все измененные элементы формы указаны в таблице 2.

Таблица 2 – Значения атрибутов элементов в приложении «Вычисление значения выражения»

Описание элементов формы	Список измененных атрибутов	Новое значение атрибута
Форма MyForm	Text	Вычисление значений выражения
TextBox ввода x	Name	x_input
TextBox ввода y	Name	y_input
TextBox вывод результата	Name	output
Label у поля ввода x	Name	x_lbl
	Text	Введите x:
Label у поля ввода y	Name	y_lbl
	Text	Введите y:
Label у поля вывода	Name	res_lbl
	Text	Результат
Кнопка "Вычислить"	Name	solvebtn
	Text	Вычислить

## 2.4 Примеры правильной и неправильной работы

После запуска программы на экране появляется окно на рисунке 7.

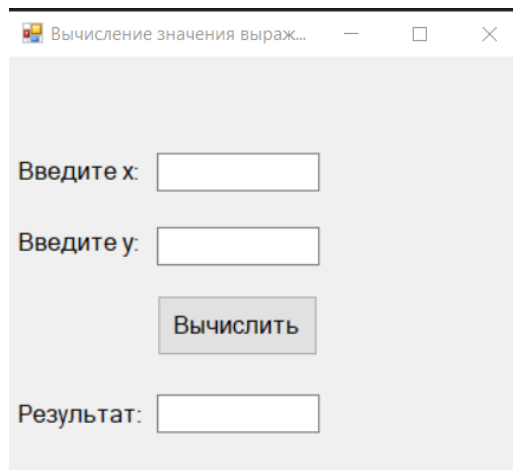


Рисунок 7 – Запуск программы

При вводе  $x$  и  $y$  в соответствующие поля и нажатии на кнопку "Вычислить"(на рисунке 8)

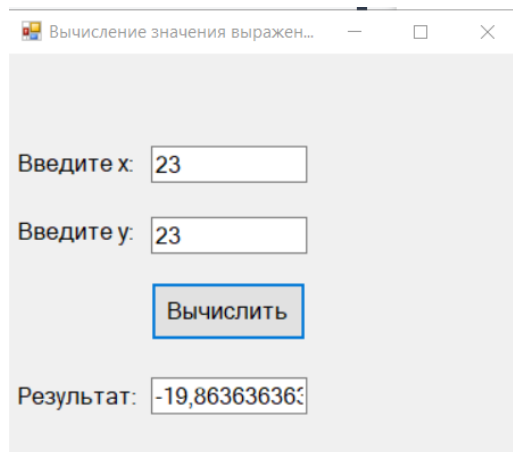


Рисунок 8 – Вычисление выражения

При попытке вычисления выражения такого, что  $x^2 - y = 0$ , вылезает ошибка «деление на ноль» в поле вывода. (на рисунке 9).

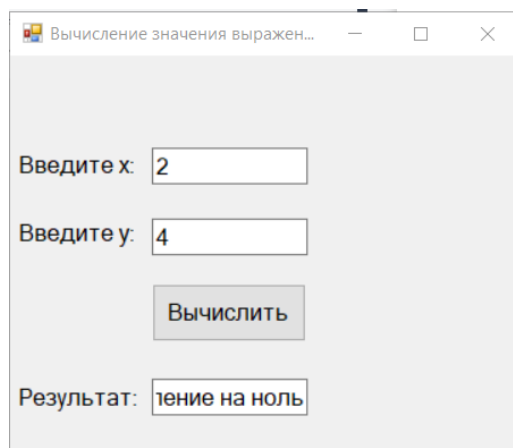


Рисунок 9 – Ошибка деления на ноль

## 2.5 Примеры исходного кода

Начало функции, выполняющейся при нажатии на кнопку «Вычислить».

```
1 private: System::Void solvebtn_Click(System::Object^ sender, System::EventArgs^ e) {
2     this->output->Text = ""; // Стираем поле вывода
3
4     errors->SetError(x_input, String::Empty); // обнуляем ошибки
5     errors->SetError(y_input, String::Empty);
6
7     Int64 x, y; // Переменные для считывания полей ввода
8     double result; // переменная для записи результата
9
10    bool result_x = Int64::TryParse(this->x_input->Text, x); // записываем из полей ввода
    ↪ в соответствующие переменные
11    bool result_y = Int64::TryParse(this->y_input->Text, y); // и проверяем на успешность
    ↪ выполнения парсинга
12
13    if (!result_x) { // если неудачно
14        errors->SetError(x_input, "Введено не целое число");
15    }
16    if (!result_y) {
17        errors->SetError(y_input, "Введено не целое число");
18    }
19    if (result_x && result_y) { // если удачно
20        if (x * x - y == 0) { // проверка на ОДЗ
21            this->output->Text = "Деление на ноль";
22        }
23        else {
24            result = (1.0) * ((x + y) * (x + y) - x * x * x) / (std::abs(x * x - y)); //Считаем
25            this->output->Text = System::Convert::ToString(result); //Записываем результат
26        }
27    }
28 }
```

Другие фрагменты кода расположены в приложении Б.

### 3 Рекурсивные вычисления

#### 3.1 Условие задания

Разработать приложение для выполнения своего варианта задания. Номер варианта задается преподавателем.

Проверить работу приложения на приведенных тестах.

Приложение должно содержать следующие компоненты:

1. Заголовок формы должен отражать суть задания.
2. Все элементы формы должны быть внятно подписаны (кнопки подписаны, у текстового поля должно быть написано, для чего оно нужно и т. д.)
3. В коде должны быть комментарии и отступы (код должен быть легко читаем).
4. В коде программы все элементы формы должны быть переименованы (btnName - для кнопок, lblName - для ссылок, txtName - для текстового поля и т. д.) Наименования должны быть понятными.
5. Приложение должно корректно работать (выводить ответ или ошибку с соответствующим сообщением) для следующих данных: ввод буквы, ввод отрицательного числа, ввод нуля, ввод положительного числа ( $< 10$ ), ввод большого положительного числа. После вывода ошибок при вводе корректных данных поля ошибок должны очищаться.

**Вариант 5.** Смотреть на рисунке 10.

5. Создать рекурсивную функцию, которая для заданных целых  $n$  и  $m$  ( $1 \leq n \leq m$ ) вычисляет количество всех различных размещений из  $m$  элементов по  $n$ , используя следующее соотношение:

$$\begin{cases} A_m^1 = m \\ A_m^n = (m - n + 1)A_m^{n-1} \end{cases}$$

$n$	$m$	Результат
5	8	6720
9	15	1816214400
2	-4	$m < 1$
-2	4	$n < 1$
-2	-4	$m < 1$ и $n < 1$
4	2	$n > m$
3f	4	$n$ — не целое число
3	4f	$m$ — не целое число
3f	4f	$n$ — не целое число, $m$ — не целое число

Рисунок 10 – Задание 3 Вариант 5

### 3.2 Вид формы в конструкторе

Создано окно приложения, содержащее три элемента `TextBox`, два элемента `Label`, один элемент `Button`. Один из элементов `TextBox` содержит атрибут `Multiline`. Вид окна представлен на рисунке 11 [3].

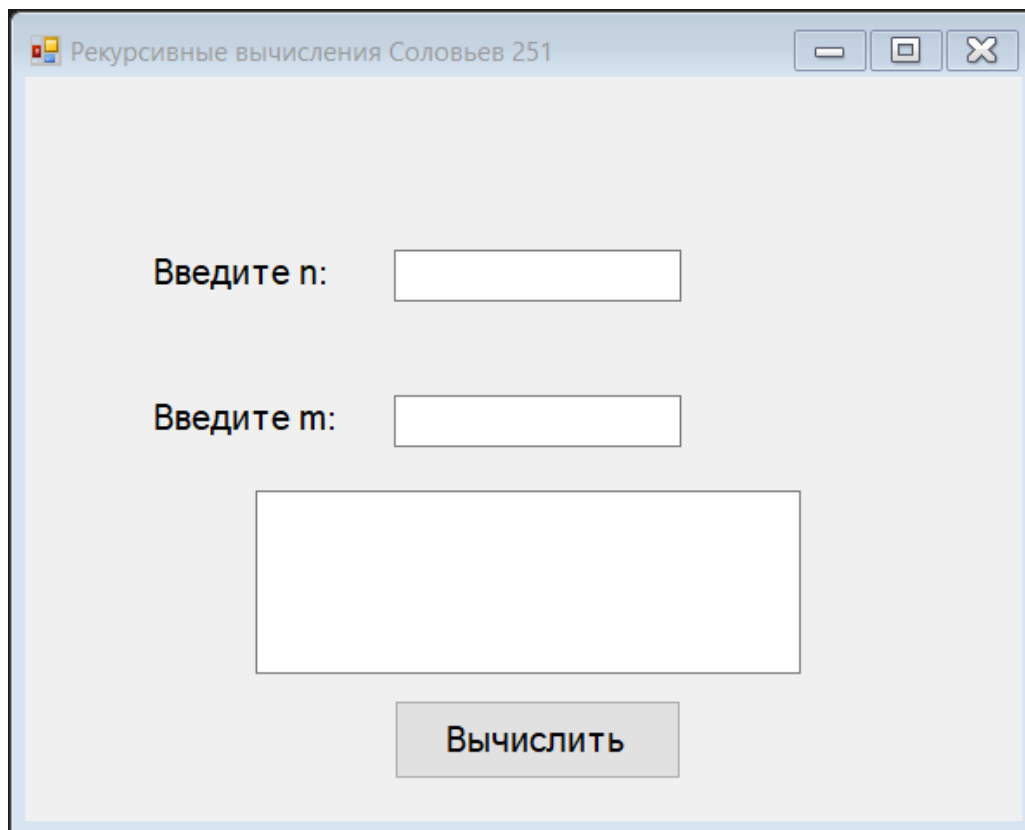


Рисунок 11 – Окно приложения «Рекурсивные вычисления» открытое в конструкторе

### 3.3 Таблица с описанием переименованных элементов формы

Все измененные элементы формы указаны в таблице 3.



Таблица 3 – Значения атрибутов элементов в приложении «Рекурсивные вычисления»

Описание элементов формы	Список измененных атрибутов	Новое значение атрибута
Форма MyForm	Text	Рекурсивные вычисления Соловьев 251
TextBox ввода n	Name	n_input
TextBox ввода m	Name	m_input
TextBox вывода	Name	Output
	Multiline	True
Label у поля ввода n	Name	n_inputlbl
	Text	Введите n:
Label у поля ввода m	Name	m_inputlbl
	Text	Введите m:
Кнопка "Вычислить"	Name	Solve
	Text	Вычислить

### 3.4 Примеры правильной и неправильной работы

После запуска программы на экране появляется окно на рисунке 12.

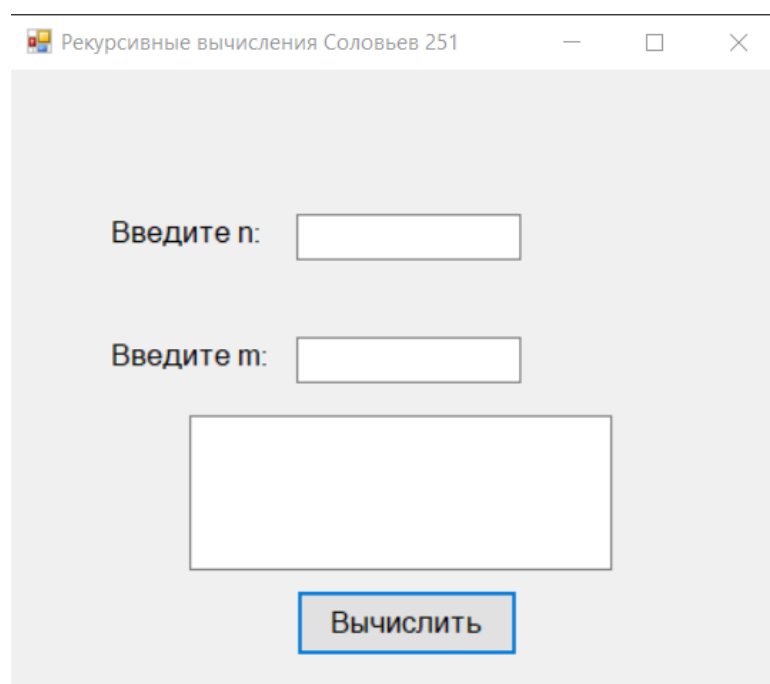


Рисунок 12 – Запуск программы

При вводе удовлетворяющих условиям  $n$  и  $m$  в соответствующие поля и нажатии на кнопку «Вычислить» (на рисунке 13).

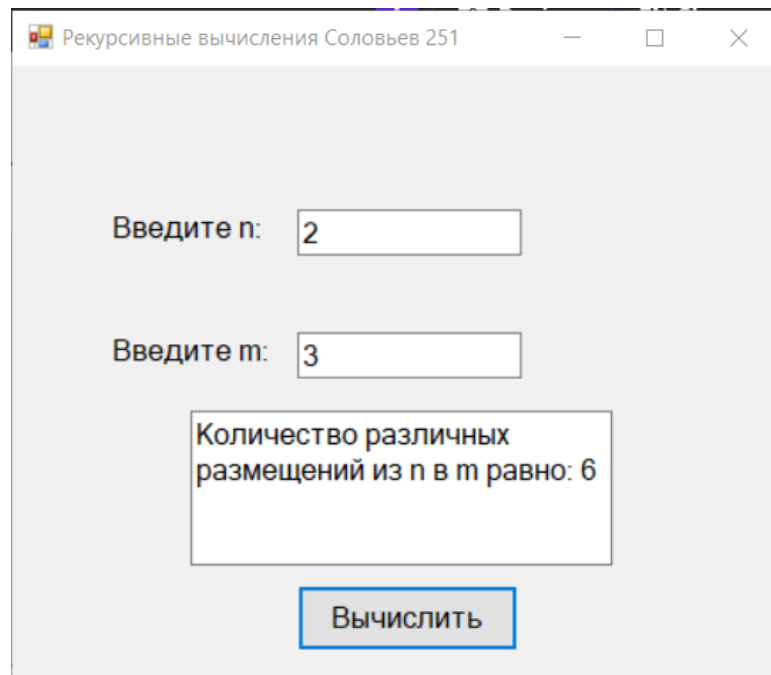


Рисунок 13 – Вычисление выражения

При попытке ввода не числа, программа выведет ошибку (на рисунке 14)

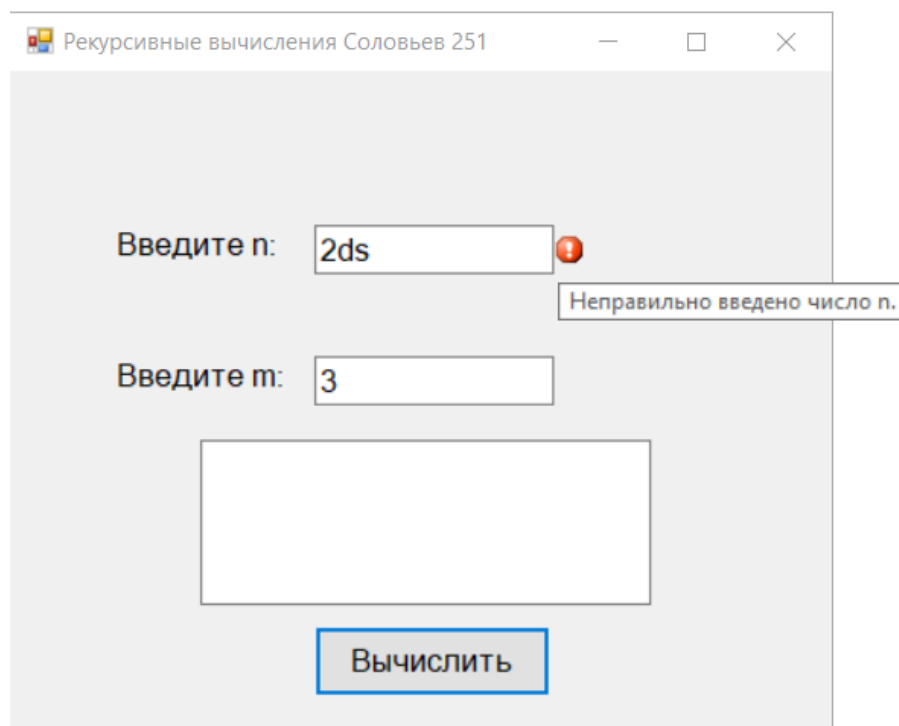


Рисунок 14 – Ошибка формата ввода

### 3.5 Примеры исходного кода

Функция вычисления числа перестановок.

```
1 long long per(long long n, long long m) {  
2     if (n == 1) return m;  
3     else return (m - n + 1) * per(n - 1, m);  
4 }
```

Другие фрагменты кода расположены в приложении В.

## 4 Обработка табличных данных. Часть 1

### 4.1 Условие задания

Создать приложение для выполнения задания. Использовать элемент формы DataGridView.

**ДИАПАЗОН [a,b] означает, что  $mas[i][j] \geq a \ \&\& \ mas[i][j] \leq b$ .**

Приложение должно выполнять следующие действия:

1. Возможность удалять и добавлять строки таблицы. Проект не должен аварийно завершаться при удалении несуществующей таблицы.
2. Проверять ввод не числовых данных как в таблицу, так и в остальные текстовые поля (если есть в задании).
3. Если есть диапазон значений [a,b], проверять, что  $a < b$ .
4. Заголовок формы должен отражать суть задания.
5. Все элементы формы должны быть внятно подписаны (кнопки подписаны, у тестового поля должно быть написано, для чего оно нужно и т. д.)
6. В коде должны быть комментарии и отступы (код должен быть легко читаем).
7. В коде программы все элементы формы должны быть переименованы (btnName - для кнопок, lblName - для ссылок, txtName - для текстового поля и т. д.) Наименования должны быть понятными.
8. Приложение должно корректно работать (выводить ответ или ошибку с соответствующим сообщением). После вывода ошибок при вводе корректных данных поля ошибок должны очищаться.

**Вариант 9.** Смотреть на рисунке 15.

9. Вывести номера элементов, кратных  $X$  и попадающих в заданный диапазон  $[a, b]$ . Если таких элементов нет, вывести сообщение об этом. Вывести минимальный четный элемент. Если таких элементов нет, вывести сообщение об этом.

Массив	$X$	$a$	$b$	Номера	Минимум
1, 2, 4, 5, 9, 4, 8	4	2	7	2, 6	2
1, 2, 5, 7, 3, 7, 3	3	4	7	Таких элементов нет	2
1, 3, 5, 7, 9, 3, 1	3	3	6	1, 5	Четных элементов нет
1, 3, 3, 5, 5, 5, 1	2	4	8	Таких элементов нет	Четных элементов нет
2, 5, 3, 8, 7, 9, 6	2	$f$	$r$	$a$ — не число, $b$ — не число	2
2, 5, 3, 8, 7, 9, 6	4	$f$	5	$a$ — не число	2
2, 5, 3, 8, 7, 9, 6	6	7	$r$	$b$ — не число	2
2, 5, 3, 8, 7, 9, 6	5	9	4	$a > b$	2
2, 5, 3, 8, 7, 9, 6	$f$	5	6	$X$ — не число	2
5, $y$ , 4, 2, 6, 8, 6	4	5	7	В таблице есть не числа	В таблице есть не числа

Рисунок 15 – Задание 4 Вариант 9

## 4.2 Вид формы в конструкторе

Создано окно приложения, содержащее пять элементов TextBox, четыре элемента Label, четыре элемента Button, один элемент gridview и один элемент ErrorProvider для обработки ошибок. Вид окна представлен на рисунке 16 [4].

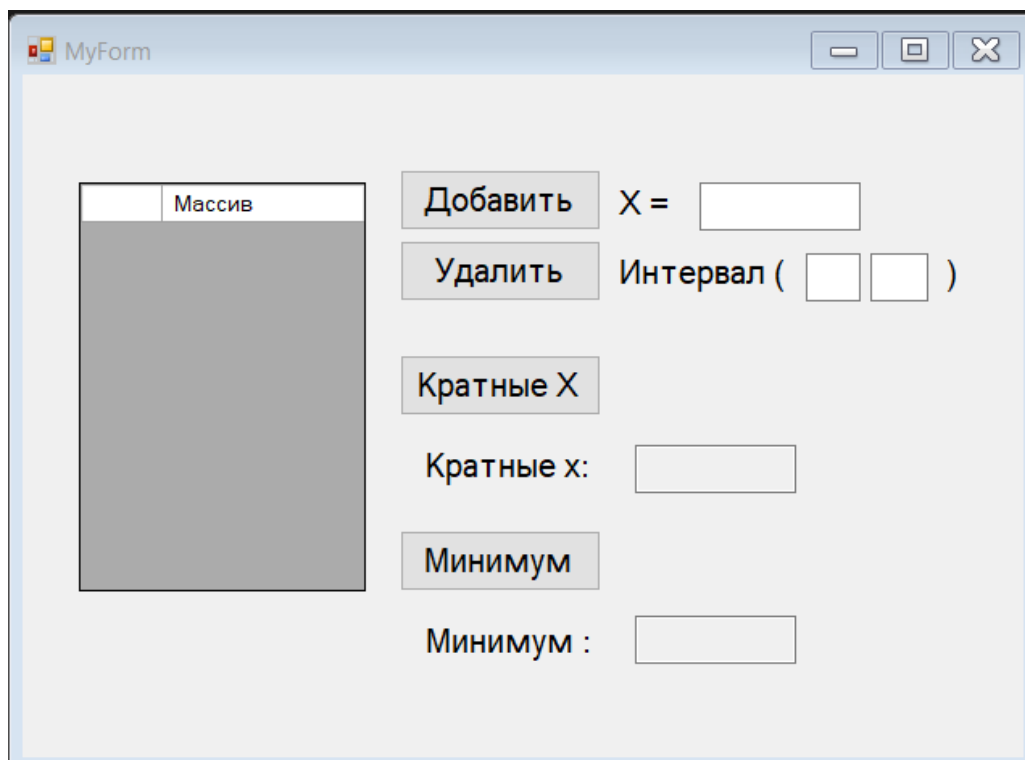


Рисунок 16 – Окно приложения «Обработка табличных данных. Часть 1» открытое в конструкторе

#### 4.3 Таблица с описанием переименованных элементов формы

Все измененные элементы формы указаны в таблице 4.

Таблица 4 – Значения атрибутов элементов в приложении «Обработка табличных данных. Часть 1»

Описание элементов формы	Список измененных атрибутов	Новое значение атрибута
Форма MyForm	Text	Обработка табличных данных. Часть 1
TextBox ввода X	Name	x_input
TextBox ввода начала интервала	Name	int_a
TextBox ввода конца интервала	Name	int_b
TextBox вывода кратных x	Name	krat_box
	ReadOnly	True
TextBox вывода минимума	Name	chet_box
	ReadOnly	True
Кнопка "Добавить"	Name	mas_add
	Text	Добавить
Кнопка "Удалить"	Name	mas_pop
	Text	Удалить
Кнопка "Кратные X"	Name	krat_btn
	Text	Кратные X
Кнопка "Минимум"	Name	min_btn
	Text	Минимум
Таблица ввода	Name	mas_grid
	AllowUserToAddRows	False
	AllowUserToDeleteRows	False

#### 4.4 Примеры правильной и неправильной работы

После запуска программы на экране появляется окно на рисунке 17.

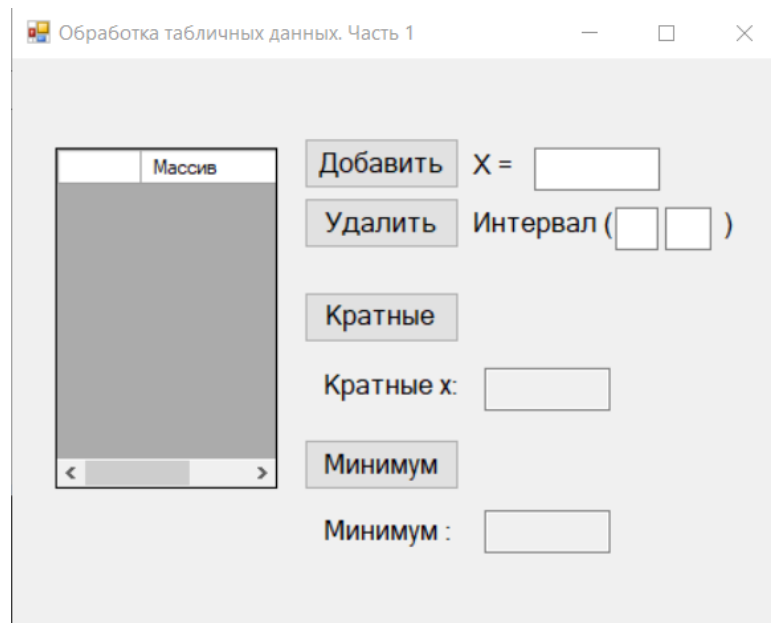


Рисунок 17 – Запуск программы

При вводе целого  $X$  в поле ввода и нажатии на кнопку «Добавить» (на рисунке 18).

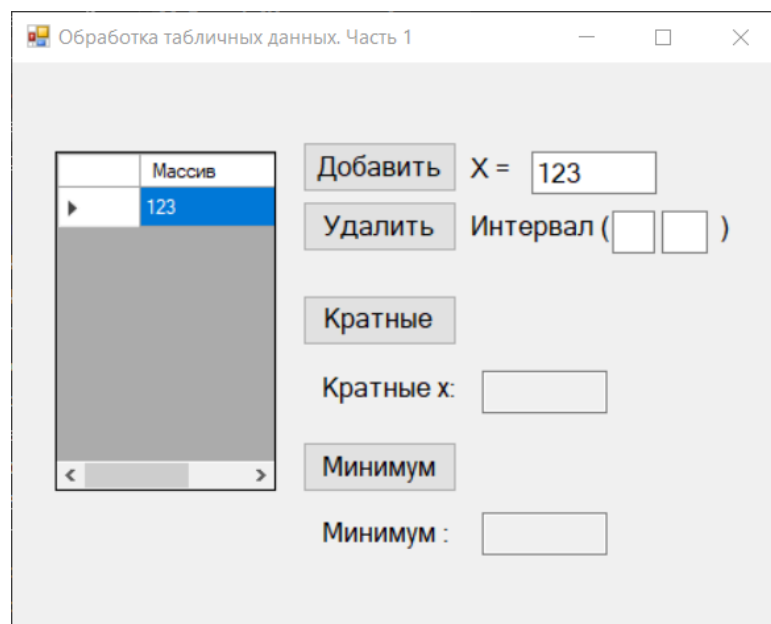


Рисунок 18 – Вычисление выражения

При попытке ввода не числа, программа выведет ошибку (на рисунке 19)



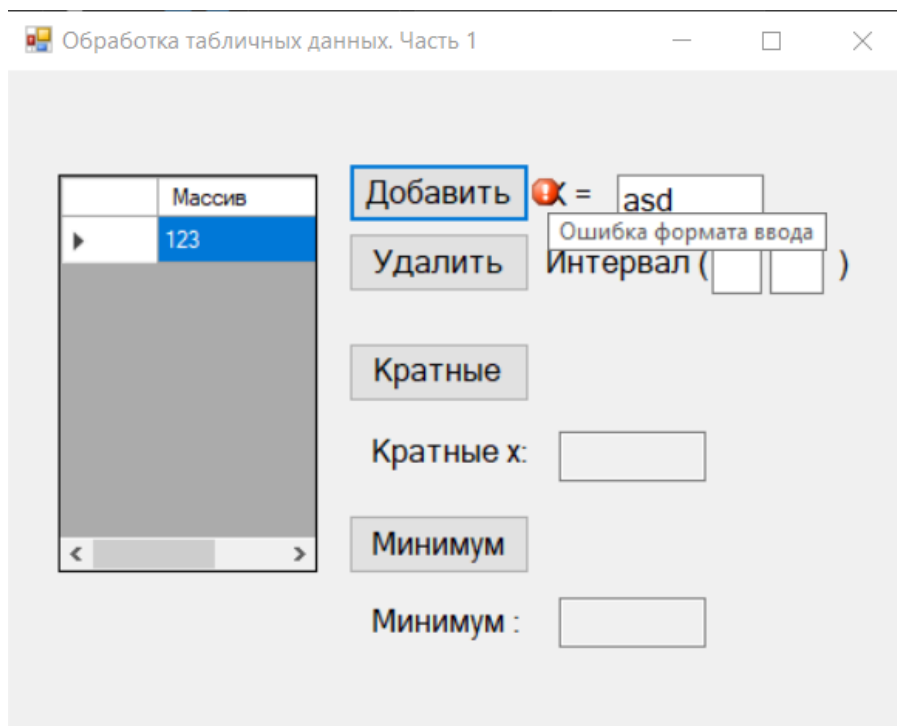


Рисунок 19 – Ошибка формата ввода

## 4.5 Примеры исходного кода

Функция при нажатии на кнопку "Кратные X".

```

1 private: System::Void krat_btn_Click(System::Object^ sender, System::EventArgs^ e) {
2     this->errorProvider1->Clear();
3     this->krat_box->Text = "";
4     long long x,y,a,b;
5     bool result = Int64::TryParse(this->x_input->Text, x);
6     bool resulta = Int64::TryParse(this->int_a->Text, a);
7     bool resultb = Int64::TryParse(this->int_b->Text, b);
8     if (!result) {
9         this->errorProvider1->SetError(this->x_input, "Неправильный формат числа X");
10        return;
11    }
12    if (!resulta || !resultb) {
13        this->errorProvider1->SetError(this->int_a, "Неправильный формат границ
        ↪ интервала");
14        return;
15    }
16    bool found_krat = false;
17    for (int i = a; i < this->mas_grid->RowCount && i < b; i++) {
18        result = Int64::TryParse(this->mas_grid->Rows[i]->Cells[0]->Value->ToString(), y);
19        if (y % x == 0) {

```

```
20     found_krat = true;
21     this->krat_box->Text += System::Convert::ToString(i+1) + ", ";
22 }
23 }
24 if (!found_krat) {
25     this->krat_box->Text = "Таких элементов нет";
26 }
27 }
```

Другие фрагменты кода расположены в приложении Г.

## 5 Обработка табличных данных. Часть 2

### 5.1 Условие задания

Разработать приложение в соответствии со своим вариантом. Номер задания выбирается преподавателем.

Проверить работу приложения на приведенных тестовых примерах.

**ЗАДАНИЕ ПРЕДПОЛАГАЕТ НАЛИЧИЕ ДВУХ ТАБЛИЦ.** В первую вводятся данные (должна быть проверка, что введены цифры). Вторую лучше сделать доступной только для чтения. В нее записывается результат.

**ФРАЗА "ЗАМЕНИТЬ СТРОКОЙ X"** предполагает, что создается либо таблица, состоящая из одной строки и туда вводятся данные, либо создается текстовое поле, куда вводится число, и вся строка таблицы заменяется этим числом.

**ДИАПАЗОН [a,b]** означает, что  $\text{mas}[i][j] \geq a \ \&\& \ \text{mas}[i][j] \leq b$ .

Приложение должно содержать следующие компоненты:

1. Заголовок формы должен отражать суть задания.
2. Все элементы формы должны быть внятно подписаны (кнопки подписаны, у текстового поля должно быть написано, для чего оно нужно и т. д.)
3. В коде должны быть комментарии и отступы (код должен быть легко читаем).
4. Таблица может быть задана двумя способами:
  - либо ввести количество строк и столбцов (тогда необходима проверка, что введено не число) и создать нужную таблицу. При изменении количества строк и столбцов, старая таблица должна быть удалена.
  - либо добавлять и удалять строки и столбцы с помощью отдельных кнопок. Проследить, чтобы приложение не завершалось аварийно (не удалять нулевую строку).
5. Должна быть проверка ошибок - ввод не числа, ввод числа, приводящего к переполнению стека или выхода результата за границы диапазона типа. В таблице также должны быть введены целые числа. В случае ошибочного ввода поля результатов должны автоматически очищаться.
6. Если данные введены корректно, но отсутствуют необходимые данные (например, надо найти сумму нечетных чисел, а в таблице только четные), то в поле результата должно быть выведено об этом сообщение.
7. В случае задач с вводом диапазона [a,b] необходима обязательная провер-

ка, что  $a < b$ .

**Вариант 12.** Смотреть на рисунке 20.

12. Все нечетные столбцы заменить столбцом  $X$ . Нумерация столбцов массива начинается с нуля.

Массив	Столбец	Результат
$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 3 & 0 & 5 \\ 1 & 0 & 3 & 0 & 5 \\ 1 & 0 & 3 & 0 & 5 \\ 1 & 0 & 3 & 0 & 5 \\ 1 & 0 & 3 & 0 & 5 \end{pmatrix}$

Рисунок 20 – Задание 5 Вариант 12

## 5.2 Вид формы в конструкторе

Создано окно приложения, содержащее два элемента `TextBox`, три элемента `Label`, шесть элементов `Button`, три элемента `gridview` (один из которых содержит атрибут `ReadOnly`) и один элемент `ErrorProvider` для обработки ошибок. Вид окна представлен на рисунке 21 [5].

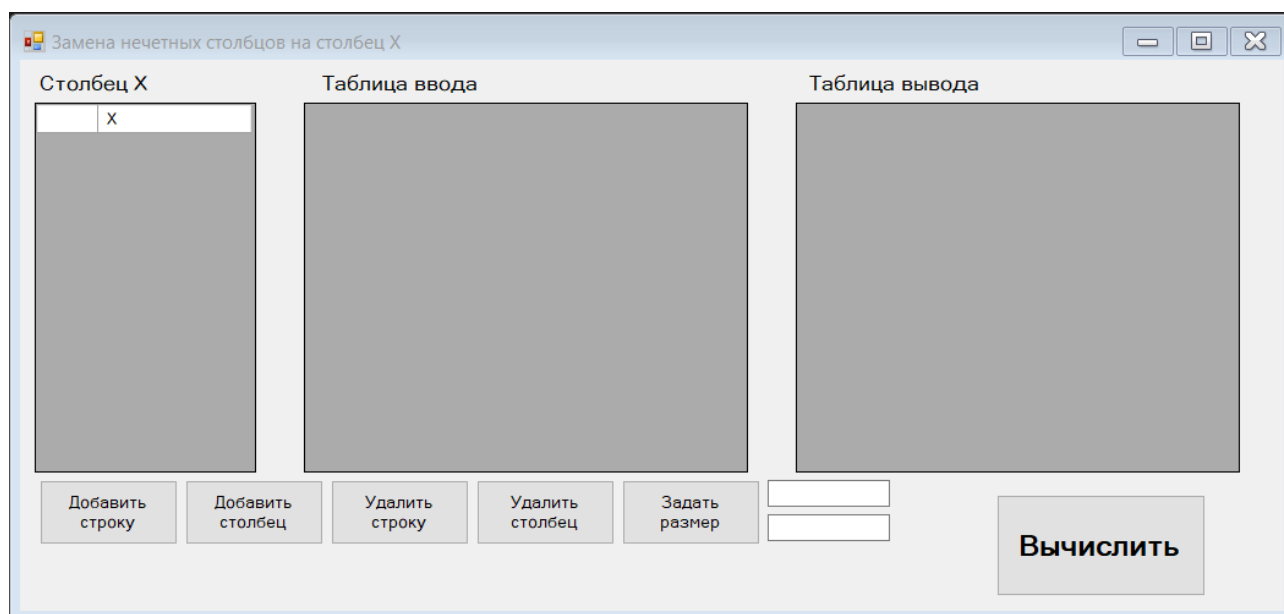


Рисунок 21 – Окно приложения «Обработка табличных данных. Часть 2» открытое в конструкторе

## 5.3 Таблица с описанием переименованных элементов формы

Все измененные элементы формы указаны в таблице 5.

Таблица 5 – Значения атрибутов элементов в приложении «Обработка табличных данных. Часть 2»

Описание элементов формы	Список измененных атрибутов	Новое значение атрибута
Форма MyForm	Text	Обработка табличных данных. Часть 2
TextBox ввода количества строк таблицы	Name	set_size_lines
TextBox ввода количества Столбцов таблицы	Name	set_size_columns
TextBox ввода конца интервала	Name	int_b
TextBox вывода кратных х	Name	krat_box
	ReadOnly	True
TextBox вывода минимума	Name	chet_box
	ReadOnly	True
Кнопка "Добавить строку"	Name	Add_line
	Text	Добавить строку
Кнопка "Добавить столбец"	Name	add_column
	Text	Добавить столбец
Кнопка "Удалить строку"	Name	remove_line
	Text	Удалить строку
Кнопка "Удалить столбец"	Name	remove_column
	Text	Удалить столбец
Кнопка "Задать размер"	Name	set_size
	Text	Задать размер
Кнопка "Вычислить"	Name	solve_btn
	Text	Вычислить
Таблица ввода столбца X	Name	X_input
	AllowUserToAddRows	False
	AllowUserToDeleteRows	False
Таблица ввода матрицы	Name	input_grid
	AllowUserToAddRows	False
	AllowUserToDeleteRows	False
Таблица вывода	Name	output_grid
	AllowUserToAddRows	False
	AllowUserToDeleteRows	False
	ReadOnly	True

## 5.4 Примеры правильной и неправильной работы

После запуска программы на экране появляется окно на рисунке 22.

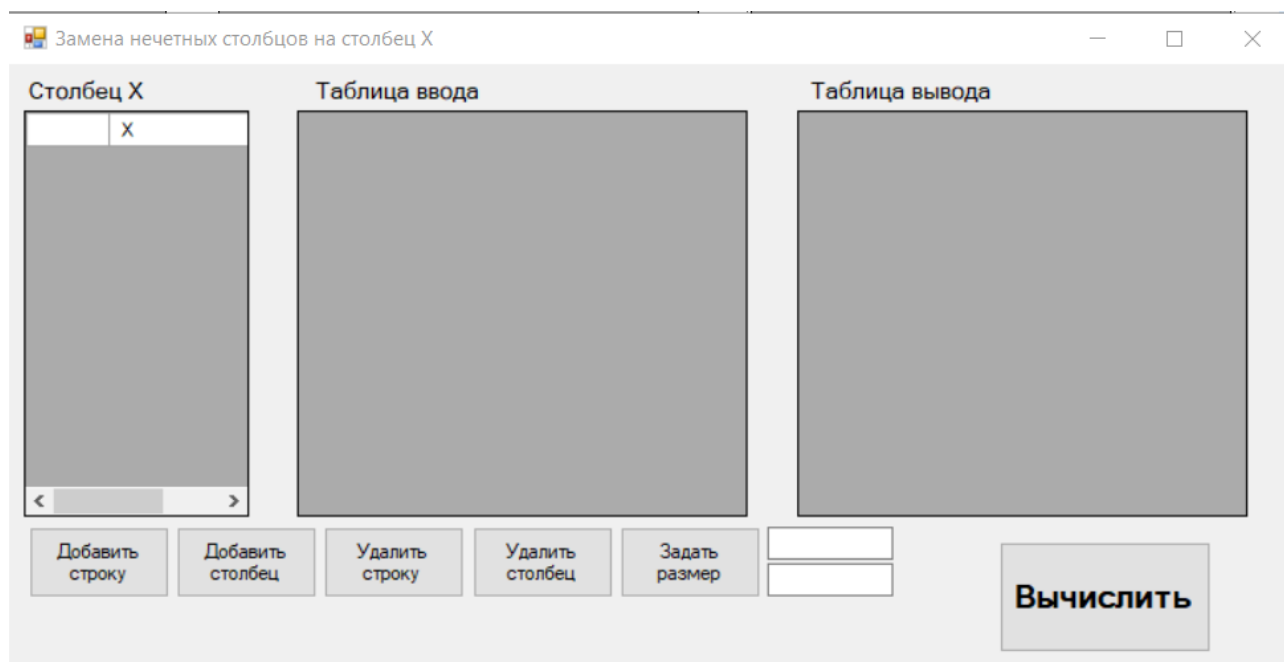


Рисунок 22 – Запуск программы

После ввода столбца X и таблицы ввода и нажатии на кнопку "Вычислить"(на рисунке 23).

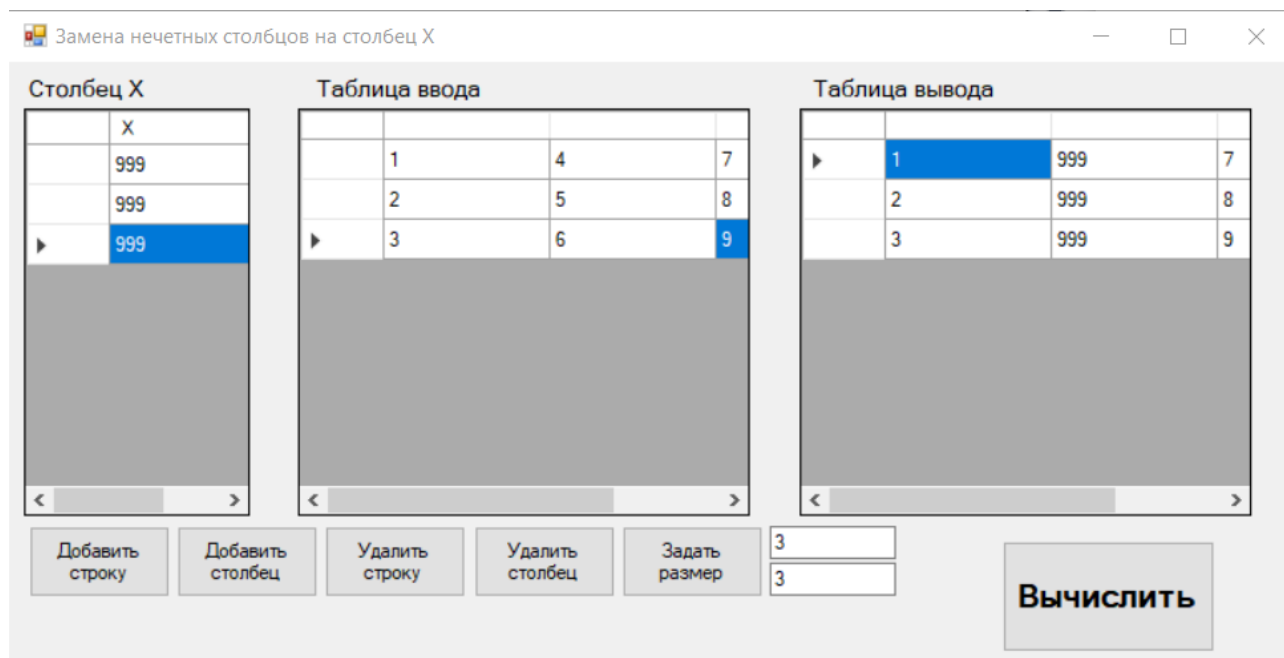


Рисунок 23 – Замена всех нечетных столбцов столбцом X (счет начнется с 0)

При попытке ввода не числа в таблицу, программа выведет ошибку (на рисунке 24)

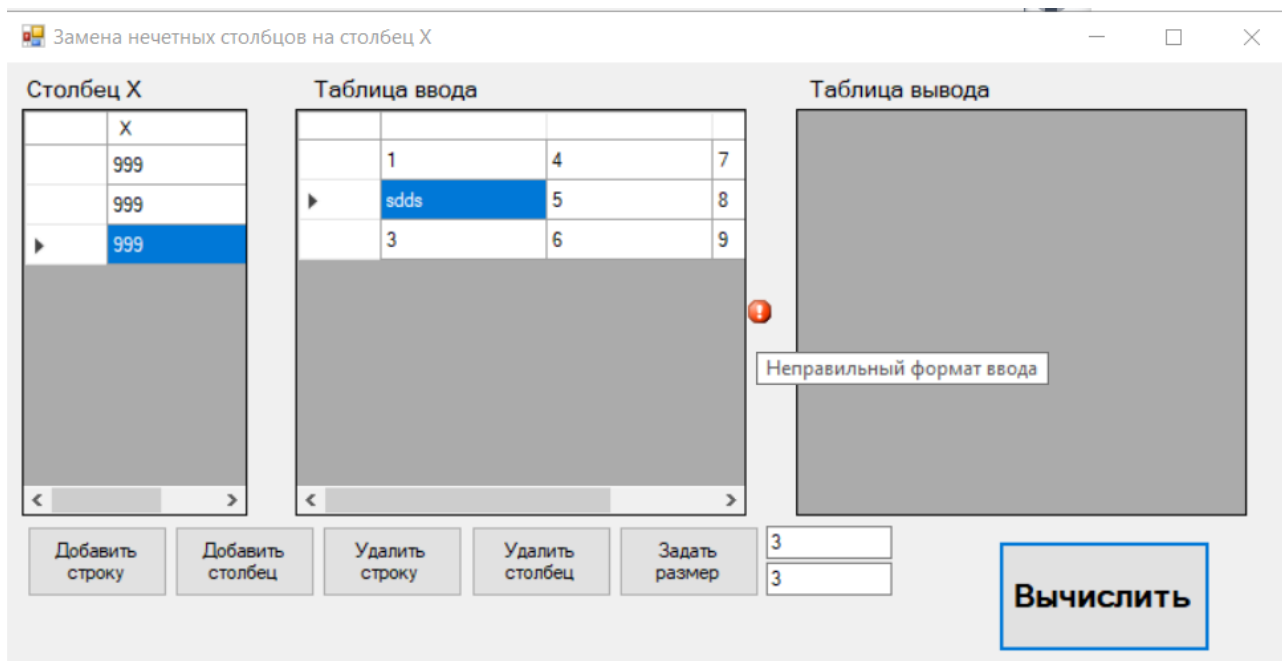


Рисунок 24 – Ошибка формата ввода

## 5.5 Примеры исходного кода

Функция при нажатии на кнопку "Удалить столбец".

```

1 private: System::Void remove_column_Click(System::Object^ sender, System::EventArgs^
  ↪ e) {
2     this->errorProvider1->Clear();
3     if (this->input_grid->ColumnCount <= 1 || this->input_grid->RowCount == 0) {
4         this->errorProvider1->SetError(remove_column, "Нельзя убрать столбец, которого
  ↪ нет");
5         return;
6     }
7     int i = this->input_grid->CurrentCell->ColumnIndex;
8     this->input_grid->Columns->Remove(this->input_grid->Columns[i]);
9 }

```

Другие фрагменты кода расположены в приложении Д.

## 6 Матричный калькулятор

### 6.1 Условие задания

Создать приложение, реализующее основные операции с векторами и матрицами:

1. Ввод матрицы, вектора
2. Создание матриц (единичная, матрица как набор векторов)
3. Умножение на число, вектор, матрицу
4. Сложение/вычитание двух матриц
5. Сложение/вычитание двух векторов
6. Скалярное и векторное произведение двух векторов
7. Транспонированная матрица
8. Определитель, ранг матрицы

Выводить сообщения об ошибках (ввод не числа, несоответствие размерностей)

### 6.2 Вид формы в конструкторе

Создано окно приложения, содержащее пять элементов TextBox, четыре элемента Label, четыре элемента Button, один элемент gridView и один элемент ErrorProvider для обработки ошибок. Вид окна представлен на рисунке 25 [6].

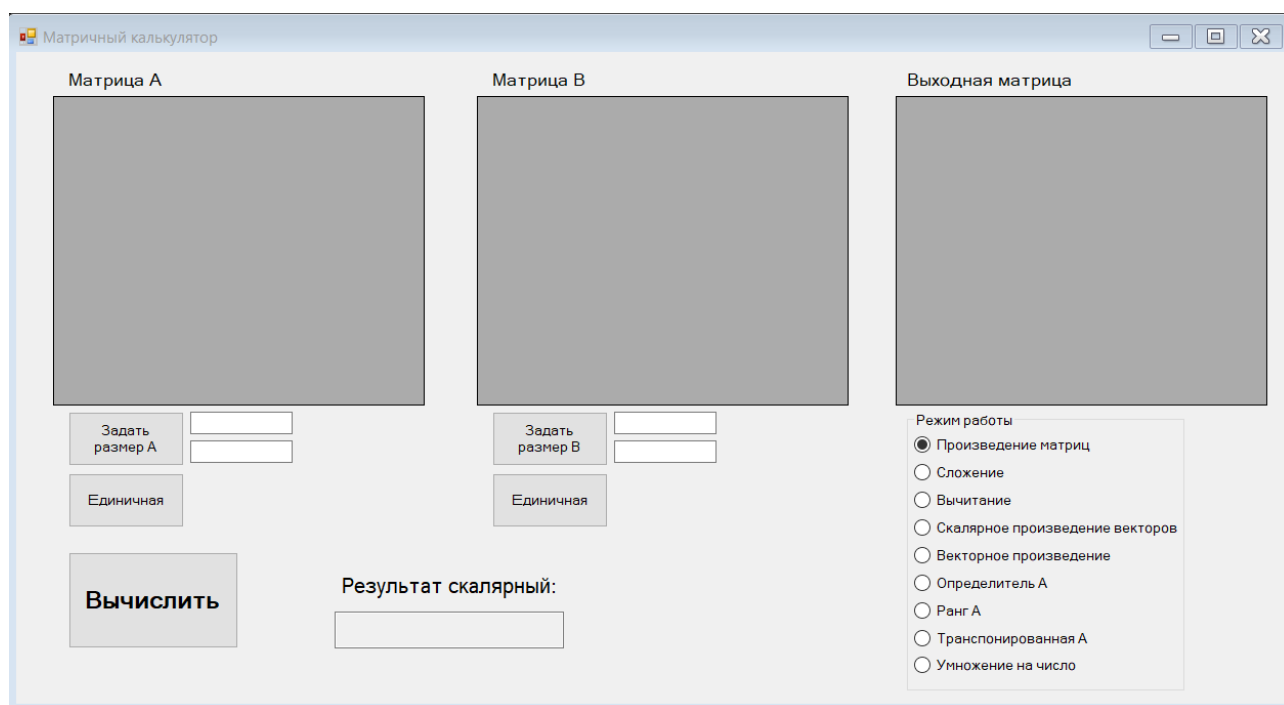


Рисунок 25 – Окно приложения «Матричный калькулятор» открытое в конструкторе



### 6.3 Таблица с описанием переименованных элементов формы

Все измененные элементы формы указаны в таблице 6.

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

Описание элементов формы	Список измененных атрибутов	Новое значение атрибута
Форма MyForm	Text	Матричный калькулятор
TextBox ввода количества строк Матрицы А	Name	set_size_A_lines
TextBox ввода количества столбцов таблицы А	Name	set_size_A_columns
TextBox ввода количества строк Матрицы В	Name	set_size_B_lines
TextBox ввода количества столбцов матрицы В	Name	set_size_B_columns
Кнопка "Единичная" под матрицей А	Name	edbtn
	Text	Единичная
Кнопка "Единичная" под матрицей В	Name	edbtnB
	Text	Единичная
Кнопка "Вычислить"	Name	solve
	Text	Вычислить
TextBox вывода скалярного ответа	Name	output_scal
Кнопка "Задать размер" А	Name	set_size_A
	Text	Задать размер А
Кнопка "Задать размер" В	Name	set_size_B
	Text	Задать размер В
Таблица Матрица А	Name	input_A
	AllowUserToAddRows	False
	AllowUserToDeleteRows	False
Таблица Матрица В	Name	input_B
	AllowUserToAddRows	False
	AllowUserToDeleteRows	False
	Name	output_grid

	AllowUserToAddRows	False
	AllowUserToDeleteRows	False
	ReadOnly	True
GroupBox выбора режима работы	Name	modes
	Text	Режим работы
RadioButton Произведение матриц	Name	mode1_proiz
	Text	Произведение матриц
	Checked	True
RadioButton Сложение	Name	mode2_sloz
	Text	Сложение
RadioButton Вычитание	Name	mode3_sub
	Text	Вычитание
RadioButton Скалярное произведение	Name	mode4_skalproz
	Text	Скалярное произведение векторов
RadioButton векторное произведение	Name	mode5_vectproz
	Text	Векторное произведение
RadioButton Определитель A	Name	mode6_oprA
	Text	Определитель A
RadioButton Ранг A	Name	mode7_rankA
	Text	Ранг A
RadioButton Транспонированная A	Name	mode8_tranA
	Text	Транспонированная A
RadioButton Умножение на число	Name	mode9_numproz
	Text	Умножение на число

#### 6.4 Примеры правильной и неправильной работы

После запуска программы на экране появляется окно на рисунке 26.

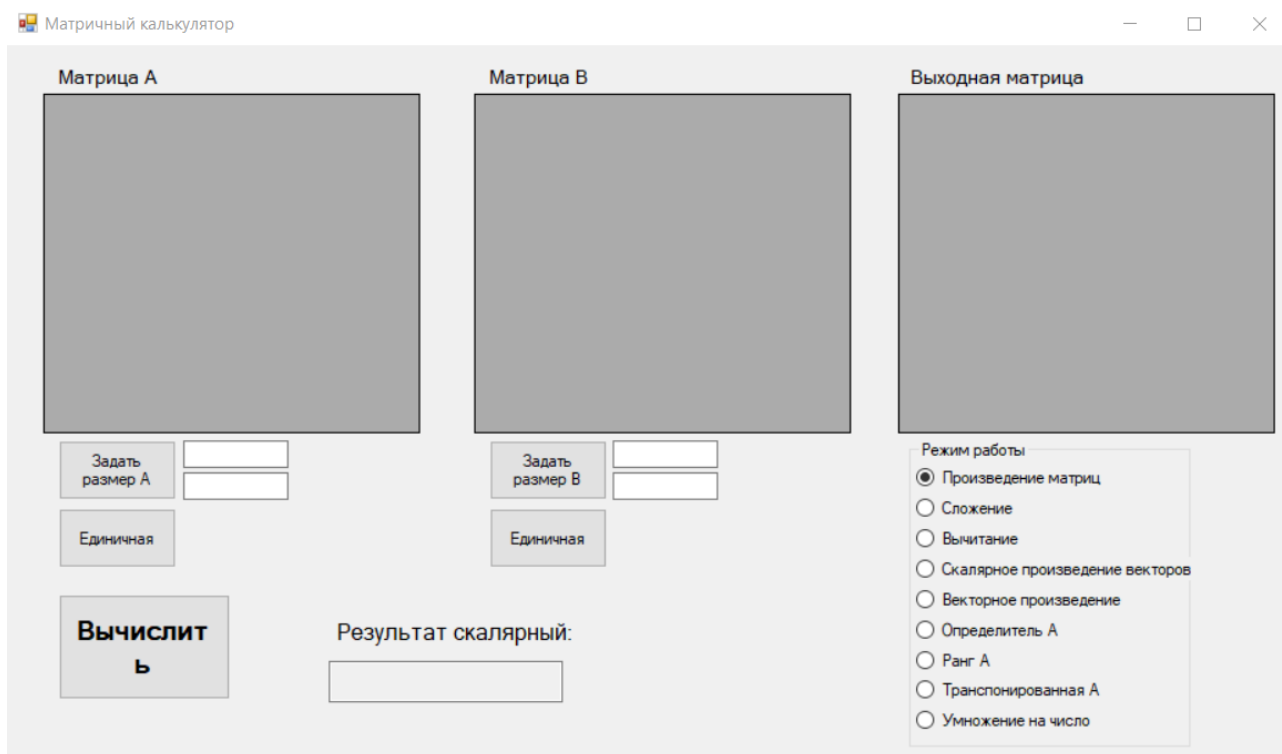


Рисунок 26 – Запуск программы

После ввода двух матриц и нажатии на кнопку "Вычислить" (на рисунке 27).

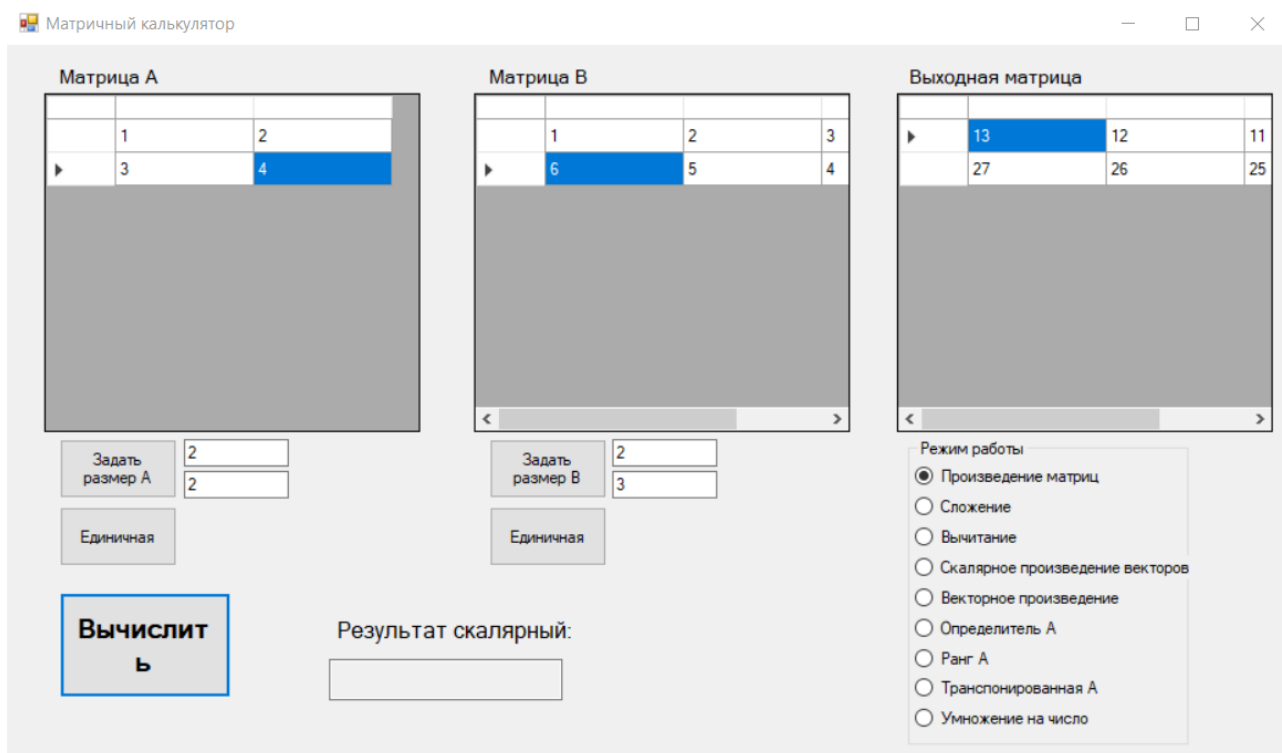


Рисунок 27 – Произведение матриц

При попытке ввода не числа в таблицу, программа выведет ошибку (на

рисунке 28)

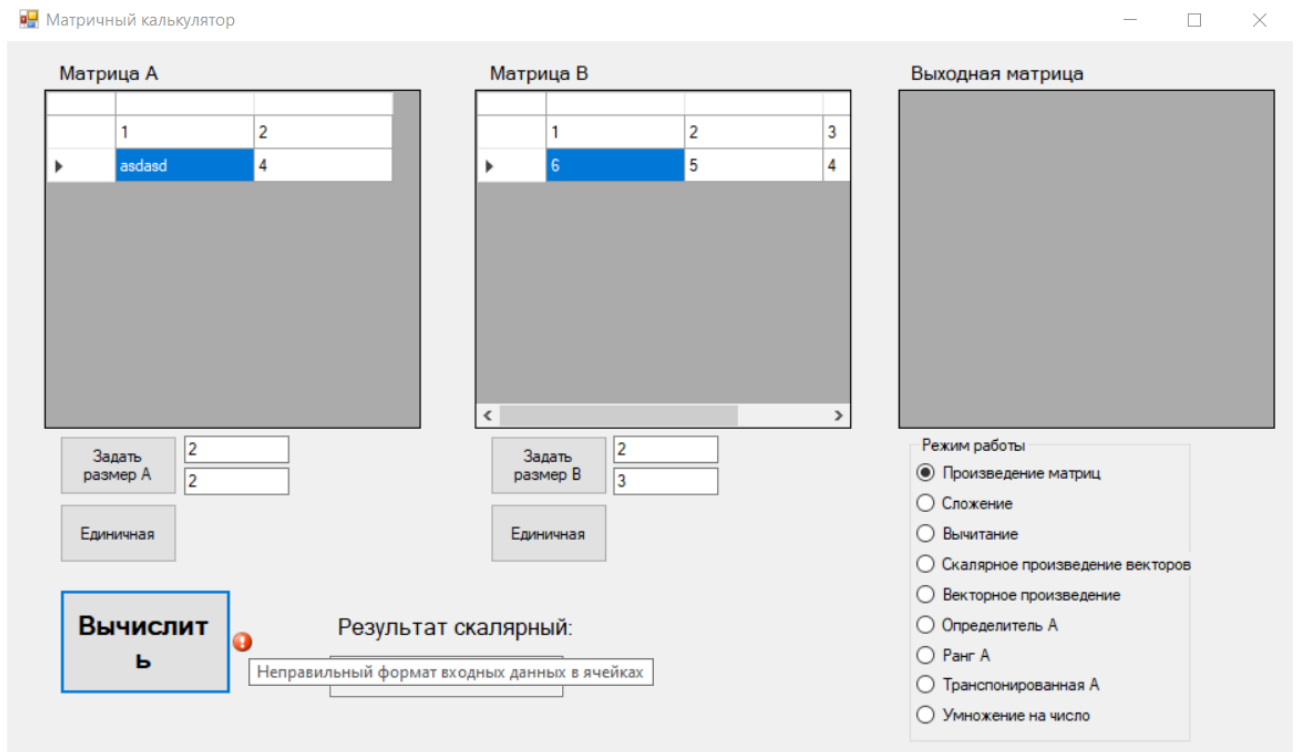


Рисунок 28 – Ошибка формата ввода

## 6.5 Примеры исходного кода

Функция создания единичной матрицы B.

```

1 private: System::Void edbtnB_Click(System::Object^ sender, System::EventArgs^ e) {
2     int line_count = this->input_B->RowCount;
3     int column_count = this->input_B->ColumnCount;
4     try {
5         if (line_count != column_count) throw gcnew Exception("Матрица не квадратная");
6         for (int i = 0; i < line_count; i++) {
7             for (int j = 0; j < line_count; j++) {
8                 if (i == j) this->input_B->Rows[i]->Cells[j]->Value = 1;
9                 else this->input_B->Rows[i]->Cells[j]->Value = 0;
10            }
11        }
12    }
13    catch (Exception^ e) {
14        this->errorProvider1->SetError(this->solve, e->Message);
15    }
16 }

```

Другие фрагменты кода расположены в приложении Е.

## **7 Использование коллекций**

### **7.1 Условие задания**

Разработать приложение в соответствии со своим вариантом. Номер варианта выдает преподаватель.

Приложение должно содержать следующие пункты:

1. Заголовок формы должен отражать суть задания.
2. Все элементы формы должны быть внятно подписаны (кнопки подписаны, у тестового поля должно быть написано, для чего оно нужно и т. д.)
3. В коде должны быть комментарии и отступы (код должен быть легко читаем).
4. В коде программы все элементы формы должны быть переименованы (btnName - для кнопок, lblName - для ссылок, txtName - для текстового поля и т. д.) Наименования должны быть понятными.
5. Должна быть возможность для ввода и вывода первоначальных данных.
6. Должна быть возможность для вставки и удаления одного элемента.
7. Должны использоваться коллекции.
8. Ответы на задания должны быть в разных полях.
9. Если нет данных для выполнения задания, выводить соответствующие данные.
10. Для графов использовать список смежности.

**Вариант 7.** Создать стек, состоящий из целых чисел. Предусмотреть возможность создания стека из набора чисел, добавления одного элемента (функция push), удаления одного элемента (функция pop), вывода результата на экран, удаления всех элементов с помощью кнопок. Найти максимальный нечетный элемент. Получить новый стек, удалив все нечетные элементы.

### **7.2 Вид формы в конструкторе**

Для реализации стека создано окно приложения, содержащее пять элементов TextBox, три элемента Label, пять элементов Button и один элемент ErrorProvider для обработки ошибок. Вид окна представлен на рисунке 29 [7].

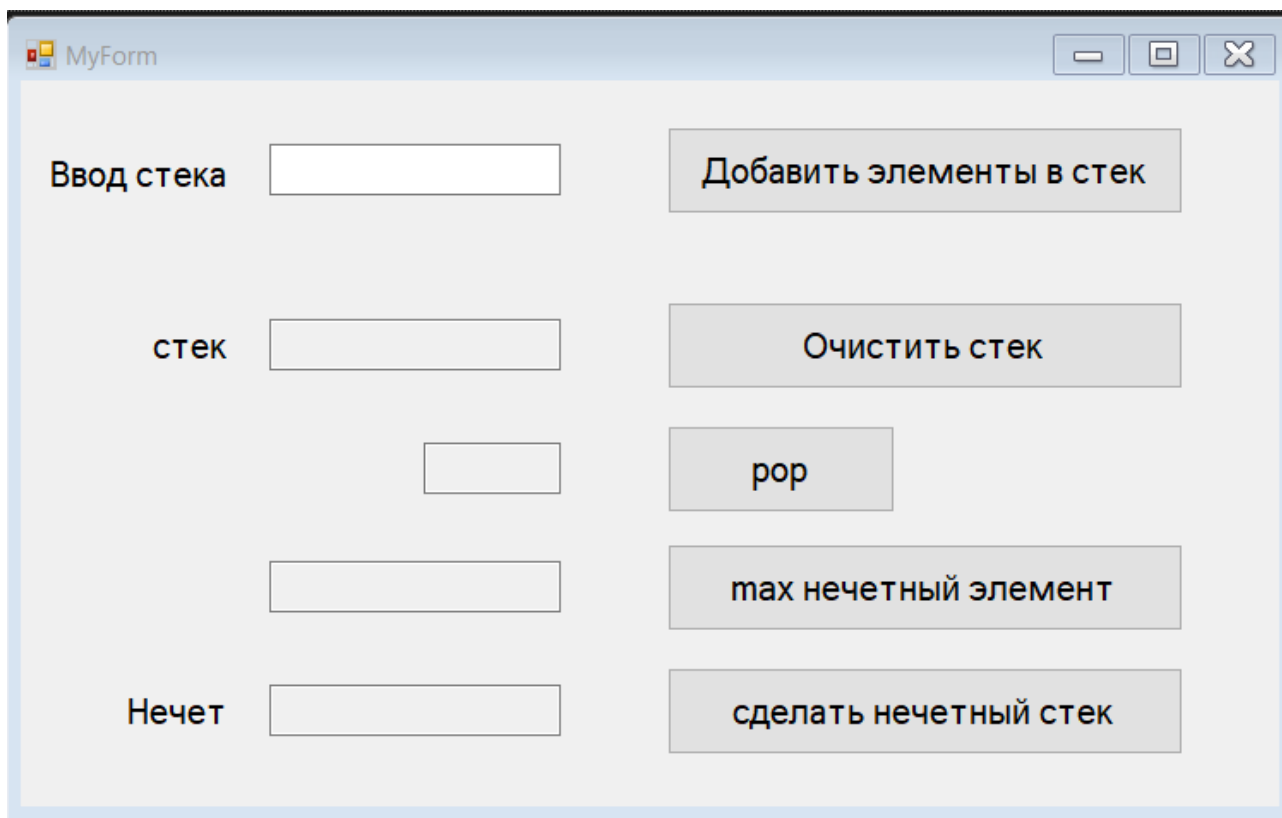


Рисунок 29 – Окно приложения «Матричный калькулятор» открытое в конструкторе

### 7.3 Таблица с описанием переименованных элементов формы

Все измененные элементы формы указаны в таблице 7.

Таблица 7 – Значения атрибутов элементов в приложении «Работа с коллекциями»

Описание элементов формы	Список измененных атрибутов	Новое значение атрибута
Форма MyForm	Text	Работа с коллекциями
TextBox ввода стека	Name	input_stack
TextBox вывода стека	Name	output
	ReadOnly	True
TextBox вывода pop элемента	Name	pop_out
	ReadOnly	True
TextBox вывода максимального нечетного элемента	Name	max_nech
	ReadOnly	True
TextBox вывода стека из нечетных элементов	Name	nech_stack
	ReadOnly	True
Кнопка "Добавить элементы в стек"	Name	create_stack

	Text	Добавить элементы в стек
Кнопка "Очистить стек"	Name	clear_stack
	Text	Очистить стек
Кнопка "pop"	Name	pop_stack
	Text	pop
Кнопка "max нечетный элемент"	Name	max_nech_btn
	Text	max нечетный элемент
Кнопка "Сделать нечетный стек"	Name	nech_stack_btn
	Text	сделать нечетный стек

## 7.4 Примеры правильной и неправильной работы

После запуска программы на экране появляется окно на рисунке 30.

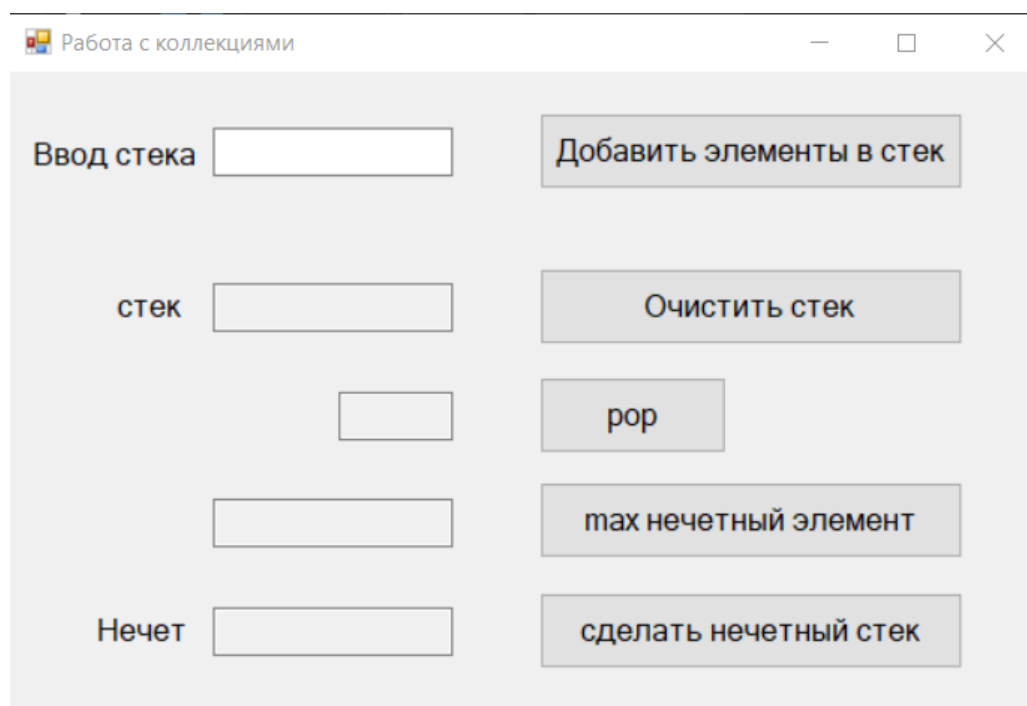


Рисунок 30 – Запуск программы

После ввода поле целых чисел через пробел и нажатии на кнопку "Добавить элементы в стек"(на рисунке 31).

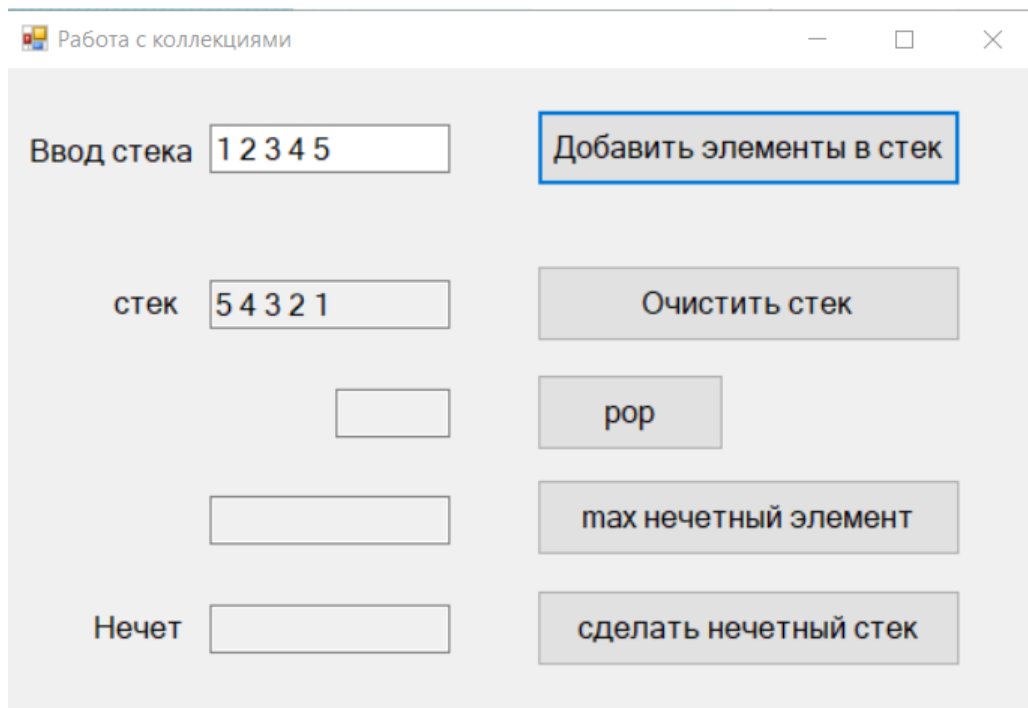


Рисунок 31 – Ввод стека

При попытке ввода не числа в поле, программа выведет ошибку (на рисунке 32)

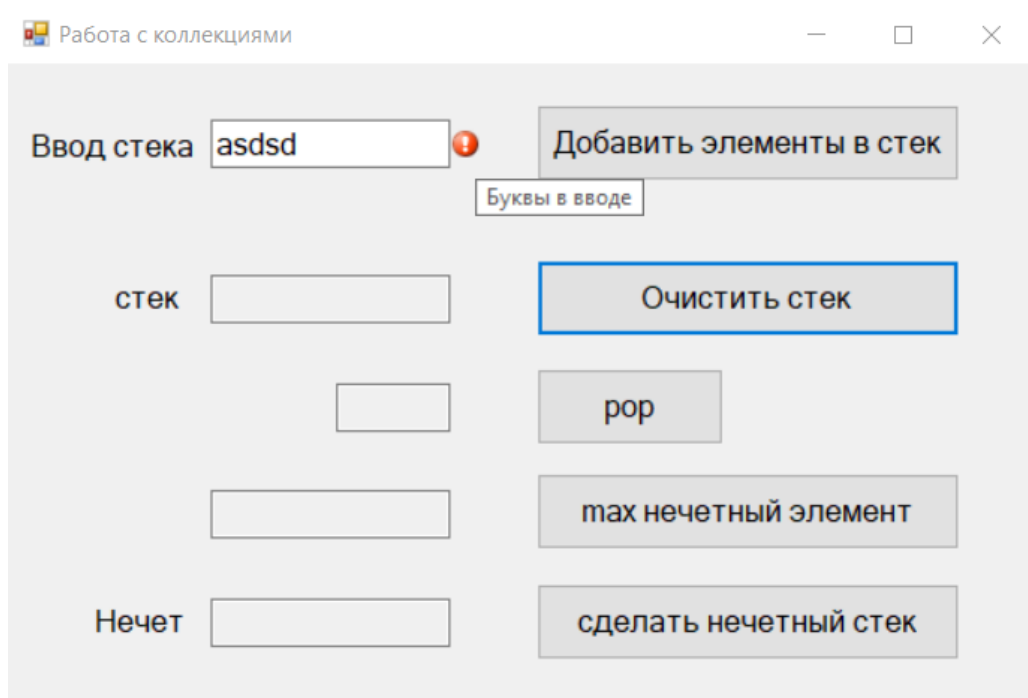


Рисунок 32 – Ошибка формата ввода

## 7.5 Примеры исходного кода

Функция очистки стека.



```

1 private: System::Void stack_out() {
2     System::Collections::Generic::Stack<int> buf; //вспомогательный стек
3     System::String^ str2 = "";
4     while (s.Count) { //пока стек не пуст
5         buf.Push(s.Peek()); //записываем во вспомогательный стек первый элемент
6         str2 += System::Convert::ToString(s.Peek()) + " "; //записываем первый элемент в
           ↪ строку
7         s.Pop(); //удаляем первый элемент из стека
8
9
10    }
11    while (buf.Count) { //пока вспомогательный стек не пуст
12        s.Push(buf.Peek()); //записываем в основной стек первый элемент вспомогательного
13        buf.Pop(); //удаляем из стека первый элемент
14
15    }
16
17    this->output->Text = str2; //записываем результат в строку
18 }

```

Другие фрагменты кода расположены в приложении Ж.

## 8 Работа с файлами

### 8.1 Условие задания

Разработать приложение для своего варианта.

Приложение должно содержать следующие компоненты:

1. Заголовок формы должен отражать суть задания.
2. Все элементы формы должны быть внятно подписаны (кнопки подписаны, у тестового поля должно быть написано, для чего оно нужно и т. д.)
3. Структура представляет собой таблицу DataGridView, в ячейках которой записаны данные разных типов.
4. Предусмотреть кнопки:
  - считывание данных из файла и запись данных в таблицу (предполагается, что в файле данные корректные);
  - возможность добавлять и удалять строки в таблице, соответственно, вводить данные вручную.
  - запись в файл всех данных. Проверять корректность ввода данных: даты должны быть реальные, номер телефона состоять из цифр и, может быть, знак тире, оценки студентов - от 2 до 5. Все остальное можно не проверять. Если есть срок пребывания, дата прибытия и дата отбытия, то срок пребывания должен быть равен разнице между датой отбытия и датой прибытия.
  - запись в файл данных по определенному критерию. Критерий можно вводить вручную через TextBox или выбирать с помощью Radiobox и т. д. Запись в файл должна быть такой, чтобы этот файл можно было открыть в приложении.
  - запись данных по определенному критерию в новую таблицу. При выборе другого критерия старая таблица должна удаляться.
5. При неправильном вводе каких-либо данных таблица выбранных данных должна очищаться.
6. В коде должны быть комментарии и отступы (код должен быть легко читаем).

**Вариант 11.** Создать таблицу Travel, содержащую следующие поля: Фамилия, имя, отчество туриста, название гостиницы, срок пребывания, дата приезда, дата отъезда. Выполнить следующие действия: считать из файла и вывести на экран данные о всех туристах, в другой файл вывести данные о туристах, при-

езжающих в данный день.

## 8.2 Вид формы в конструкторе

Создано окно приложения, содержащее один элемент TextBox, один элемент Label, шесть элементов Button, два элемента gridView, один элемент ErrorProvider для обработки ошибок и saveFileDialog и openFileDialog для работы с файлами. Вид окна представлен на рисунке 33 [8][9].

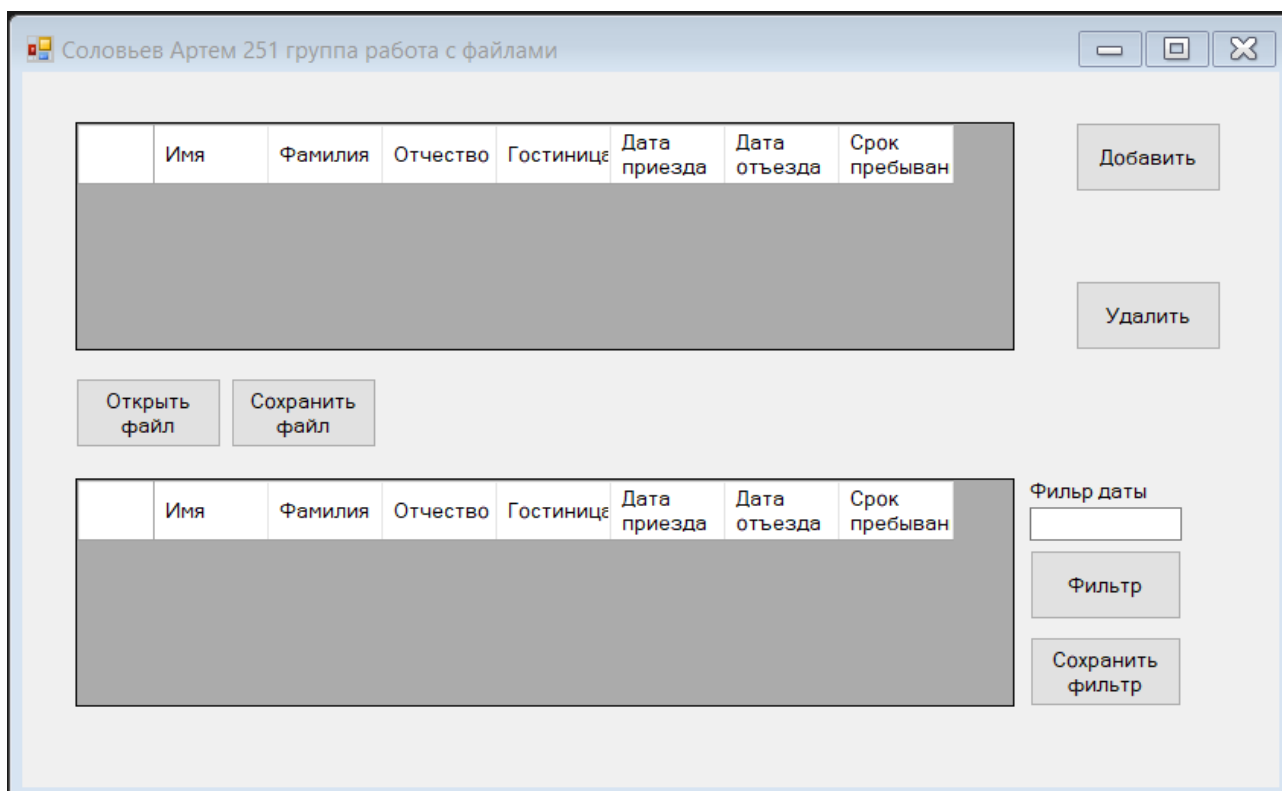


Рисунок 33 – Окно приложения «Матричный калькулятор» открытое в конструкторе

## 8.3 Таблица с описанием переименованных элементов формы

Все измененные элементы формы указаны в таблице 8.

Таблица 8 – Значения атрибутов элементов в приложении «Работа с файлами»

Описание элементов формы	Список измененных атрибутов	Новое значение атрибута
Форма MyForm	Text	Соловьев Артем 251 группа работа с файлами
TextBox фильтра даты	Name	FilterDate

Кнопка "Добавить"	Name	AddRow
	Text	Добавить
Кнопка "Удалить"	Name	DeleteRow
	Text	Удалить
Кнопка открытия файла	Name	FileOpen
	Text	Открыть файл
Кнопка сохранения файла	Name	FileSave
	Text	Сохранить файл
Кнопка фильтр даты	Name	FilterBtn
	Text	Фильтр
Кнопка сохранить отфильтрованную таблицу	Name	FileSaveFilter
	Text	Сохранить фильтр
Таблица постояльцев	Name	MainTable
	AllowUserToAddRows	False
Таблица постояльцев (отфильтрованная)	Name	FilteredTable
	AllowUserToAddRows	False
	AllowUserToDeleteRows	False
	ReadOnly	True

#### 8.4 Примеры правильной и неправильной работы

После запуска программы на экране появляется окно на рисунке 34.

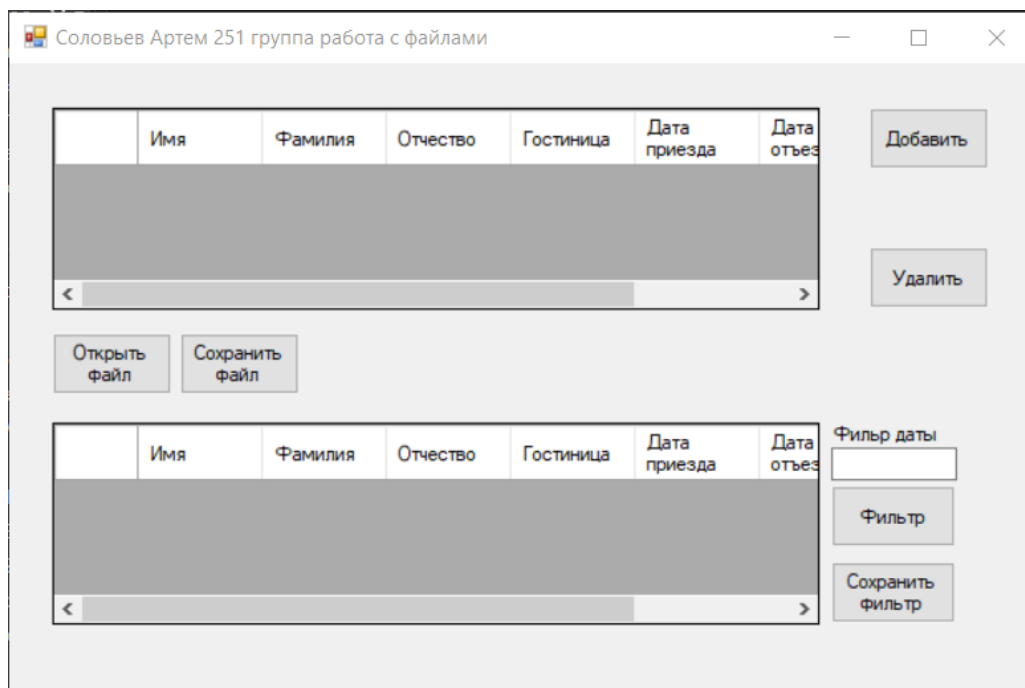


Рисунок 34 – Запуск программы

После ввода данных постояльцев, даты в фильтр даты и нажатии на кнопку "Фильтр"(на рисунке 35).

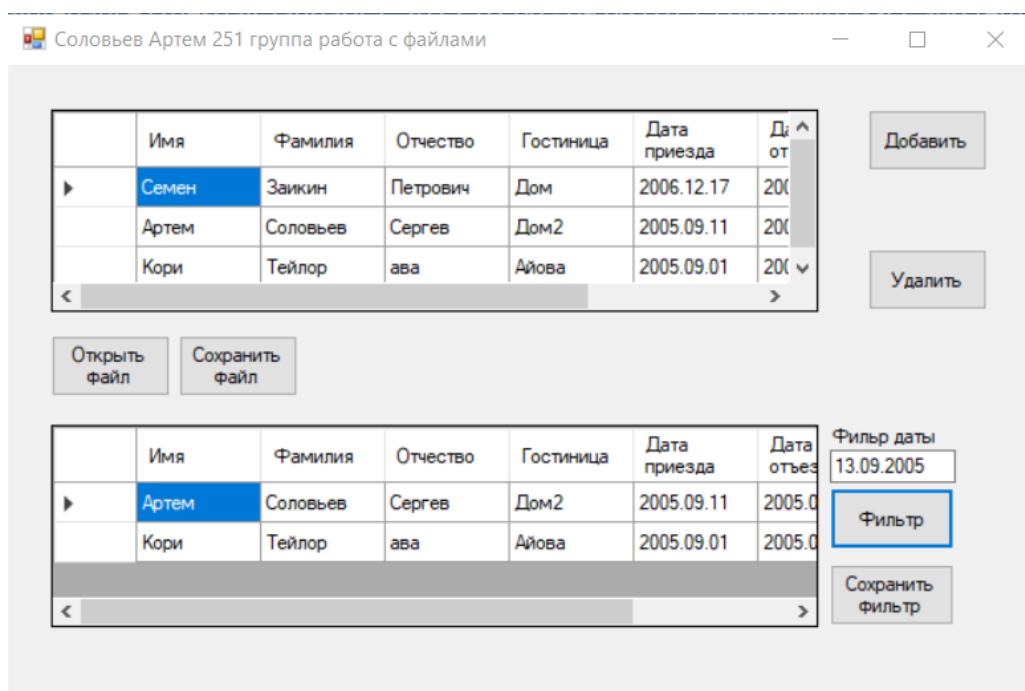


Рисунок 35 – Отфильтрованная таблица

При попытке ввода не даты в поле, программа выведет ошибку (на рисунке 36)

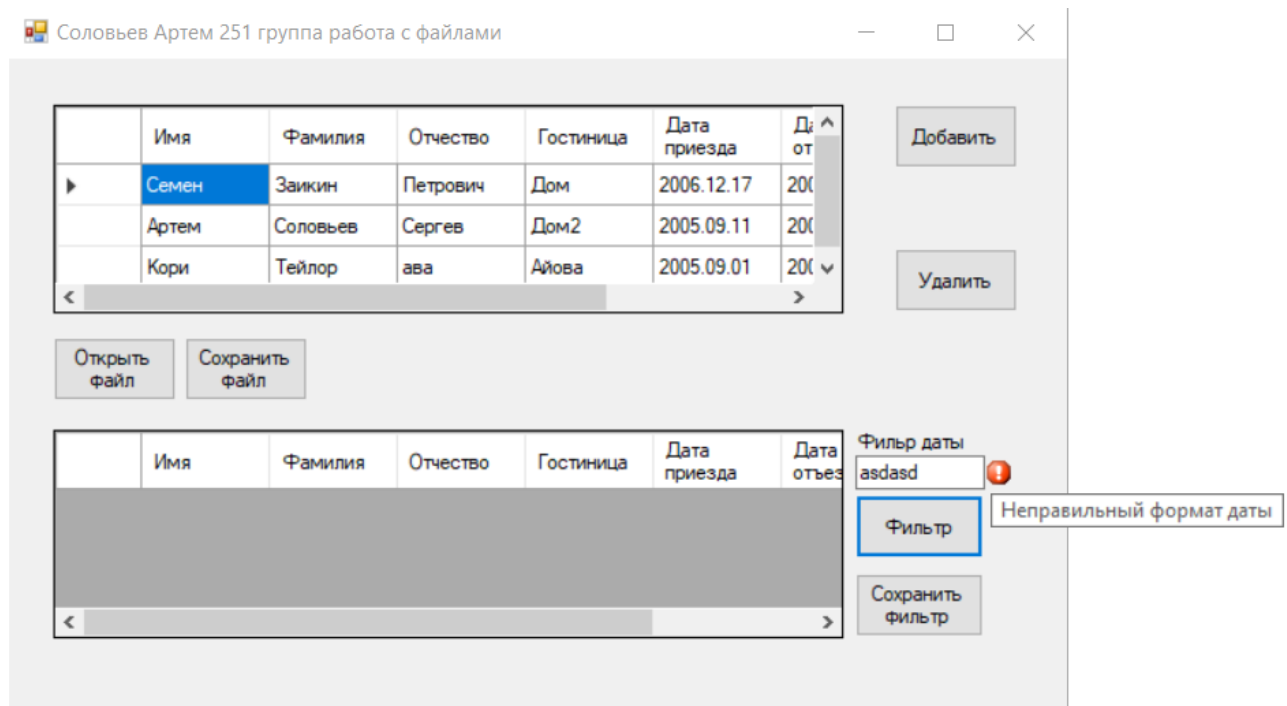


Рисунок 36 – Ошибка формата ввода

## 8.5 Примеры исходного кода

Функция удаления строки таблицы.

```

1 private: System::Void DeleteRow_Click(System::Object^ sender, System::EventArgs^ e) {
2     this->errorProvider1->Clear();
3     if (this->MainTable->RowCount == 0) {
4         this->errorProvider1->SetError(DeleteRow, "Нельзя убрать строку, которой нет");
5         return;
6     }
7     int i = this->MainTable->CurrentRow->Index;
8     this->MainTable->Rows->Remove(this->MainTable->Rows[i]);
9 }

```

Другие фрагменты кода расположены в приложении 3.

## **9 Приложение ТЕСТ**

### **9.1 Условие задания**

Создать приложение для проведения тестирования.

Должно содержать:

1. Набор вопросов по какой-то теме (и вопросы и ответы должны быть реальными) - не менее 10
2. Вопросы должны выбираться случайным образом.
3. Вопросы должны быть нескольких типов - "Да/нет Выбор одного ответа, Выбор нескольких ответов, Короткий ответ.
4. Необходимо создать сообщения для правильного и неправильного ответа (Молодец, Не правильно и т.д.)
5. Необходимо подсчитать количество правильных ответов и вывести результат.

### **9.2 Вид формы в конструкторе**

Создано две формы приложения, содержащие три элемента TextBox, три элемента Label, один элемент Button, один элемент ErrorProvider для обработки ошибок и один элемент groupBox с четырьмя элементами checkbox. Вид окна вопроса представлен на рисунке 37 [10].

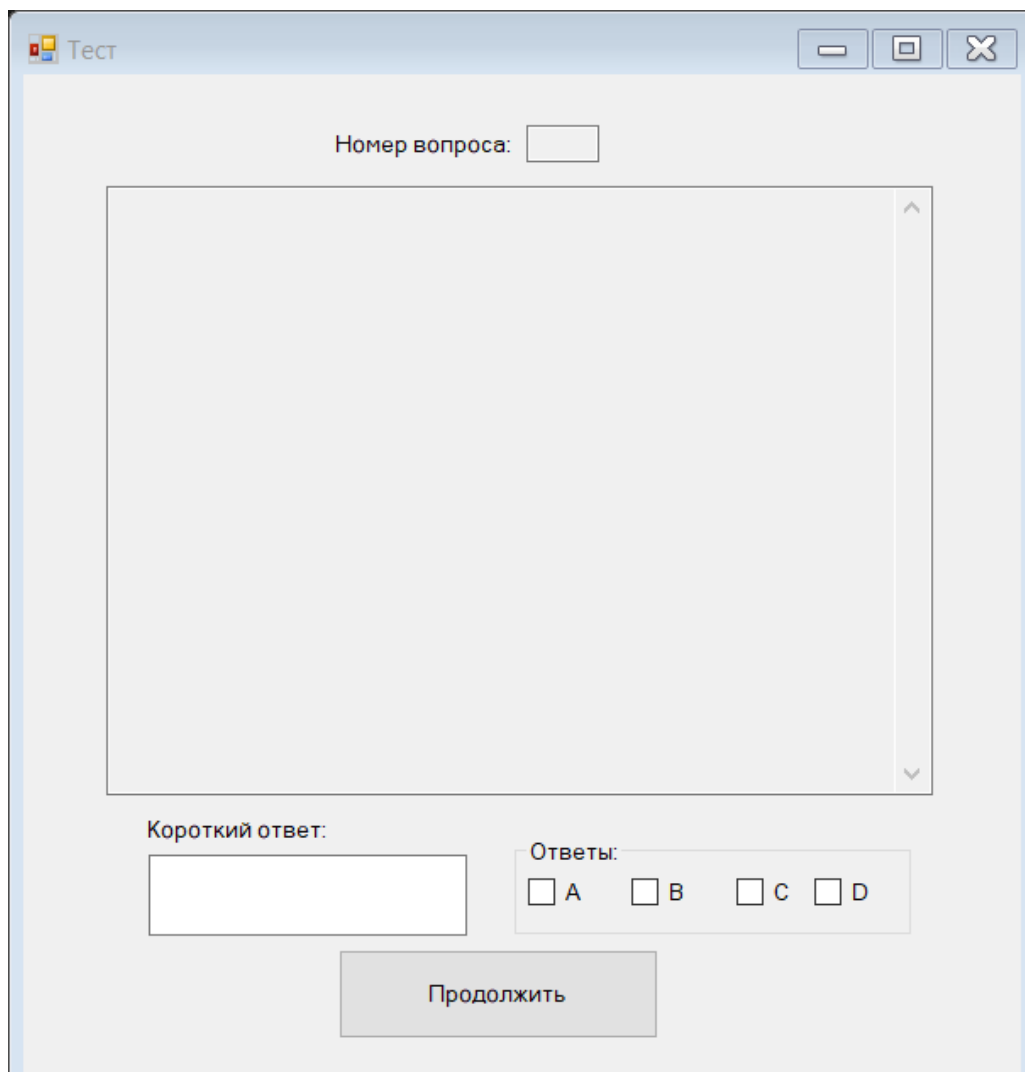


Рисунок 37 – Окно вопросов приложения «Тест» открытое в конструкторе

Вид окна результата представлен на рисунке 38.

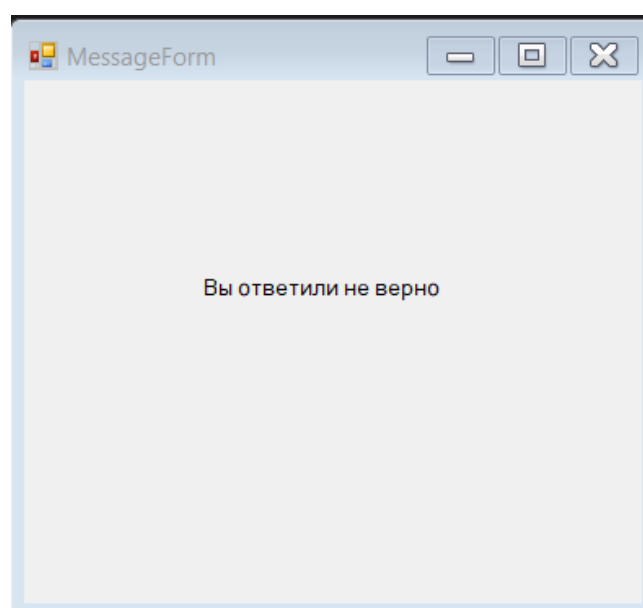


Рисунок 38 – Окно результата приложения «Тест» открытое в конструкторе



### 9.3 Таблица с описанием переименованных элементов формы

Все измененные элементы формы указаны в таблице 9.

Таблица 9 – Значения атрибутов элементов в приложении «Тест»

Описание элементов формы	Список измененных атрибутов	Новое значение атрибута
Форма MyForm	Text	Тест
TextBox номера вопроса	Name	countBox
	ReadOnly	True
TextBox короткого ответа	Name	shortAns
	Multiline	True
TextBox текста вопроса	Name	questBox
	Multiline	True
	ReadOnly	True
Кнопка ответа	Name	act
	Text	Продолжить
GroupBox вариантов ответа	Name	answerGroup
	Text	Ответы:
CheckBox варианта ответа A	Name	answerA
	Text	A
CheckBox варианта ответа B	Name	answerB
	Text	B
CheckBox варианта ответа C	Name	answerC
	Text	C
CheckBox варианта ответа D	Name	answerD
	Text	D

### 9.4 Примеры правильной и неправильной работы

После запуска программы на экране появляется окно на рисунке 39.

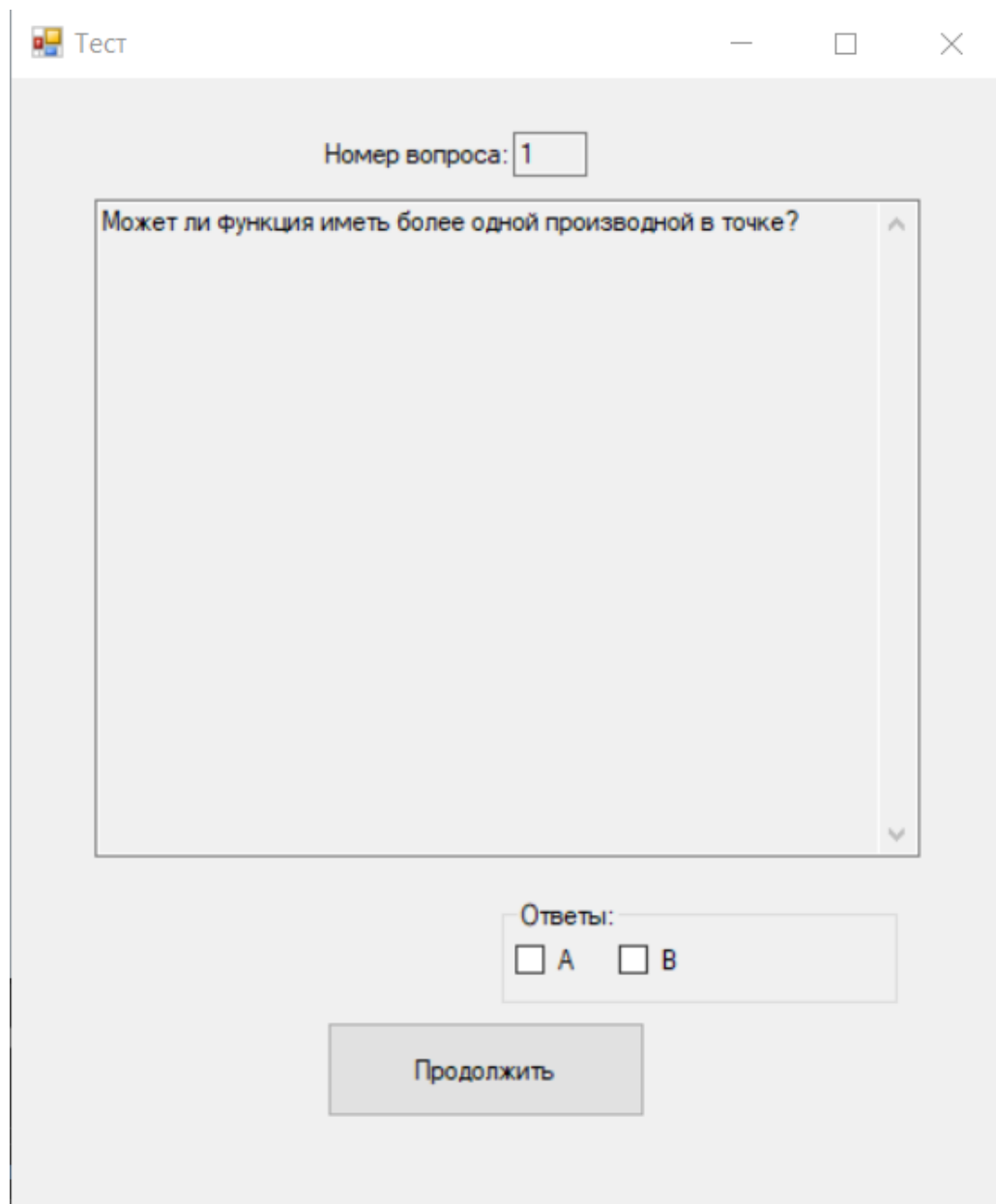


Рисунок 39 – Запуск программы

После выбора варианта ответа и нажатия кнопки "Продолжить"( на рисунке 40).

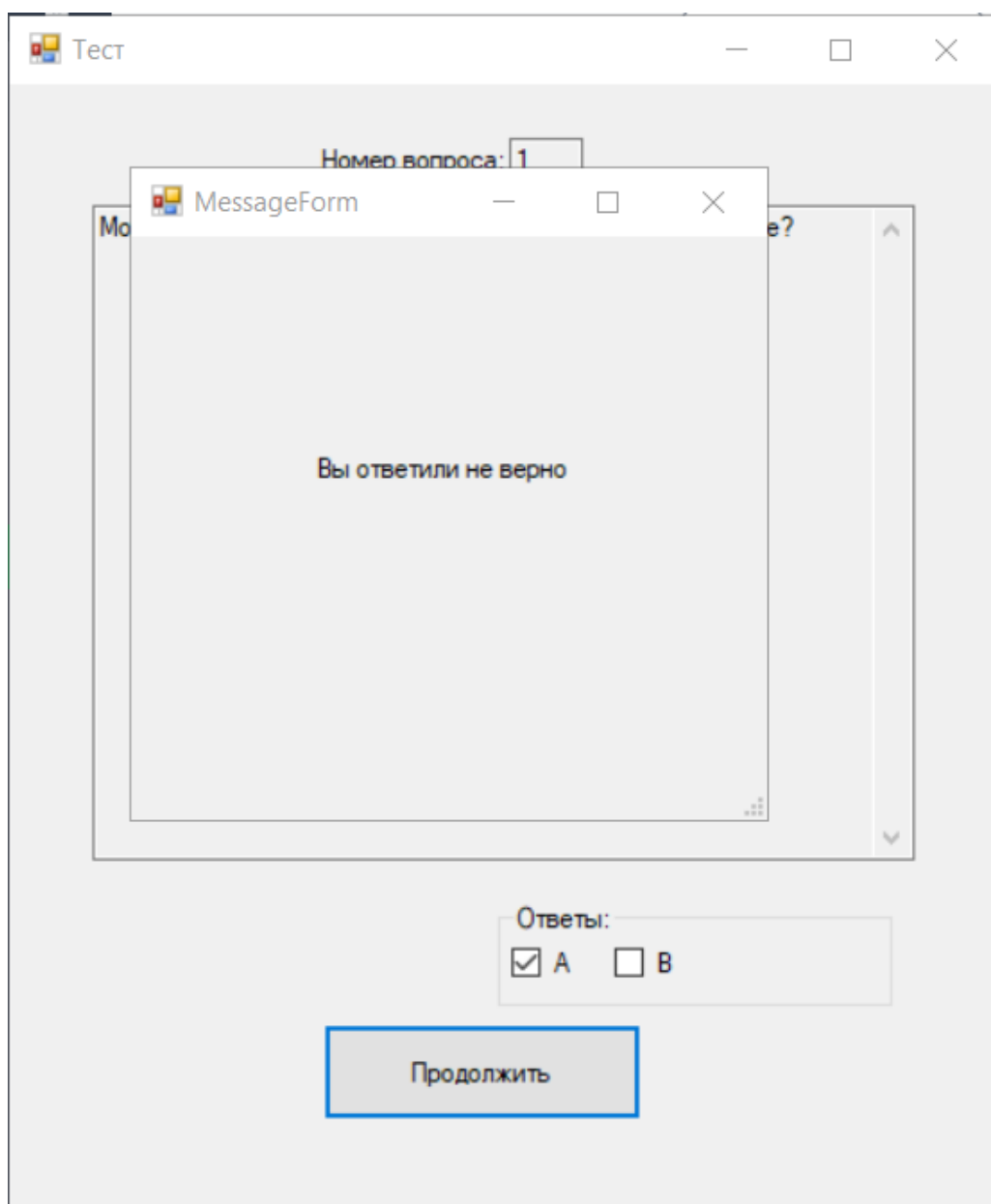


Рисунок 40 – Отфильтрованная таблица

При выбрать одновременно "Да" и "Нет" вылезит ошибка (на рисунке 41)

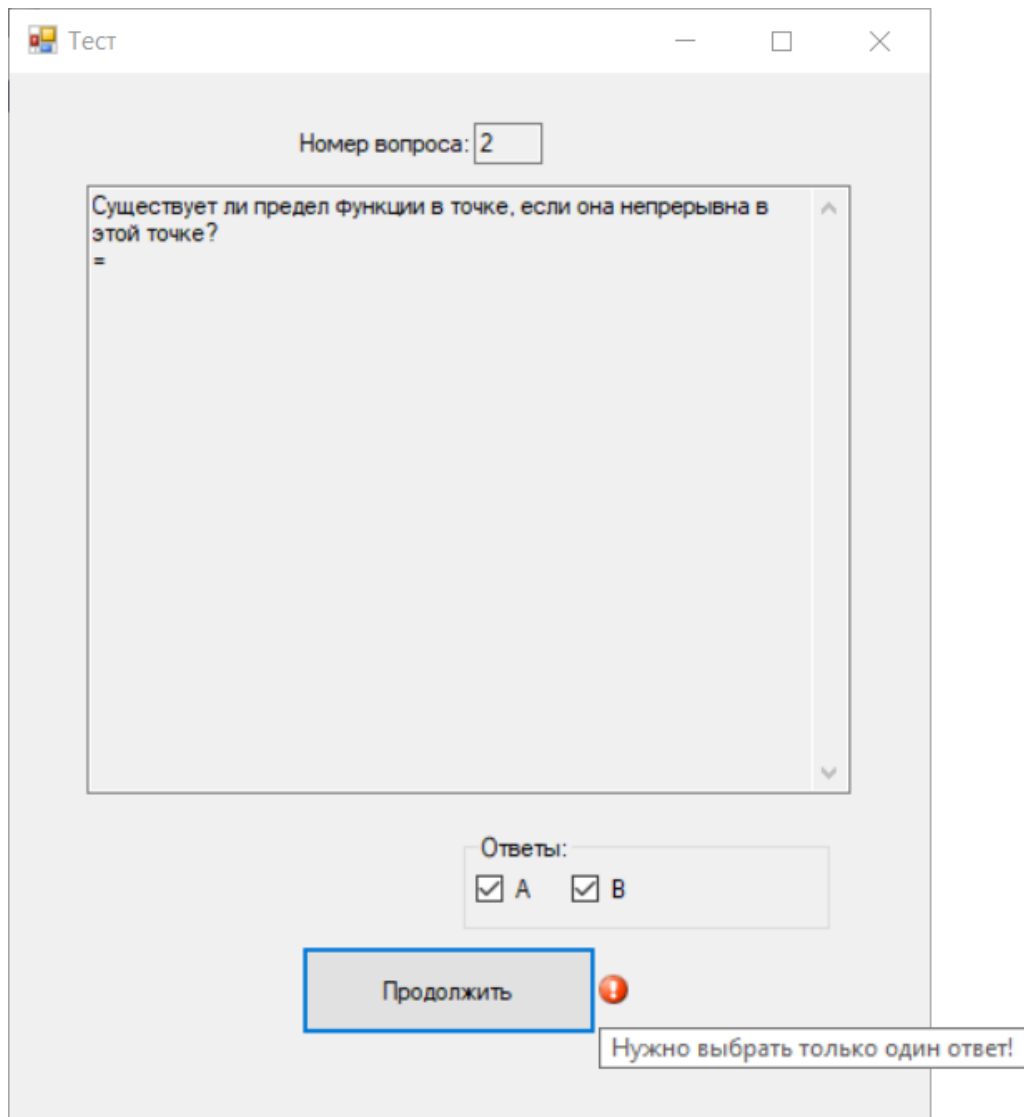


Рисунок 41 – Ошибка формата ввода

## 9.5 Примеры исходного кода

Функция считывания вопроса из вектора.

```
1 std::string ReadAnswer() {
2     int i = RandIndex->at(iter);
3     Quest temp = Questions->at(i);
4     if (temp.getType() == "ДаНет") {
5         return YesNoRead();
6     }
7     else if (temp.getType() == "ОдинОтвет") {
8         return OneAnswerRead();
9     }
10    else if (temp.getType() == "НесколькоОтветов") {
11        return SomeAnswersRead();
12    }
```

```
12     }  
13     else {  
14         return ShortAnswerRead();  
15     }  
16  
17 }
```

Другие фрагменты кода расположены в приложении И.

## **ЗАКЛЮЧЕНИЕ**

В ходе практики было реализовано несколько приложений в среде Microsoft Visual Studio с целью закрепления навыков построения оконного интерфейса и программирования с использованием C++/CLI.

На практике разработаны приложения, содержащие такие элементы интерфейса, как TextBox, Label, Button, DataGridView, OpenFileDialog, SaveFileDialog, ErrorProvider.

Было освоено динамическое создание элементов формы в процессе выполнения программы, изучены виды панелей и способы размещения элементов в них. Также были применены навыки программирования на основе обратных вызовов при обработке событий формы.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Лаврентьев, ДО. Использование технологии Windows forms для разработки информационных систем (на примере электронного журнала) [Текст] / ДО Лаврентьев, ВЮ Белаш // Вестник Калужского университета. — 2020. — № 4. — С. 82–85.
- 2 Никлаус, Вирт. Алгоритмы и структуры данных [Текст] / Вирт Никлаус. — Москва : БМК Пресс, 2022.
- 3 Комракова, Евгения Владимировна. Приложение Windows forms для формирования рейтинга и отзывов к японской мультипликации [Текст] / Евгения Владимировна Комракова, Анна Сергеевна Лещун. — 2022.
- 4 Ахо, Альфред В. Структуры данных и алгоритмы [Текст] / Альфред В Ахо. — Москва : Издательский дом Вильямс, 2000.
- 5 Лафоре, Роберт. Объектно-ориентированное программирование в C++ [Текст] / Роберт Лафоре. — Санкт-Петербург : Питер, 2011.
- 6 Stroustrup, Bjarne. The C++ Programming Language [Text]. "— 2013.
- 7 Chang, Gao-Wei. Analysis of Paint Messages with Multiple C++/CLI Controls [Text] / Gao-Wei Chang. "— 2021. "— P. 138–141.
- 8 Gladstone, Adam. Building a C++/CLI Wrapper [Text] / Adam Gladstone. "— 2022. "— P. 41–66.
- 9 Kaiser, Richard. C++/CLI,. NET-Bibliotheken und C++ Interoperabilität [Text] / Richard Kaiser. "— 2022. "— P. 937–974.
- 10 Зиборов, Виктор Владимирович. MS Visual C++ 2010 в среде. NET. Библиотека программиста [Текст] / Виктор Владимирович Зиборов. — Санкт-Петербург : Издательский дом "Питер 2011.

## ПРИЛОЖЕНИЕ А

### Фрагменты кода программы «Вычисление факториала»

Функция вычисления факториала:

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



## ПРИЛОЖЕНИЕ Б

### Фрагменты кода программы «Простые вычисления»

Функция, при нажатии на кнопку «Вычислить».

```
1 private: System::Void solvebtn_Click(System::Object^ sender, System::EventArgs^ e) {
2     this->output->Text = ""; // Стираем поле вывода
3
4     errors->SetError(x_input, String::Empty); // обнуляем ошибки
5     errors->SetError(y_input, String::Empty);
6
7     Int64 x, y; // Переменные для считывания полей ввода
8     double result; // переменная для записи результата
9
10    bool result_x = Int64::TryParse(this->x_input->Text, x); // записываем из полей ввода
    ↪ в соответствующие переменные
11    bool result_y = Int64::TryParse(this->y_input->Text, y); // и проверяем на успешность
    ↪ выполнения парсинга
12
13    if (!result_x) { // если неудачно
14        errors->SetError(x_input, "Введено не целое число");
15    }
16    if (!result_y) {
17        errors->SetError(y_input, "Введено не целое число");
18    }
19    if (result_x && result_y) { // если удачно
20        if (x * x - y == 0) { // проверка на ОДЗ
21            this->output->Text = "Деление на ноль";
22        }
23        else {
24            result = (1.0) * ((x + y) * (x + y) - x * x * x) / (std::abs(x * x - y)); //Считаем
25            this->output->Text = System::Convert::ToString(result); //Записываем результат
26        }
27    }
28 }
```

## ПРИЛОЖЕНИЕ В

### Фрагменты кода программы «Рекурсивные вычисления»

Функция обработки нажатия на кнопку «Вычислить».

```
1 private: System::Void Solve_Click(System::Object^ sender, System::EventArgs^ e) {
2     ClearAll();
3     long long n, m, A;
4     bool result_n = Int64::TryParse(this->n_input->Text, n);
5     bool result_m = Int64::TryParse(this->m_input->Text, m);
6     if (!result_n) {
7         errorProvider1->SetError(this->n_input, "Неправильно введено число n.");
8     }
9     if (!result_m) {
10        errorProvider1->SetError(this->m_input, "Неправильно введено число m.");
11    }
12    if (result_m && result_n) {
13
14        if (n > m) {
15            this->Output->Text = "Ошибка: n > m";
16        }
17        else {
18            if (n < 1) {
19                this->Output->Text = "" + "Ошибка: n < 1 " + System::Environment::NewLine;
20            }
21            if (m < 1) {
22                this->Output->Text += "Ошибка: m < 1\n";
23            }
24            if (n >= 1 && m >= 1) {
25                this->Output->Text = "Количество различных размещений из n в m равно: " +
26                    ↪ System::Convert::ToString(per(n, m));
27            }
28        }
29    }
```

## ПРИЛОЖЕНИЕ Г

### Фрагменты кода программы «Обработка табличных данных. Часть 1»

#### Функция обработки нажатия на кнопку "Минимум"

```
1 private: System::Void min_btn_Click(System::Object^ sender, System::EventArgs^ e) {
2     long long y, maxel = 1e9;
3     bool found_chet = false;
4     for (int i = 0; i < this->mas_grid->RowCount; i++) {
5         bool result = Int64::TryParse(this->mas_grid->Rows[i]->Cells[0]->Value->ToString(),
6             ↪ y);
7         if (y % 2 == 0) {
8             found_chet = true;
9             maxel = (y < maxel) ? y : maxel;
10        }
11    }
12    if (found_chet) this->chet_box->Text = System::Convert::ToString(maxel);
13    else {
14        this->chet_box->Text = "Четных чисел нет";
15    }
16 }
```

#### Функция добавления числа в таблицу

```
1 private: System::Void mas_add_Click(System::Object^ sender, System::EventArgs^ e) {
2     long long x;
3     bool result = Int64::TryParse(this->x_input->Text, x);
4     if (!result) {
5         this->errorProvider1->SetError(this->mas_add, "Ошибка формата ввода");
6         return;
7     }
8     this->mas_grid->Rows->Add(1);
9     this->mas_grid->Rows[this->mas_grid->RowCount - 1]->SetValues(x);
10
11 }
```

## ПРИЛОЖЕНИЕ Д

### Фрагменты кода программы «Обработка табличных данных. Часть 2»

#### Функция установки размера таблицы.

```
1 private: System::Void set_size_Click(System::Object^ sender, System::EventArgs^ e) {
2     this->errorProvider1->Clear();
3     int n, m;
4     bool resn = Int32::TryParse(this->set_size_lines->Text, n);
5     bool resm = Int32::TryParse(this->set_size_columns->Text, m);
6     if (!(resn && resm)) {
7         this->errorProvider1->SetError(this->input_grid, "Неправильный формат ввода
            ↪ размера таблицы");
8         return;
9     }
10    else if (n == 0 || m == 0) {
11        this->errorProvider1->SetError(this->input_grid, "Нельзя задать размер с 0");
12        return;
13    }
14
15
16    int line_count = this->input_grid->RowCount;
17
18    for (int i = 0; i < line_count; i++) {
19        this->input_grid->Rows->Remove(this->input_grid->Rows[0]);
20        this->X_input->Rows->Remove(this->X_input->Rows[0]);
21    }
22    int column_count = this->input_grid->ColumnCount;
23    for (int i = 0; i < column_count; i++) {
24        this->input_grid->Columns->Remove(this->input_grid->Columns[0]);
25    }
26
27
28    for (int i = 0; i < m; i++) {
29        this->input_grid->Columns->Add("", "");
30    }
31
32    this->input_grid->Rows->Add(n);
33    this->X_input->Rows->Add(n);
34
35
36 }
```

## Функция очистки вывода

```
1 System::Void clearoutput() {  
2     int line_count = this->output_grid->RowCount;  
3  
4     for (int i = 0; i < line_count; i++) {  
5         this->output_grid->Rows->Remove(this->output_grid->Rows[0]);  
6  
7     }  
8     int column_count = this->output_grid->ColumnCount;  
9     for (int i = 0; i < column_count; i++) {  
10         this->output_grid->Columns->Remove(this->output_grid->Columns[0]);  
11     }  
12 }
```

## ПРИЛОЖЕНИЕ Е

### Фрагменты кода программы «Матричный калькулятор»

#### Функция установки размера матрицы В

```
1 private: System::Void set_size_B_Click(System::Object^ sender, System::EventArgs^ e) {
2     this->errorProvider1->Clear();
3     int n, m;
4     bool resn = Int32::TryParse(this->set_size_B_lines->Text, n);
5     bool resm = Int32::TryParse(this->set_size_B_columns->Text, m);
6     if (!(resn && resm)) {
7         this->errorProvider1->SetError(this->input_B, "Неправильный формат ввода
            ↪ размера таблицы");
8         return;
9     }
10    else if (n <= 0 || m <= 0) {
11        this->errorProvider1->SetError(this->input_B, "Нельзя задать размер с 0");
12        return;
13    }
14
15
16    int line_count = this->input_B->RowCount;
17
18    for (int i = 0; i < line_count; i++) {
19        this->input_B->Rows->Remove(this->input_B->Rows[0]);
20    }
21    int column_count = this->input_B->ColumnCount;
22    for (int i = 0; i < column_count; i++) {
23        this->input_B->Columns->Remove(this->input_B->Columns[0]);
24    }
25
26
27    for (int i = 0; i < m; i++) {
28        this->input_B->Columns->Add("", "");
29    }
30
31    this->input_B->Rows->Add(n);
32 }
```

## ПРИЛОЖЕНИЕ Ж

### Фрагменты кода программы «Использование коллекций в Windows Forms»

Функция нахождения максимального нечетного элемента стека.

```
1 private: System::Void max_nech_btn_Click(System::Object^ sender, System::EventArgs^
   ↪ e) {
2     System::Collections::Generic::Stack<int> buf; //вспомогательный стек
3     int maxel = -1e9;
4     bool nech = false;
5     while (s.Count) { //пока стек не пуст
6         if (maxel < s.Peek() && s.Peek() % 2 != 0) {
7             maxel = s.Peek();
8             nech = true;
9         }
10        buf.Push(s.Peek()); //записываем во вспомогательный стек первый элемент
11        s.Pop(); //удаляем первый элемент из стека
12
13
14    }
15    while (buf.Count) { //пока вспомогательный стек не пуст
16        s.Push(buf.Peek()); //записываем в основной стек первый элемент вспомогательного
17        buf.Pop(); //удаляем из стека первый элемент
18    }
19    if (nech) this->max_nech->Text = System::Convert::ToString(maxel); //записываем
   ↪ результат в строку
20    else this->max_nech->Text = "Стек не содержит четных элементов";
21 }
```

## ПРИЛОЖЕНИЕ 3

### Фрагменты кода программы «Работа с файлами»

#### Функция сохранения таблицы в файл

```
1 private: System::Void FileSaveFilter_Click(System::Object^ sender, System::EventArgs^ e)
2     {
3         System::IO::Stream^ myStream;
4         if (this->saveFileDialog->ShowDialog() == System::Windows::Forms::DialogResult::OK) {
5             if ((myStream = saveFileDialog->OpenFile()) != nullptr) {
6                 System::IO::StreamWriter^ sw =
7                     gcnew System::IO::StreamWriter(myStream,
8                         System::Text::Encoding::GetEncoding(65001));
9                 for (int i = 0; i < this->FilteredTable->RowCount; i++) {
10
11                     sw->Write(this->FilteredTable->Rows[i]->Cells[0]->Value);
12                     sw->Write(" ");
13
14                     sw->Write(this->FilteredTable->Rows[i]->Cells[1]->Value);
15                     sw->Write(" ");
16
17                     sw->Write(this->FilteredTable->Rows[i]->Cells[2]->Value);
18                     sw->Write(" ");
19
20                     sw->Write(this->FilteredTable->Rows[i]->Cells[3]->Value);
21                     sw->Write(" ");
22
23                     sw->Write(this->FilteredTable->Rows[i]->Cells[4]->Value);
24                     sw->Write(" ");
25                     sw->Write(this->FilteredTable->Rows[i]->Cells[5]->Value);
26                     sw->Write(" ");
27                     sw->Write(this->FilteredTable->Rows[i]->Cells[6]->Value);
28                     if (i != this->FilteredTable->RowCount - 1) sw->WriteLine();
29                 }
30                 sw->Close();
31             }
32         }
```



## ПРИЛОЖЕНИЕ И

### Фрагменты кода программы «Приложение тест»

#### Функция при клике на кнопку "Продолжить"

```
1 System::Void act_Click(System::Object^ sender, System::EventArgs^ e) {
2     errorProvider1->Clear();
3     std::string answer = "";
4     if (iter != Questions->size()) {
5         answer = ReadAnswer();
6     }
7     // если ответ правильного формата
8     if (answer != "") {
9         int i = RandIndex->at(iter);
10        int flag = 0;
11        // проверка ответа
12        if (Questions->at(i).getRightAnswer() == answer) {
13            // выставление баллов за ответ
14            Questions->at(i).setResult(1);
15            flag = 1;
16        }
17        else {
18            // выставление баллов за ответ
19            Questions->at(i).setResult(0);
20        }
21        MessageForm^ msgForm = gcnew MessageForm();
22        msgForm->setFlag(flag);
23        msgForm->ShowDialog();
24        iter++;
25        if (iter != Questions->size()) {
26            countBox->Text = Convert::ToString(iter + 1);
27            ChooseForm();
28            int i = RandIndex->at(iter);
29            questBox->Text = gcnew String(Questions->at(i).getText().c_str());
30        }
31        else {
32            msgForm->result(CalcResult());
33            msgForm->setFlag(-1);
34            msgForm->ShowDialog();
35        }
36    }
37 }
```



## ПРИЛОЖЕНИЕ К

### Флешка с отчетом о выполненной работе

На приложенной флешке можно ознакомиться со следующими файлами:

**Папка** Otchet — L<sup>A</sup>T<sub>E</sub>X- вариант отчета о практике;

**Папка** task\_1 — задание № 1;

**Папка** task\_2 — задание № 2, вариант № 4;

**Папка** task\_3 — задание № 3, вариант № 5;

**Папка** task\_4 — задание № 4, вариант № 9;

**Папка** task\_5 — задание № 5, вариант № 12;

**Папка** task\_6 — задание № 6;

**Папка** task\_7 — задание № 7, вариант № 7;

**Папка** task\_8 — задание № 8, вариант № 11;

**Папка** task\_9 — задание № 9;

Otchet.pdf — отчет о практике.