

1 Работа с табличными данными. Часть 2

ЗАДАНИЕ. Необходимо представить двумерный массив в виде таблицы `DataGridView`, содержащей произвольное количество строк и столбцов. Предусмотреть возможность добавления и удаления строк и столбцов. Провести проверку на ввод чисел. Все строки, содержащие минимальный элемент, заменить строкой X.

Создано окно приложения, содержащее три элемента `DataGridView`, пять элементов `Button` и три элемента `Label`. Для отображения сообщений об ошибках в окно добавлен элемент `ErrorProvider`. Вид окна представлен на рисунке 1.

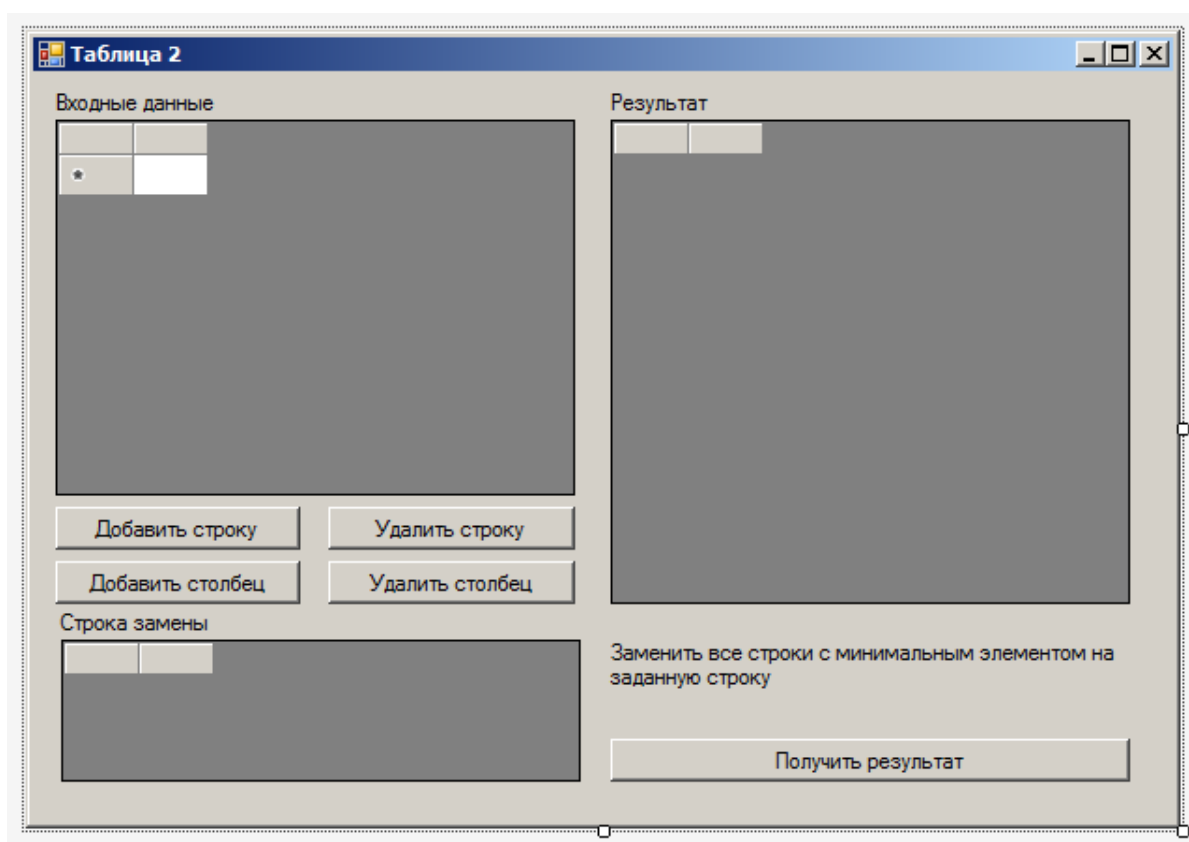


Рисунок 1 – Окно приложения «Таблица 2», открытое в конструкторе

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

Таблица 1 – Значения атрибутов элементов в приложении «Таблица 2»

Наименование атрибута	Значение
Для формы	
Text	Таблица 2
Для первой надписи	

(Name)	lblInput
Text	Входные данные
Для второй надписи	
(Name)	lblOutput
Text	Результат
Для третьей надписи	
(Name)	lblTask
Text	Заменить все строки с минимальным элементом на заданную строку
Для первой таблицы	
(Name)	dataGridInput
Для второй таблицы	
(Name)	dataGridOutput
AllowUserToAddRows	False
AllowUserToDeleteRows	False
ReadOnly	True
Для третьей таблицы	
(Name)	dataGridRow
AllowUserToAddRows	False
AllowUserToDeleteRows	False
Для первой кнопки	
(Name)	btnRowAdd
Text	Добавить строку
Для второй кнопки	
(Name)	btnRowRemove
Text	Удалить строку
Для третьей кнопки	
(Name)	btnColAdd
Text	Добавить столбец
Для четвертой кнопки	
(Name)	btnColRemove
Text	Удалить столбец
Для пятой кнопки	
(Name)	btnColAdd
Text	Получить результат
Для обработчика ошибок	
(Name)	errorProvider

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

```
1 this->dataGridInput->Rows->Add(1);
```

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

```
1  this->dataGridInput->Rows->Add(1);
2  }
3  private: System::Void btnColAdd_Click(System::Object^ sender, System::EventArgs^ e) {
4      //добавляем колонки с шириной по умолчанию 40 пикселей
5      int colCount = this->dataGridInput->ColumnCount;
6      this->dataGridInput->Columns->Add("ColumnIn" + colCount, String::Empty);
7      this->dataGridInput->Columns[colCount]->Width = 40;
8      this->dataGridRow->Columns->Add("ColumnRow" + colCount, String::Empty);
9      this->dataGridRow->Columns[colCount]->Width = 40;
```

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

```
1  //добавляем колонки с шириной по умолчанию 40 пикселей
2  int colCount = this->dataGridInput->ColumnCount;
3  this->dataGridInput->Columns->Add("ColumnIn" + colCount, String::Empty);
4  this->dataGridInput->Columns[colCount]->Width = 40;
5  this->dataGridRow->Columns->Add("ColumnRow" + colCount, String::Empty);
6  this->dataGridRow->Columns[colCount]->Width = 40;
```

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

```
1  if ( this->dataGridInput->ColumnCount > 1) {
2      int i = 0;
3      i = this->dataGridInput->ColumnCount - 1;
4      this->dataGridInput->Columns->Remove(this->dataGridInput->Columns[i]);
5      this->dataGridRow->Columns->Remove(this->dataGridRow->Columns[i]);
6  }
```

На нажатие кнопки «Получить результат» установлено выполнение следующего кода:

```
1  //чистим ошибки. очищаем вывод
2  this->errorProvider->SetError(this->dataGridInput, String::Empty);
3  this->errorProvider->SetError(this->dataGridRow, String::Empty);
4  this->dataGridOutput->Columns->Clear();
5  /* Коды ошибок
6      * 1 — нет ни одной строки
7      * 2 — в таблице есть пустые ячейки
8      * 3 — в таблице есть не целые числа
9      */
10 //проверяем матрицу ввода и ищем минимум
11 int min;
12 try {
13     int errorCode;
14     //проверка на количество строк
```

```

15     if (this->dataGridInput->RowCount < 2) {
16         throw 1;
17     }
18     // min = первому элементу с проверкой на пустоту и число
19     if (!this->dataGridInput->Rows[0]->Cells[0]->Value) {
20         throw 2;
21     }
22     bool resultParse;
23     resultParse = Int32::TryParse(this->dataGridInput->Rows[0]->Cells[0]->Value->
        ToString(), min);
24     if (!resultParse) {
25         throw 3;
26     }
27     //ищем минимум
28     for (int i = 0; i < this->dataGridInput->RowCount - 1; i++) {
29         for (int j = 0; j < this->dataGridInput->ColumnCount; j++) {
30             //проверка на пустоту
31             if (!this->dataGridInput->Rows[i]->Cells[j]->Value) {
32                 throw 2;
33             }
34             //проверка на число
35             int current;
36             resultParse = Int32::TryParse(this->dataGridInput->Rows[i]->Cells[j]->Value
                ->ToString(), current);
37             if (!resultParse) {
38                 throw 3;
39             }
40             //проверка на минимум
41             if (current < min)
42                 min = current;
43         }
44     }
45 }
46 catch (int errorCode) {
47     switch (errorCode) {
48     case 1:
49         this->errorProvider->SetError(this->dataGridInput, "В_таблице_нет_ни_одной_
        _строки");
50         break;
51     case 2:
52         this->errorProvider->SetError(this->dataGridInput, "В_таблице_есть_пустые_

```

```

        ячейки");
53     break;
54     case 3:
55         this->errorProvider->SetError(this->dataGridInput, "В_таблице_есть_не_целые_
            _числа");
56     break;
57 }
58 }
59 /*
60 * 1 --в строке замены есть пустые ячейки
61 * 2 --в строке замены есть не целые числа
62 */
63 //проверяем строку замены
64 try {
65     for (int i = 0; i < this->dataGridRow->ColumnCount; i++) {
66         bool resultParse;
67         //проверка на пустоту
68         if (!(String ^)this->dataGridRow->Rows[0]->Cells[i]->Value) {
69             throw 1;
70         }
71         //проверка на число
72         int current;
73         resultParse = Int32::TryParse(this->dataGridRow->Rows[0]->Cells[i]->Value->
            ToString(), current);
74         if (!resultParse) {
75             throw 2;
76         }
77     }
78 }
79 catch (int errorCode) {
80     switch (errorCode) {
81     case 1:
82         this->errorProvider->SetError(this->dataGridRow, "В_строке_есть_пустые_
            ячейки");
83         break;
84     case 2:
85         this->errorProvider->SetError(this->dataGridRow, "В_строке_есть_не_целые_
            числа");
86         break;
87     }
88 }

```

```

89  //заменяем строки
90  try {
91      //если есть ошибки в заполнении
92      if (this->errorProvider->GetError(this->dataGridInput)->ToString() != String::
          Empty
93          || this->errorProvider->GetError(this->dataGridRow)->ToString() != String::
          Empty) {
94          throw 0;
95      }
96      //добавляем столбцы в матрице вывода
97      for (int i = 0; i < this->dataGridInput->ColumnCount; i++) {
98          this->dataGridOutput->Columns->Add("ColumnOutput" + i, String::Empty);
99          this->dataGridOutput->Columns[i]->Width = 40;
100     }
101     //добавляем строки
102     this->dataGridOutput->Rows->Add(this->dataGridInput->RowCount - 1);
103
104     //заполняем ячейки
105     for (int i = 0; i < this->dataGridInput->RowCount - 1; i++) {
106         bool f = false;
107         for (int j = 0; j < this->dataGridInput->ColumnCount; j++) {
108             if (Int32::Parse(this->dataGridInput->Rows[i]->Cells[j]->Value->ToString())
                == min) {
109                 f = true;
110             }
111             //если есть минимальный
112             if (f) {
113                 //переписываем из строки для замены
114                 for (int j = 0; j < this->dataGridOutput->ColumnCount; j++) {
115                     this->dataGridOutput->Rows[i]->Cells[j]->Value = this->dataGridRow
                        ->Rows[0]->Cells[j]->Value;
116                 }
117             }
118             else {
119                 //переписываем из исходной строки
120                 for (int j = 0; j < this->dataGridOutput->ColumnCount; j++) {
121                     this->dataGridOutput->Rows[i]->Cells[j]->Value = this->dataGridInput
                        ->Rows[i]->Cells[j]->Value;
122                 }
123             }
124         }

```

```

125     }
126 }
127 catch (int errorCode) {
128     }столбец

```

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

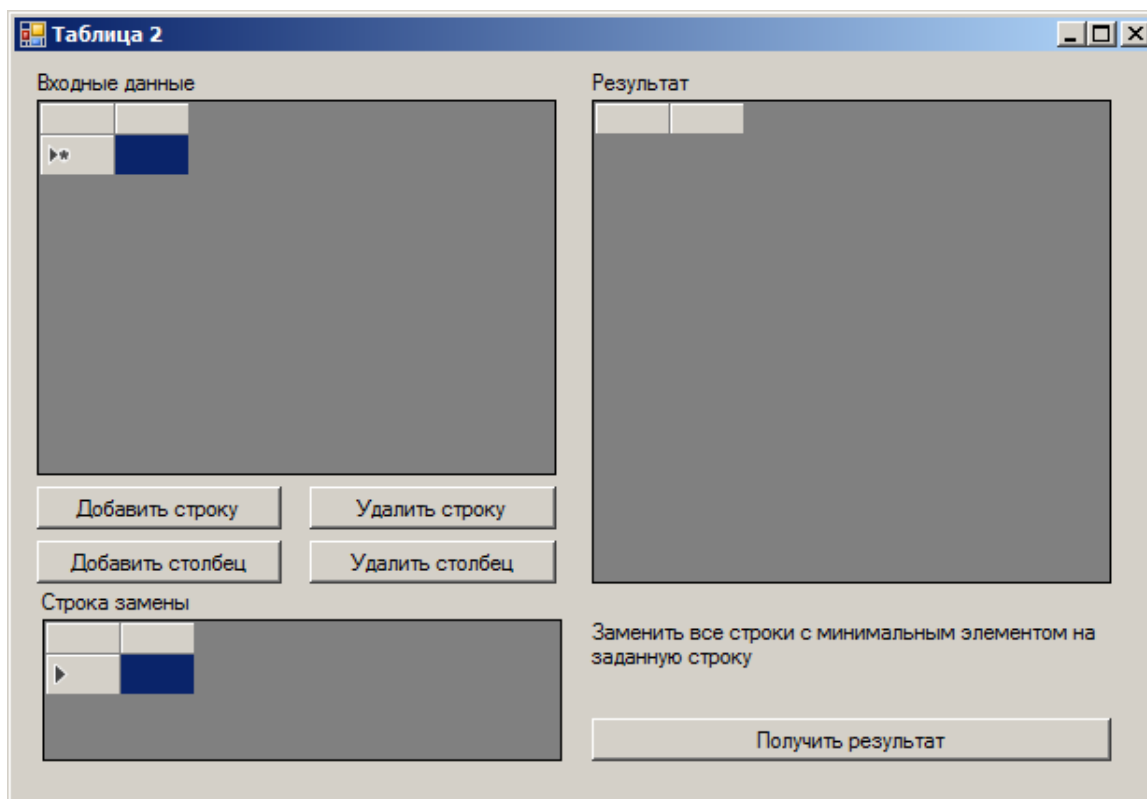


Рисунок 2 – Окно приложения «Таблица 2»: начальный запуск

При нажатии кнопки «Добавить строку» в таблицу добавляется новая строка (см. рисунок 3).

При нажатии кнопки «Удалить строку» из таблицы удаляется выбранная строка (см. рисунок 4).

При нажатии кнопки «Добавить столбец» в таблицу и строку замены добавляется новый столбец (см. рисунок 5).

При нажатии кнопки «Удалить столбец» из таблицы и из строки замены удаляется выбранный столбец (см. рисунок 6).

При корректном вводе все строки, в которых есть минимальный элемент, заменяются строкой замены. Результат выводится в новую таблицу (см. рисунок 7).

При попытке замены строк из таблицы, в которой нет ни одной строки, возникает сообщение об ошибке (см. рисунок 8).

При попытке замены строк из таблицы, в которой есть пустые ячейки, возникает сообщение об ошибке (см. рисунок 9).

При попытке замены строк из таблицы, в ячейках которой есть не целые числа, возникает сообщение об ошибке (см. рисунок 10).

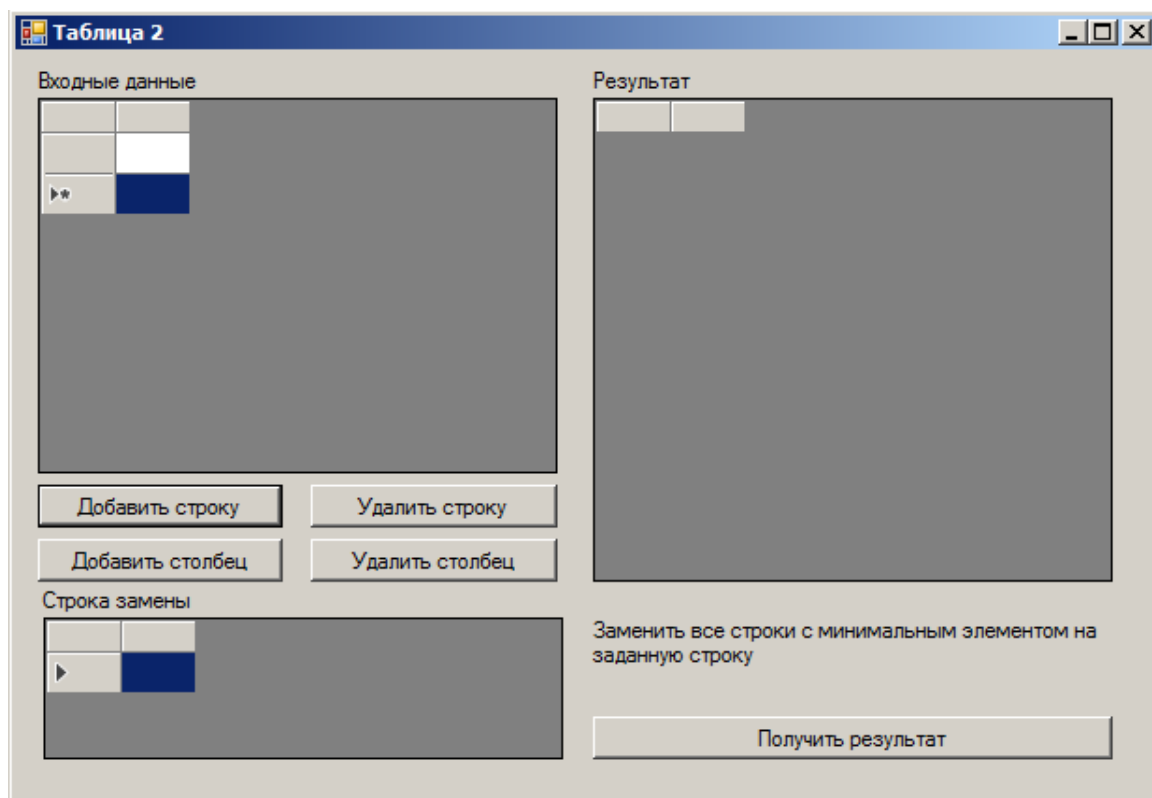


Рисунок 3 – Окно приложения «Таблица 2»: добавление новой строки

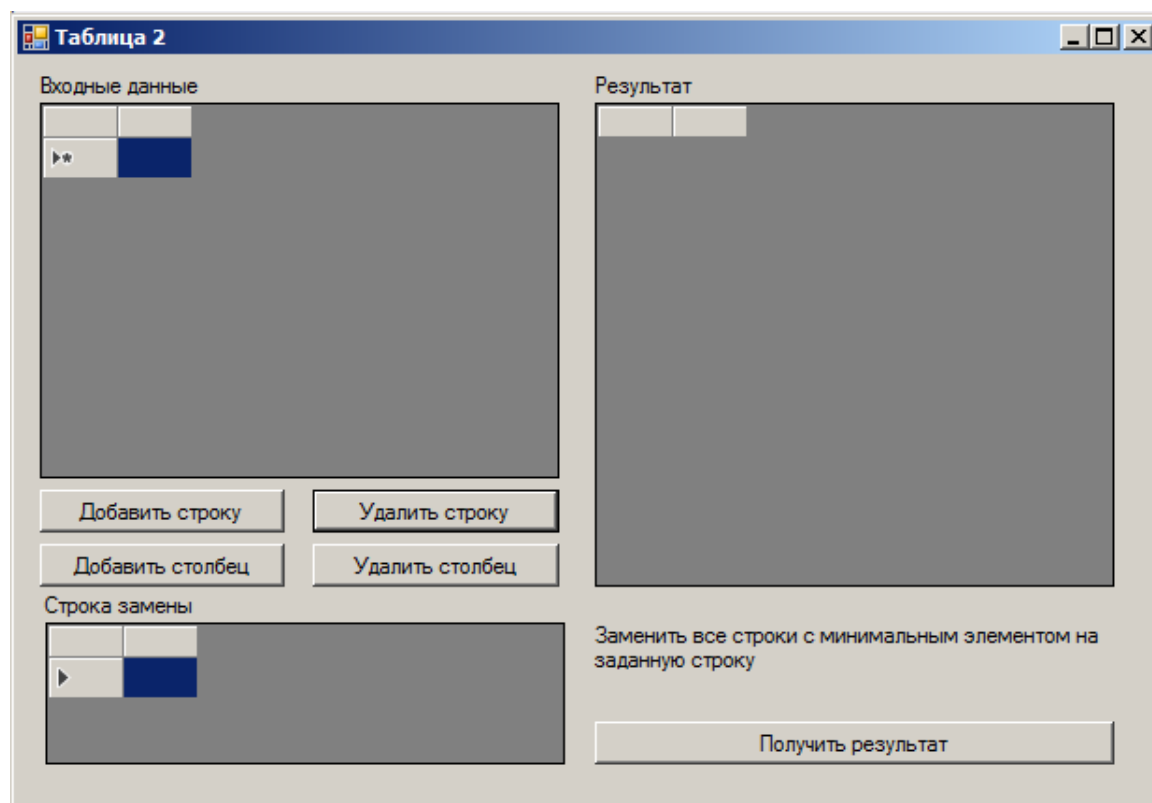


Рисунок 4 – Окно приложения «Таблица 2»: удаление строки

При попытке ввода пустых ячеек в строке замены возникает сообщение об ошибке (см. рисунок 11).

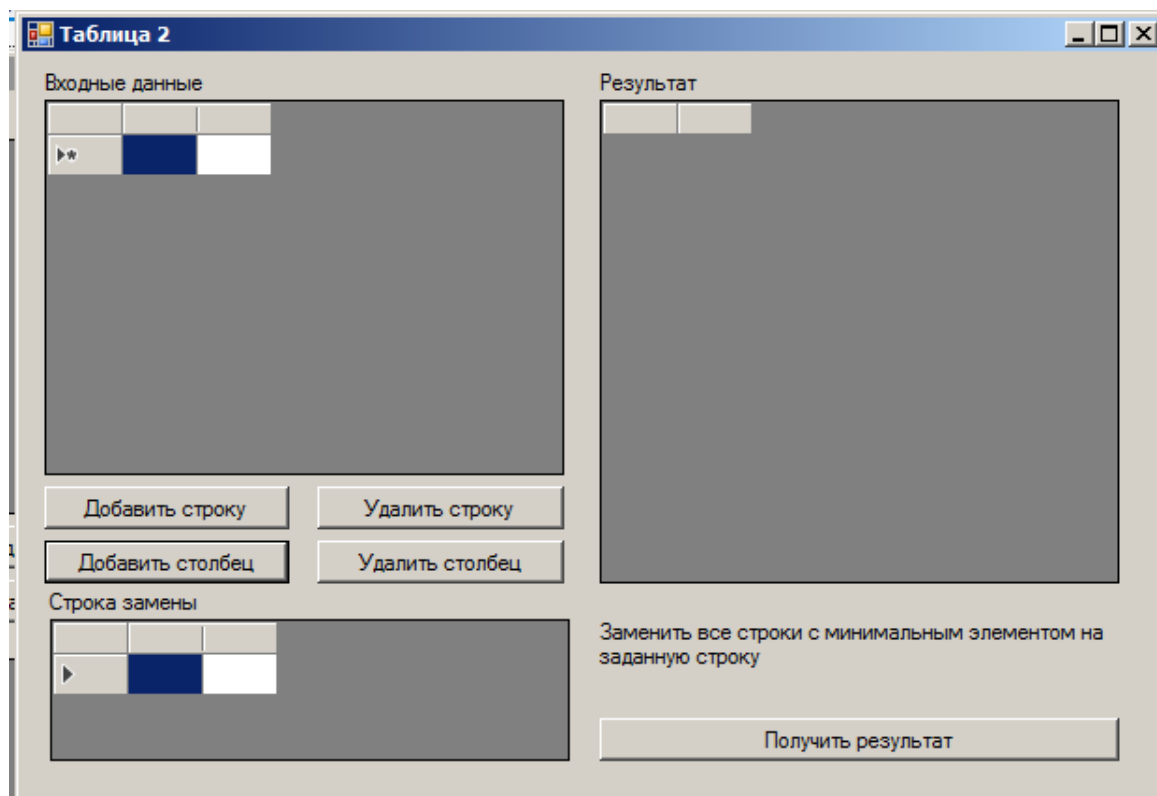


Рисунок 5 – Окно приложения «Таблица 2»: добавление столбца

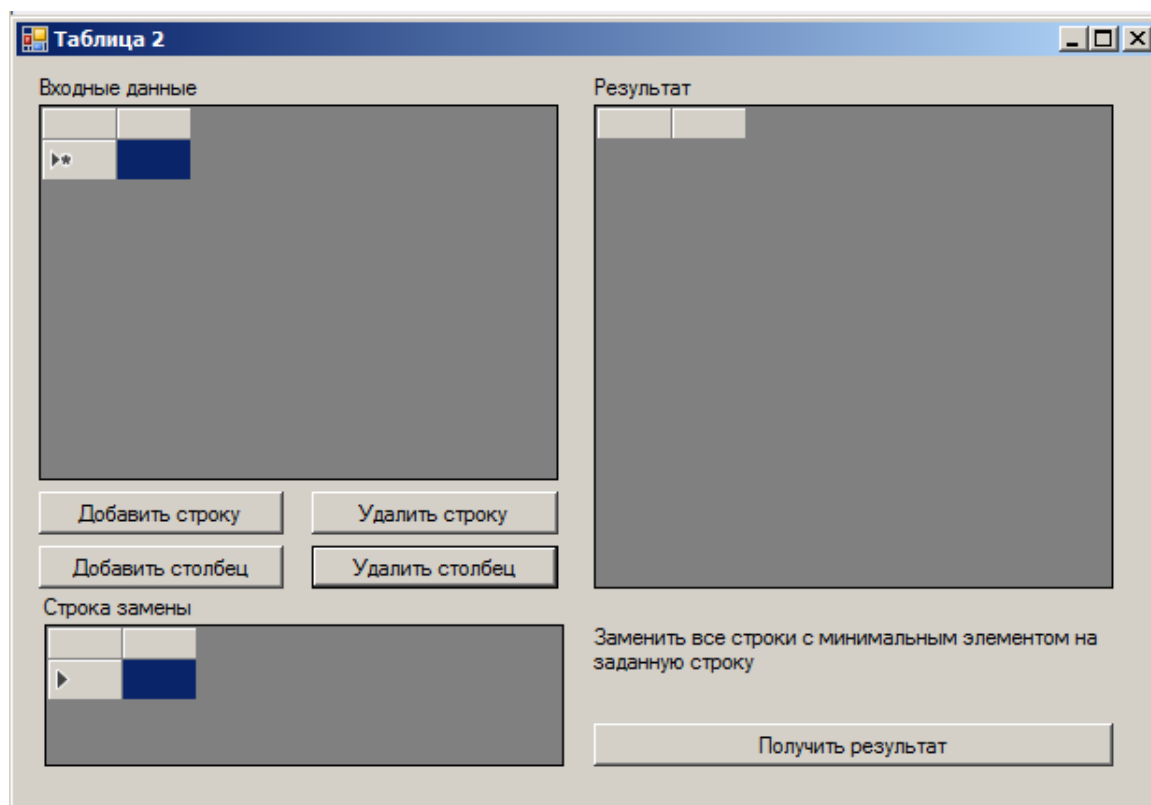


Рисунок 6 – Окно приложения «Таблица 2»: удаление столбца

При попытке ввода строки, в ячейках которой есть не целые числа, возникает сообщение об ошибке (см. рисунок 12).

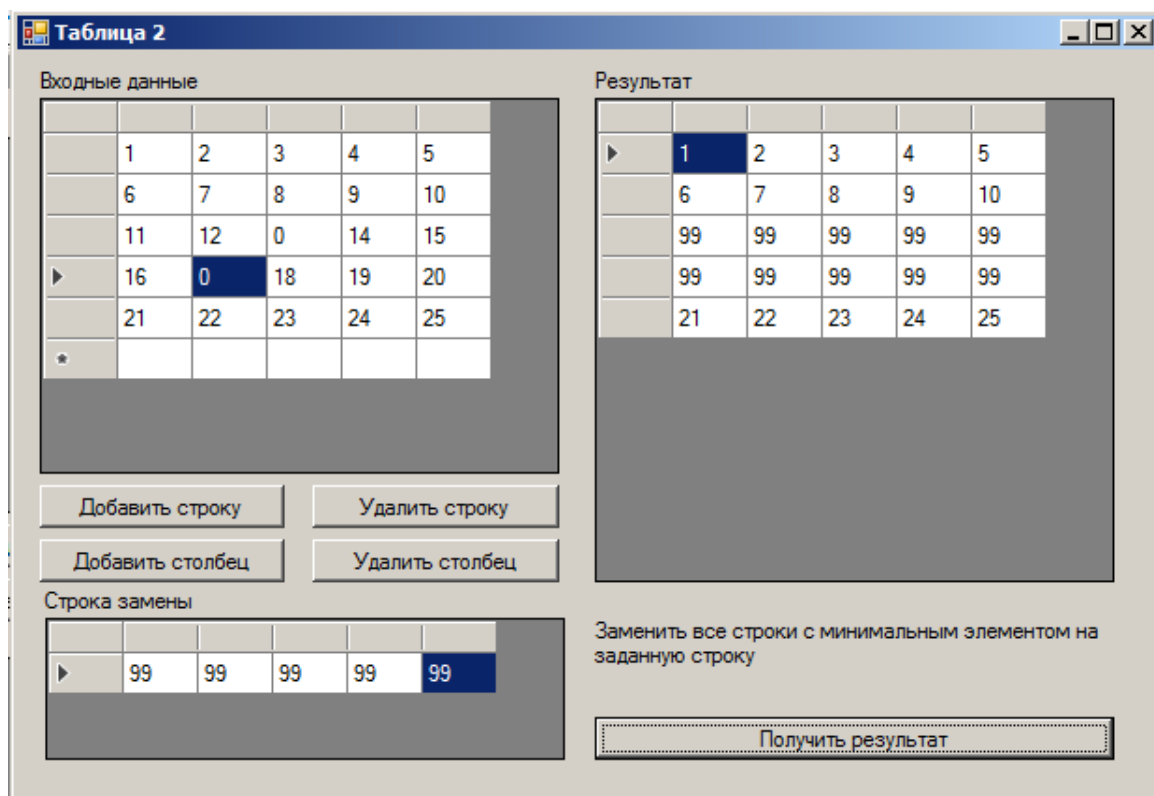


Рисунок 7 – Окно приложения «Таблица 2»: корректные данные

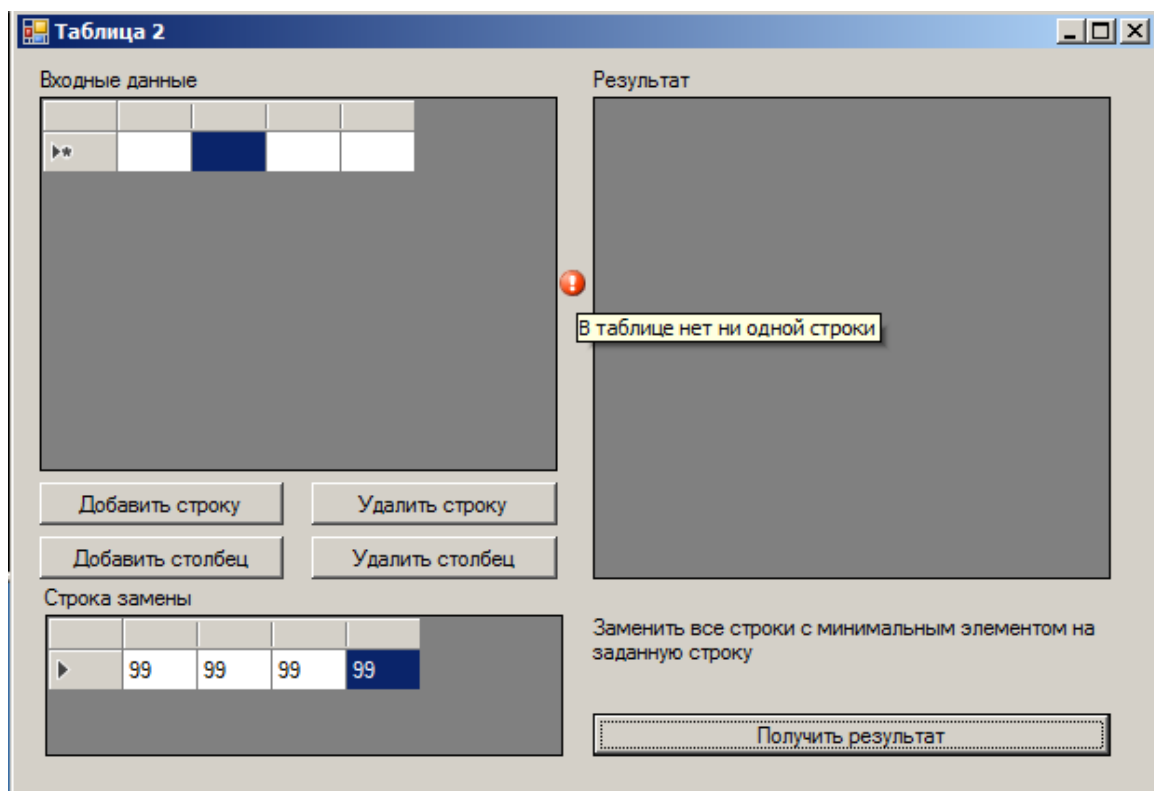


Рисунок 8 – Окно приложения «Таблица 2»: сообщение об отсутствии строк

Полный код программы приведен в приложении А.

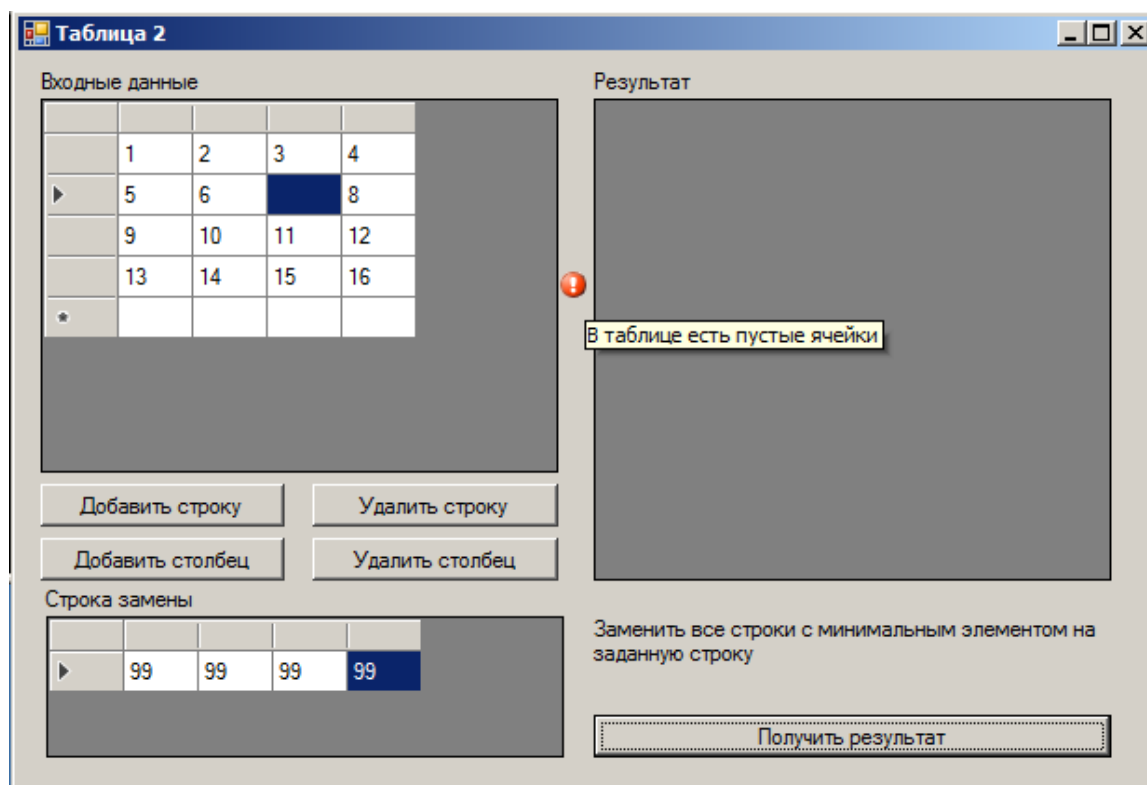


Рисунок 9 – Окно приложения «Таблица 2»: сообщение о пустых ячейках

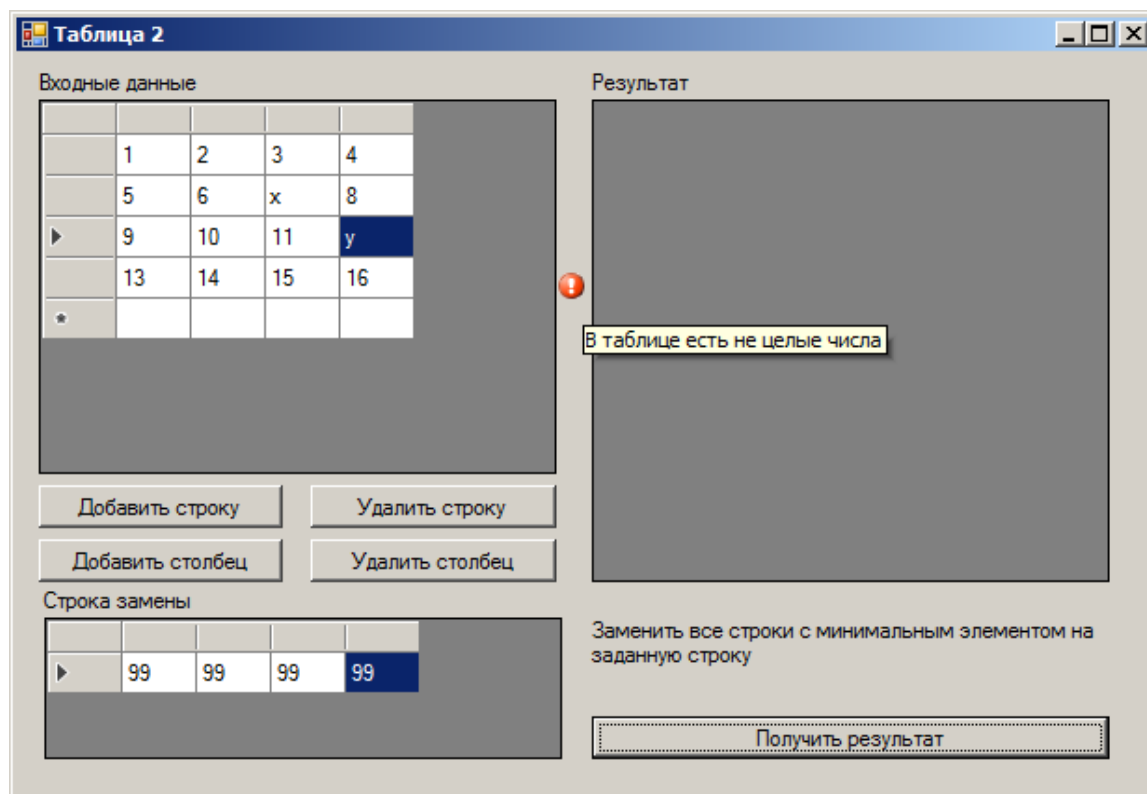


Рисунок 10 – Окно приложения «Таблица 2»: сообщение о наличии не целых чисел в строке

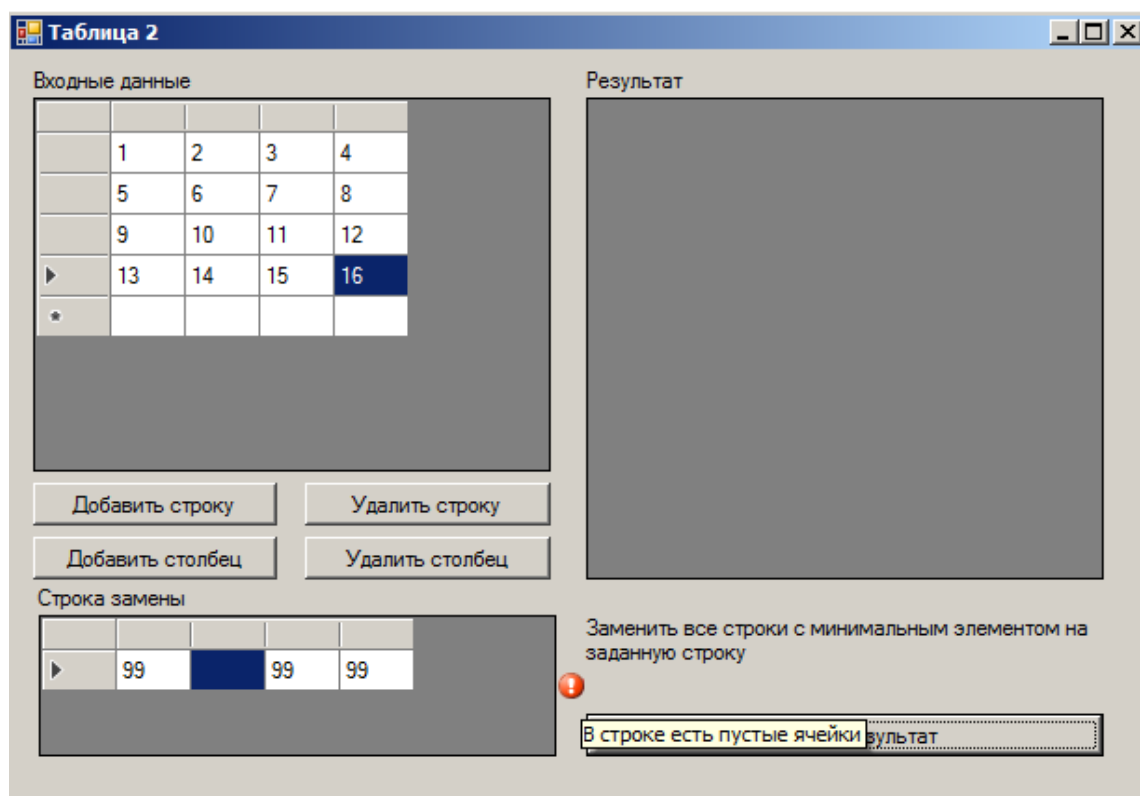


Рисунок 11 – Окно приложения «Таблица 2»: сообщение о пустых ячейках строки

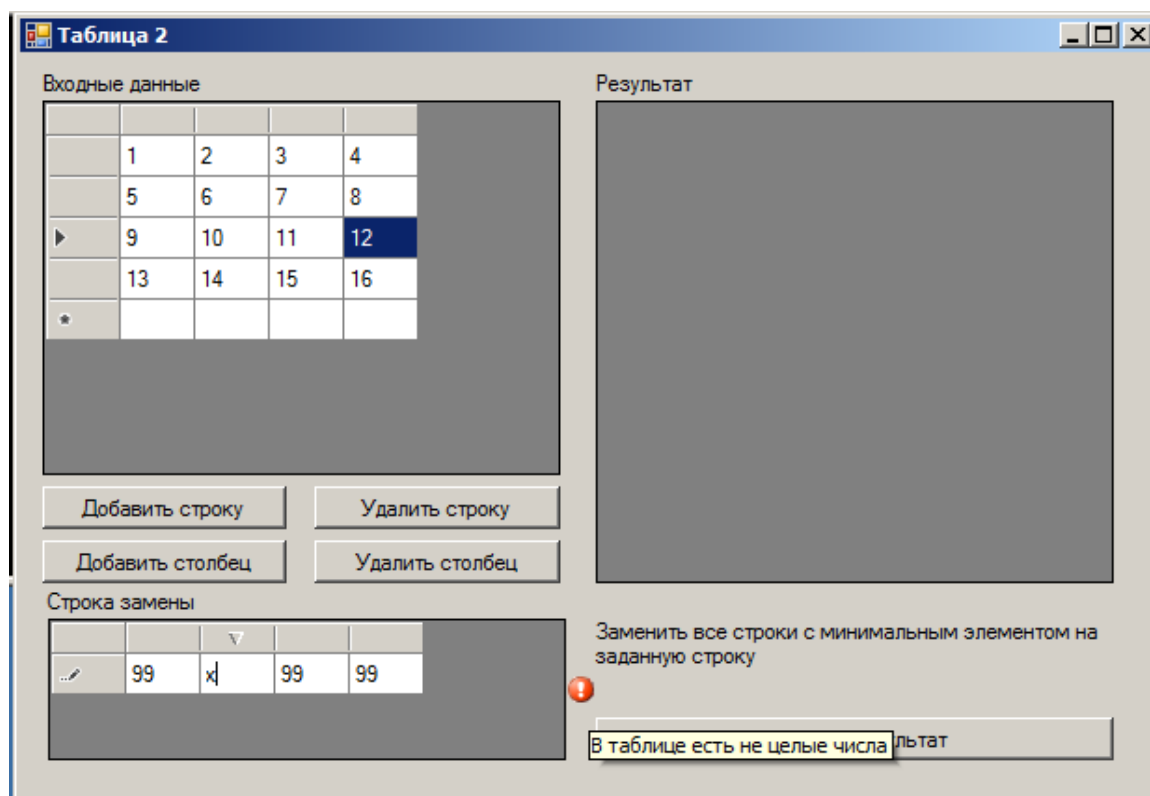


Рисунок 12 – Окно приложения «Таблица 2»: сообщение о наличии не целых чисел в строке

ПРИЛОЖЕНИЕ А

Листинг файлов проекта

1.1 Листинг файла table.cpp

```
1  #include "MyForm.h"
2
3  using namespace table2;
4
5  [STAThreadAttribute]
6  int main(array<System::String ^> ^args) {
7
8      Application::EnableVisualStyles();
9      Application::SetCompatibleTextRenderingDefault(false);
10
11     Application::Run(gcnew MyForm());
12     return 0;
13 }
```

1.2 Листинг файла MyForm.h

```
1  #pragma once
2  namespace table2 {
3      using namespace System;
4      using namespace System::ComponentModel;
5      using namespace System::Collections;
6      using namespace System::Windows::Forms;
7      using namespace System::Data;
8      using namespace System::Drawing;
9      /// <summary>
10     /// Сводка для MyForm
11     /// </summary>
12     public ref class MyForm : public System::Windows::Forms::Form
13     {
14     public:
15         MyForm(void)
16         {
17             InitializeComponent();
18             this->dataGridView->Rows->Add(1);
19         }
20     protected:
21         /// <summary>
22         /// Освободить все используемые ресурсы.
23         /// </summary>
```

```

24     ~MyForm()
25     {
26         if (components)
27         {
28             delete components;
29         }
30     }
31     private: System::Windows::Forms::DataGridView^ dataGridInput;
32     private: System::Windows::Forms::Label^ lblInput;
33     private: System::Windows::Forms::Label^ lblOutput;
34     private: System::Windows::Forms::DataGridView^ dataGridOutput;
35     private: System::Windows::Forms::Button^ btnRowAdd;
36     private: System::Windows::Forms::Button^ btnRowRemove;
37     private: System::Windows::Forms::Button^ btnColRemove;
38     private: System::Windows::Forms::Button^ btnColAdd;
39     private: System::Windows::Forms::Button^ btnResult;
40     private: System::Windows::Forms::Label^ lblTask;
41     private: System::Windows::Forms::DataGridView^ dataGridRow;
42     private: System::Windows::Forms::Label^ lblRow;
43     private: System::Windows::Forms::DataGridViewTextBoxColumn^ ColumnIn0;
44     private: System::Windows::Forms::DataGridViewTextBoxColumn^ ColumnOut0;
45     private: System::Windows::Forms::DataGridViewTextBoxColumn^ ColumnRow0;
46     private: System::Windows::Forms::ErrorProvider^ errorProvider;
47     private: System::ComponentModel::IContainer^ components;
48     private:
49         /// <summary>
50         /// Обязательная переменная конструктора.
51         /// </summary>
52     #pragma region Windows Form Designer generated code
53         /// <summary>
54         /// Требуемый метод для поддержки конструктора — не изменяйте
55         /// содержимое этого метода с помощью редактора кода.
56         /// </summary>
57     void InitializeComponent(void)
58     {
59         this->components = (gcnew System::ComponentModel::Container());
60         this->dataGridInput = (gcnew System::Windows::Forms::DataGridView());
61         this->ColumnIn0 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());
62         ;
63         this->lblInput = (gcnew System::Windows::Forms::Label());
64         this->lblOutput = (gcnew System::Windows::Forms::Label());

```

```

64 this->dataGridViewOutput = (gcnew System::Windows::Forms::DataGridView());
65 this->ColumnOut0 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn
    ());
66 this->btnRowAdd = (gcnew System::Windows::Forms::Button());
67 this->btnRowRemove = (gcnew System::Windows::Forms::Button());
68 this->btnColRemove = (gcnew System::Windows::Forms::Button());
69 this->btnColAdd = (gcnew System::Windows::Forms::Button());
70 this->btnResult = (gcnew System::Windows::Forms::Button());
71 this->lblTask = (gcnew System::Windows::Forms::Label());
72 this->dataGridViewRow = (gcnew System::Windows::Forms::DataGridView());
73 this->ColumnRow0 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn
    ());
74 this->lblRow = (gcnew System::Windows::Forms::Label());
75 this->errorProvider = (gcnew System::Windows::Forms::ErrorProvider(this->
    components));
76 (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridViewInput)
    )->BeginInit();
77 (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(this->
    dataGridViewOutput))->BeginInit();
78 (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridViewRow))
    ->BeginInit();
79 (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(this->errorProvider))
    ->BeginInit();
80 this->SuspendLayout();
81 //
82 // dataGridViewInput
83 //
84 this->dataGridViewInput->ColumnHeadersHeightSizeMode = System::Windows::Forms::
    DataGridViewColumnHeadersHeightSizeMode::AutoSize;
85 this->dataGridViewInput->Columns->AddRange(gcnew cli::array< System::Windows::
    Forms::DataGridViewColumn^ >(1) { this->ColumnIn0 });
86 this->dataGridViewInput->Location = System::Drawing::Point(12, 25);
87 this->dataGridViewInput->Name = L"dataGridInput";
88 this->dataGridViewInput->Size = System::Drawing::Size(278, 201);
89 this->dataGridViewInput->TabIndex = 0;
90 //
91 // ColumnIn0
92 //
93 this->ColumnIn0->HeaderText = L"";
94 this->ColumnIn0->Name = L"ColumnIn0";
95 this->ColumnIn0->Width = 40;

```

```

96    //
97    // lblInput
98    //
99    this->lblInput->AutoSize = true;
100   this->lblInput->Location = System::Drawing::Point(9, 9);
101   this->lblInput->Name = L"lblInput";
102   this->lblInput->Size = System::Drawing::Size(92, 13);
103   this->lblInput->TabIndex = 2;
104   this->lblInput->Text = L"Входные_данные";
105   //
106   // lblOutput
107   //
108   this->lblOutput->AutoSize = true;
109   this->lblOutput->Location = System::Drawing::Point(306, 9);
110   this->lblOutput->Name = L"lblOutput";
111   this->lblOutput->Size = System::Drawing::Size(59, 13);
112   this->lblOutput->TabIndex = 3;
113   this->lblOutput->Text = L"Результат";
114   //
115   // dataGridOutput
116   //
117   this->dataGridOutput->AllowUserToAddRows = false;
118   this->dataGridOutput->AllowUserToDeleteRows = false;
119   this->dataGridOutput->ColumnHeadersHeightSizeMode = System::Windows::Forms::
        DataGridViewColumnHeadersHeightSizeMode::AutoSize;
120   this->dataGridOutput->Columns->AddRange(gcnew cli::array< System::Windows::
        Forms::DataGridViewColumn ^ >(1) { this->ColumnOut0 });
121   this->dataGridOutput->Location = System::Drawing::Point(309, 25);
122   this->dataGridOutput->Name = L"dataGridOutput";
123   this->dataGridOutput->ReadOnly = true;
124   this->dataGridOutput->Size = System::Drawing::Size(278, 259);
125   this->dataGridOutput->TabIndex = 4;
126   //
127   // ColumnOut0
128   //
129   this->ColumnOut0->HeaderText = L"";
130   this->ColumnOut0->Name = L"ColumnOut0";
131   this->ColumnOut0->ReadOnly = true;
132   this->ColumnOut0->Width = 40;
133   //
134   // btnRowAdd

```



```

135 //
136 this->btnRowAdd->Location = System::Drawing::Point(12, 232);
137 this->btnRowAdd->Name = L"btnRowAdd";
138 this->btnRowAdd->Size = System::Drawing::Size(131, 23);
139 this->btnRowAdd->TabIndex = 5;
140 this->btnRowAdd->Text = L"Добавить_строку";
141 this->btnRowAdd->UseVisualStyleBackColor = true;
142 this->btnRowAdd->Click += gcnew System::EventHandler(this, &MyForm::
    btnRowAdd_Click);
143 //
144 // btnRowRemove
145 //
146 this->btnRowRemove->Location = System::Drawing::Point(158, 232);
147 this->btnRowRemove->Name = L"btnRowRemove";
148 this->btnRowRemove->Size = System::Drawing::Size(132, 23);
149 this->btnRowRemove->TabIndex = 8;
150 this->btnRowRemove->Text = L"Удалить_строку";
151 this->btnRowRemove->UseVisualStyleBackColor = true;
152 this->btnRowRemove->Click += gcnew System::EventHandler(this, &MyForm::
    btnRowRemove_Click);
153 //
154 // btnColRemove
155 //
156 this->btnColRemove->Location = System::Drawing::Point(158, 261);
157 this->btnColRemove->Name = L"btnColRemove";
158 this->btnColRemove->Size = System::Drawing::Size(132, 23);
159 this->btnColRemove->TabIndex = 9;
160 this->btnColRemove->Text = L"Удалить_столбец";
161 this->btnColRemove->UseVisualStyleBackColor = true;
162 this->btnColRemove->Click += gcnew System::EventHandler(this, &MyForm::
    btnColRemove_Click);
163 //
164 // btnColAdd
165 //
166 this->btnColAdd->Location = System::Drawing::Point(12, 261);
167 this->btnColAdd->Name = L"btnColAdd";
168 this->btnColAdd->Size = System::Drawing::Size(131, 23);
169 this->btnColAdd->TabIndex = 10;
170 this->btnColAdd->Text = L"Добавить_столбец";
171 this->btnColAdd->UseVisualStyleBackColor = true;
172 this->btnColAdd->Click += gcnew System::EventHandler(this, &MyForm::

```

```

        btnColAdd_Click);
173 //
174 // btnResult
175 //
176 this->btnResult->Location = System::Drawing::Point(309, 356);
177 this->btnResult->Name = L"btnResult";
178 this->btnResult->Size = System::Drawing::Size(278, 23);
179 this->btnResult->TabIndex = 11;
180 this->btnResult->Text = L"Получить_результат";
181 this->btnResult->UseVisualStyleBackColor = true;
182 this->btnResult->Click += gcnew System::EventHandler(this, &MyForm::
        btnResult_Click);
183 //
184 // lblTask
185 //
186 this->lblTask->Location = System::Drawing::Point(306, 303);
187 this->lblTask->Name = L"lblTask";
188 this->lblTask->Size = System::Drawing::Size(281, 41);
189 this->lblTask->TabIndex = 12;
190 this->lblTask->Text = L"Заменить_все_строки_с_минимальным_элементом_на_
        заданную_строку";
191 //
192 // dataGridRow
193 //
194 this->dataGridRow->AllowUserToAddRows = false;
195 this->dataGridRow->AllowUserToDeleteRows = false;
196 this->dataGridRow->ColumnHeadersHeightSizeMode = System::Windows::Forms::
        DataGridViewColumnHeadersHeightSizeMode::AutoSize;
197 this->dataGridRow->Columns->AddRange(gcnew cli::array< System::Windows::
        Forms::DataGridViewColumn ^ >(1) { this->ColumnRow0 });
198 this->dataGridRow->Location = System::Drawing::Point(15, 303);
199 this->dataGridRow->Name = L"dataGridRow";
200 this->dataGridRow->Size = System::Drawing::Size(278, 76);
201 this->dataGridRow->TabIndex = 13;
202 //
203 // ColumnRow0
204 //
205 this->ColumnRow0->HeaderText = L"";
206 this->ColumnRow0->Name = L"ColumnRow0";
207 this->ColumnRow0->Width = 40;
208 //

```

```

209 // lblRow
210 //
211 this->lblRow->AutoSize = true;
212 this->lblRow->Location = System::Drawing::Point(12, 287);
213 this->lblRow->Name = L"lblRow";
214 this->lblRow->Size = System::Drawing::Size(86, 13);
215 this->lblRow->TabIndex = 14;
216 this->lblRow->Text = L"Строка_замены";
217 //
218 // errorProvider
219 //
220 this->errorProvider->ContainerControl = this;
221 //
222 // MyForm
223 //
224 this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
225 this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
226 this->ClientSize = System::Drawing::Size(610, 400);
227 this->Controls->Add(this->lblRow);
228 this->Controls->Add(this->dataGridRow);
229 this->Controls->Add(this->lblTask);
230 this->Controls->Add(this->btnResult);
231 this->Controls->Add(this->btnColAdd);
232 this->Controls->Add(this->btnColRemove);
233 this->Controls->Add(this->btnRowRemove);
234 this->Controls->Add(this->btnRowAdd);
235 this->Controls->Add(this->dataGridOutput);
236 this->Controls->Add(this->lblOutput);
237 this->Controls->Add(this->lblInput);
238 this->Controls->Add(this->dataGridInput);
239 this->Name = L"MyForm";
240 this->Text = L"Таблица_2";
241 (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridInput)
    )->EndInit();
242 (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(this->
    dataGridOutput))->EndInit();
243 (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridRow))
    ->EndInit();
244 (cli :: safe_cast<System::ComponentModel::ISupportInitialize^>(this->errorProvider))
    ->EndInit();
245 this->ResumeLayout(false);

```

```

246     this->PerformLayout();
247 }
248 #pragma endregion
249 private: System::Void btnRowRemove_Click(System::Object^ sender, System::EventArgs^ e
    ) {
250     if (!this->dataGridInput->CurrentRow->IsNewRow) {
251         int i = this->dataGridInput->CurrentRow->Index;
252         this->dataGridInput->Rows->Remove(this->dataGridInput->Rows[i]);
253     }
254 }
255 private: System::Void btnRowAdd_Click(System::Object^ sender, System::EventArgs^ e) {
256     this->dataGridInput->Rows->Add(1);
257 }
258 private: System::Void btnColAdd_Click(System::Object^ sender, System::EventArgs^ e) {
259     //добавляем колонки с шириной по умолчанию 40 пикселей
260     int colCount = this->dataGridInput->ColumnCount;
261     this->dataGridInput->Columns->Add("ColumnIn" + colCount, String::Empty);
262     this->dataGridInput->Columns[colCount]->Width = 40;
263     this->dataGridRow->Columns->Add("ColumnRow" + colCount, String::Empty);
264     this->dataGridRow->Columns[colCount]->Width = 40;
265 }
266 private: System::Void btnColRemove_Click(System::Object^ sender, System::EventArgs^ e)
    {
267     if (this->dataGridInput->ColumnCount > 1) {
268         int i = 0;
269         i = this->dataGridInput->ColumnCount - 1;
270         this->dataGridInput->Columns->Remove(this->dataGridInput->Columns[i]);
271         this->dataGridRow->Columns->Remove(this->dataGridRow->Columns[i]);
272     }
273 }
274 private: System::Void btnResult_Click(System::Object^ sender, System::EventArgs^ e) {
275     //чистим ошибки. очищаем вывод
276     this->errorProvider->SetError(this->dataGridInput, String::Empty);
277     this->errorProvider->SetError(this->dataGridRow, String::Empty);
278     this->dataGridOutput->Columns->Clear();
279     /* Коды ошибок
280     * 1 -- нет ни одной строки
281     * 2 -- в таблице есть пустые ячейки
282     * 3 -- в таблице есть не целые числа
283     */
284     //проверяем матрицу ввода и ищем минимум

```

```

285  int min;
286  try {
287      int errorCode;
288      //проверка на количество строк
289      if (this->dataGridInput->RowCount < 2) {
290          throw 1;
291      }
292      // min = первому элементу с проверкой на пустоту и число
293      if (!this->dataGridInput->Rows[0]->Cells[0]->Value) {
294          throw 2;
295      }
296      bool resultParse;
297      resultParse = Int32::TryParse(this->dataGridInput->Rows[0]->Cells[0]->Value->
          ToString(), min);
298      if (!resultParse) {
299          throw 3;
300      }
301      //ищем минимум
302      for (int i = 0; i < this->dataGridInput->RowCount - 1; i++) {
303          for (int j = 0; j < this->dataGridInput->ColumnCount; j++) {
304              //проверка на пустоту
305              if (!this->dataGridInput->Rows[i]->Cells[j]->Value) {
306                  throw 2;
307              }
308              //проверка на число
309              int current;
310              resultParse = Int32::TryParse(this->dataGridInput->Rows[i]->Cells[j]->Value
                  ->ToString(), current);
311              if (!resultParse) {
312                  throw 3;
313              }
314              //проверка на минимум
315              if (current < min)
316                  min = current;
317          }
318      }
319  }
320  catch (int errorCode) {
321      switch (errorCode) {
322          case 1:
323              this->errorProvider->SetError(this->dataGridInput, "В_таблице_нет_ни_одной

```

```

        _строки");
324     break;
325     case 2:
326         this->errorProvider->SetError(this->dataGridInput, "В_таблице_есть_пустые_
            ячейки");
327         break;
328     case 3:
329         this->errorProvider->SetError(this->dataGridInput, "В_таблице_есть_не_целые
            _числа");
330         break;
331     }
332 }
333 /*
334  * 1 -- в строке замены есть пустые ячейки
335  * 2 -- в строке замены есть не целые числа
336 */
337 //проверяем строку замены
338 try {
339     for (int i = 0; i < this->dataGridRow->ColumnCount; i++) {
340         bool resultParse;
341         //проверка на пустоту
342         if (!(String ^)this->dataGridRow->Rows[0]->Cells[i]->Value) {
343             throw 1;
344         }
345         //проверка на число
346         int current;
347         resultParse = Int32::TryParse(this->dataGridRow->Rows[0]->Cells[i]->Value->
            ToString(), current);
348         if (!resultParse) {
349             throw 2;
350         }
351     }
352 }
353 catch (int errorCode) {
354     switch (errorCode) {
355     case 1:
356         this->errorProvider->SetError(this->dataGridRow, "В_строке_есть_пустые_
            ячейки");
357         break;
358     case 2:
359         this->errorProvider->SetError(this->dataGridRow, "В_строке_есть_не_целые_

```

```

        числа");
360     break;
361 }
362 }
363 //заменяем строки
364 try {
365     //если есть ошибки в заполнении
366     if (this->errorProvider->GetError(this->dataGridInput)->ToString() != String::
        Empty
367         || this->errorProvider->GetError(this->dataGridRow)->ToString() != String::
        Empty) {
368         throw 0;
369     }
370     //добавляем столбцы в матрице вывода
371     for (int i = 0; i < this->dataGridInput->ColumnCount; i++) {
372         this->dataGridOutput->Columns->Add("ColumnOutput" + i, String::Empty);
373         this->dataGridOutput->Columns[i]->Width = 40;
374     }
375     //добавляем строки
376     this->dataGridOutput->Rows->Add(this->dataGridInput->RowCount - 1);
377
378     //заполняем ячейки
379     for (int i = 0; i < this->dataGridInput->RowCount - 1; i++) {
380         bool f = false;
381         for (int j = 0; j < this->dataGridInput->ColumnCount; j++) {
382             if (Int32::Parse(this->dataGridInput->Rows[i]->Cells[j]->Value->ToString())
                == min) {
383                 f = true;
384             }
385             //если есть минимальный
386             if (f) {
387                 //переписываем из строки для замены
388                 for (int j = 0; j < this->dataGridOutput->ColumnCount; j++) {
389                     this->dataGridOutput->Rows[i]->Cells[j]->Value = this->dataGridRow
                        ->Rows[0]->Cells[j]->Value;
390                 }
391             }
392             else {
393                 //переписываем из исходной строки
394                 for (int j = 0; j < this->dataGridOutput->ColumnCount; j++) {
395                     this->dataGridOutput->Rows[i]->Cells[j]->Value = this->dataGridInput

```

```

        ->Rows[i]->Cells[j]->Value;
396     }
397   }
398 }
399 }
400 }
401 catch (int errorCode) {
402 }
403 }
404 };
405 }

```