Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

УТВЕРЖДАЮ
Зав.кафедрой,
к.ф.-м.н.
_____ С. В. Миронов

ОТЧЕТ О ПРАКТИКЕ

студента 1 курса 111 группы факультета КНиИТ
Мирзоева Никиты Романовича

вид практики: учебная
кафедра: математической кибернетики и компьютерных наук
курс: 1
семестр: 2
продолжительность: 2 нед., с 29.06.2015 г. по 12.07.2015 г.

Руководитель практики от университета,

доцент, к. ф.-м. н _____ А. С. Иванова

Руководитель практики от организации (учреждения, предприятия),

доцент, к. ф.-м. н. _____ А. С. Иванова

Тема практики:«Разработка приложения для проверки заданий по методу резолюций и по методу Вонга»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Целью практики является создание приложения для проверки знаний студентов в рамках дисциплины «Математические основы искусственного интеллекта», которое улучшит учебный процесс. При создании приложения был изучен и использован API Windows Forms на языке C++ в среде Microsoft VisualStudio [1].

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 Создание приложения, используемого преподавателем

ЗАДАНИЕ. Реализовать приложение для создания, редактирования, распределения между студентами и проверки заданий по методу резолюций и по методу Вонга.

Создано главное окно приложения, содержащее четыре элемента `Button`, три элемента `OpenFileDialog`, три элемента `ToolStripMenuItem` и один элемент `MenuStrip`. Вид окна представлен на рисунке 1.

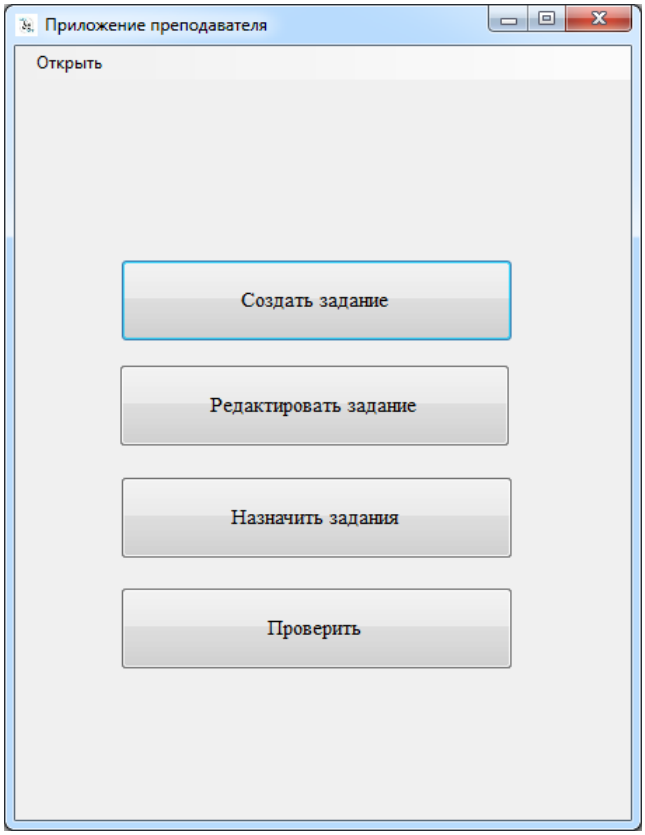

Рисунок 1 – Главное окно приложения преподавателя

При выборе пункта меню «Открыть файл с заданиями» производится сохранение имени файла при помощи элемента OpenFileDialog (см. рисунок 2):

```
1  private: System::Void открытьФайлСЗаданиямиToolStripMenuItem_Click(System::Object^
       sender, System::EventArgs^ e)
2  {
3      if ( this->openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::
           OK)
4      {
5      }
6      for (int i = 0; i < openFileDialog1->FileName->Length; i++)
7      {
```

```
8           fileEditor [i] = openFileDialog1->FileName[i]; // сохранение имени файла с
                заданиями
9       }
10      fileEditor [openFileDialog1->FileName->Length] = '\0';
11      editor_open = true; // открыт файл с заданиями
12  }
```

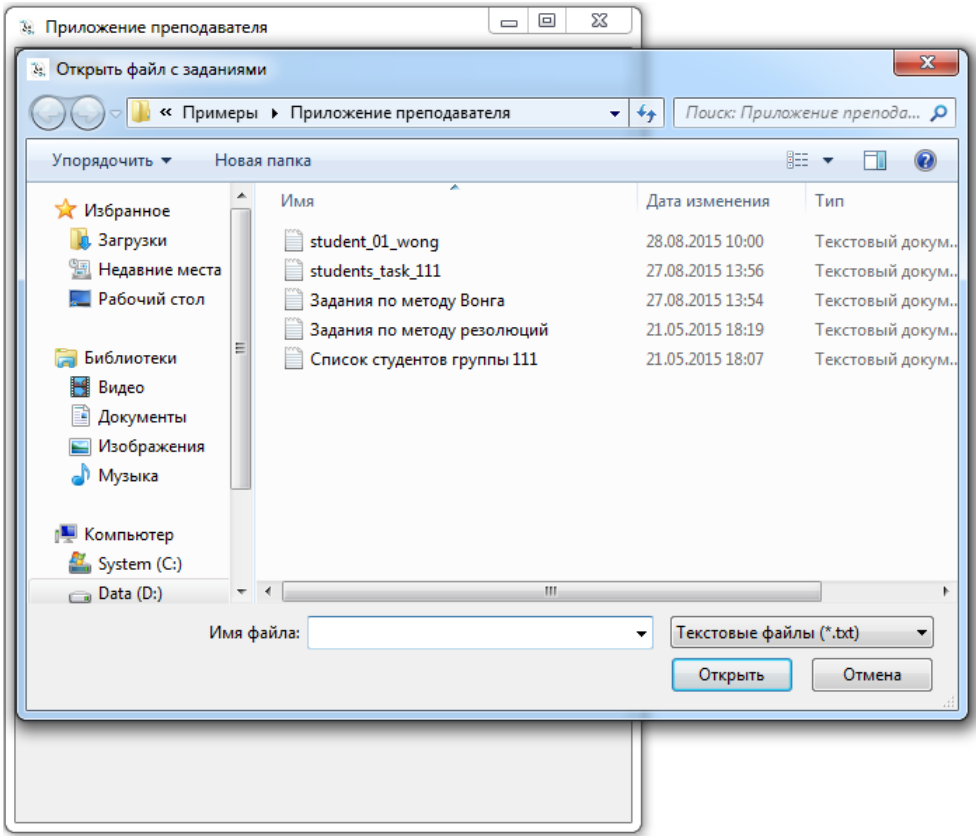

Рисунок 2 – Главное окно приложения преподавателя: сообщение сигнализирует, что файл с заданиями не выбран

Была написана функция подсчёта количества заданий в файле с заданиями:

```
1   // подсчёт количества заданий
2   public: static void kolvo_zad()
3   {
4     kol_zad = 0;
5     ifstream in(Form1:: fileEditor );
6     while (in.peek() != EOF)
7     {
8       string s;
9       getline (in, s);
10      kol_zad++;
11    }
```

```
        in . close () ;
    }
```

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

```
// кнопка "Создать задание"
System::Void Form1::button1_Click(System::Object^  sender, System::EventArgs^  e)
{
    if (! editor_open)
    {
        MessageBox::Show( "Откройте файл для добавления заданий.", "Внимание!");
        return;
    }
    Form1::kolvo_zad();  // подсчёт количества заданий
    var = kol_zad + 1;

    // перейти к окну "Редактор"
    editor ^ed = gcnew editor( this );
    ed->Show();
    this->Hide();
}
```

Если при нажатии кнопки «Создать задание» не выбран файл с заданиями, то возникает сообщение об ошибке (см. рисунок 3).

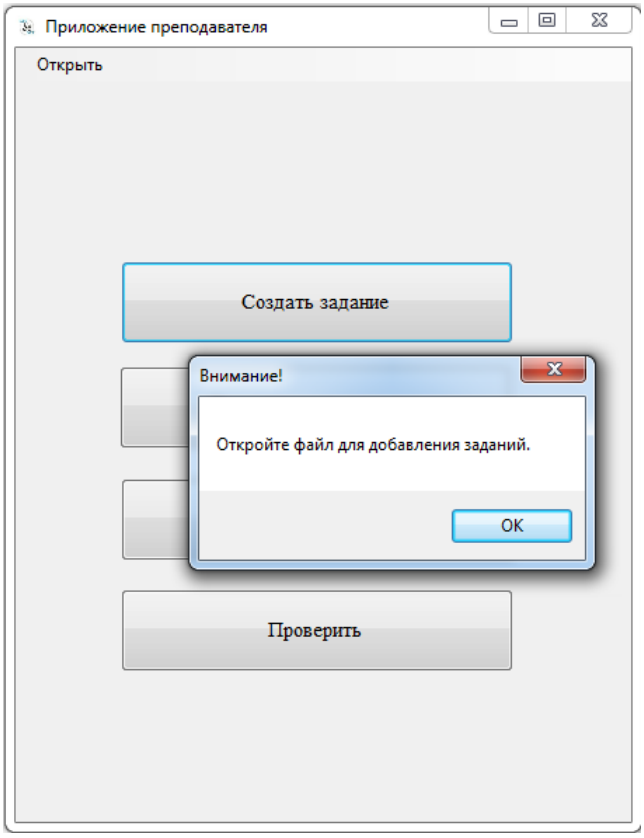

Рисунок 3 – Главное окно приложения преподавателя: сообщение сигнализирует, что файл с заданиями не выбран

Создано окно **Редактор**, содержащее четыре элемента `Label`, три элемента `ListBox`, шесть элементов `Button` и один элемент `TextBox`. Вид окна представлен на рисунке 4.

При создании дизъюнктов и конъюнктов нужно выбирать операции из списка операций (см. рисунок 5):

```
1  // выбор операций
2  private: System::Void inform_SelectedIndexChanged(System::Object^ sender, System::
       EventArgs^ e)
3  {
4    kate = what_kate(); // определение текущей операции
5    oper = what_oper(); // определение последнего символа
6    switch (inform->SelectedIndex)
7    {
8      case 0: // конъюнкция
9      {
10       if (kate == 1)
11       {
12         MessageBox::Show(this, "Конъюнкция и дизъюнкция не могут быть вместе в
             одном выражении.", "Внимание!");
```

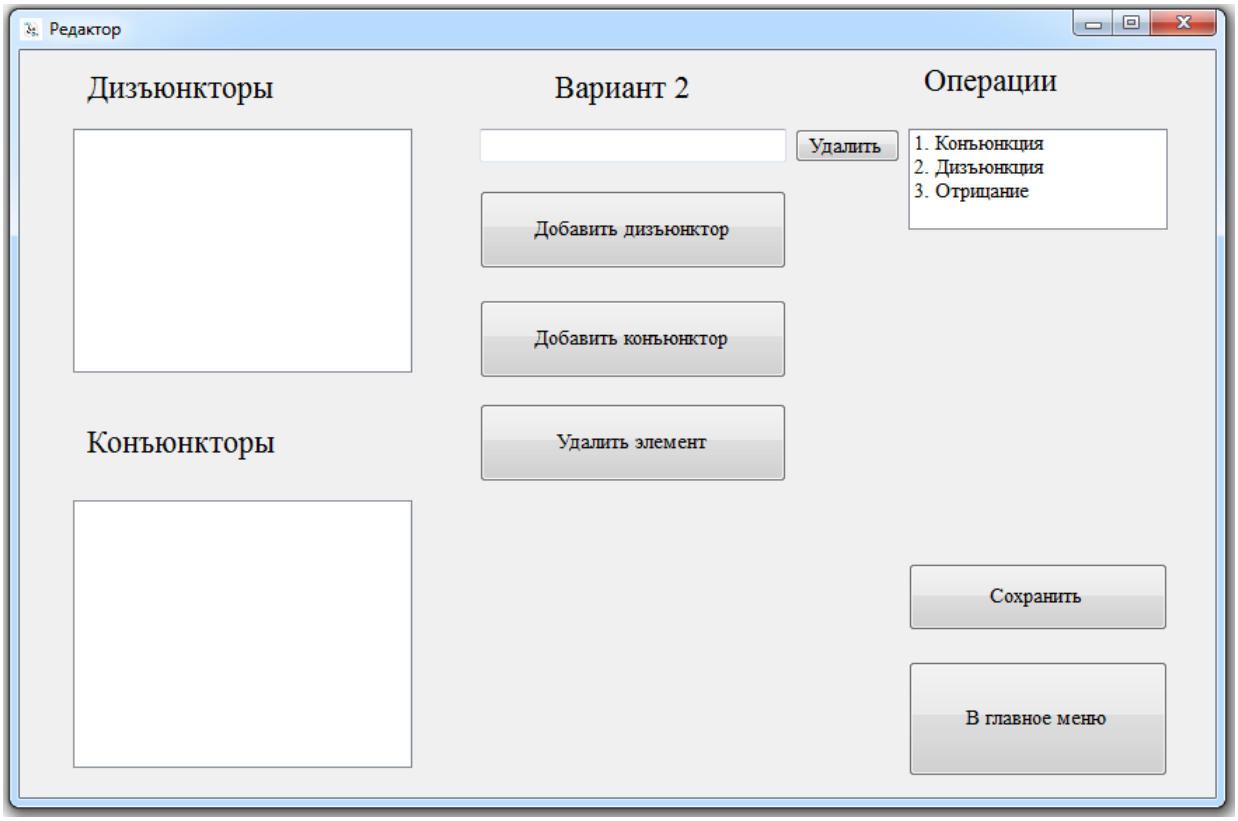

Рисунок 4 – Окно **Редактор** приложения преподавателя

```
13            break;
14          }
15          if (oper == 2 || oper == 3)
16          {
17            MessageBox::Show(this,"Две операции не могут идти подряд.", "Внимание!");
18            break;
19          }
20          if (oper < 0)
21          {
22            MessageBox::Show(this,"Выражение не может начинаться со знака конъюнкции.",
                  "Внимание!");
23            break;
24          }
25          oper = 2;
26          kate = 2;
27          textBox1->Text = textBox1->Text + " & ";
28          break;
29        }
30        case 1: //дизъюнкция
31        {
32          if (kate == 2)
```

```
33          {
34            MessageBox::Show(this, "Конъюнкция и дизъюнкция не могут быть вместе в
                  одном выражении.", "Внимание!");
35            break;
36          }
37          if (oper == 1 || oper == 3)
38          {
39            MessageBox::Show(this,"Две операции не могут идти подряд.", "Внимание!");
40            break;
41          }
42          if (oper < 0)
43          {
44            MessageBox::Show(this,"Выражение не может начинаться со знака дизъюнкции.", "
                  Внимание!");
45            break;
46          }
47          oper = 1;
48          kate = 1;
49          textBox1->Text = textBox1->Text + " v ";
50          break;
51        }
52        case 2: // отрицание
53        {
54          if (oper == 3)
55          {
56            MessageBox::Show(this,"Две операции не могут идти подряд.", "Внимание!");
57            break;
58          }
59          else
60          {
61            if (oper == 0)
62            {
63              MessageBox::Show(this,"Недопустимое действие!", "Внимание!");
64              break;
65            }
66          }
67          textBox1->Text = textBox1->Text + "- ";
68          oper = 3;
69          break;
70        }
71      }
```

```
72
73      inform->SelectedIndex = -1;
74      textBox1->Focus();
75      textBox1->Select(textBox1->Text->Length, 0);
76
77      is_changes_ed = true;  // произошли изменения
78      return;
79  }
```

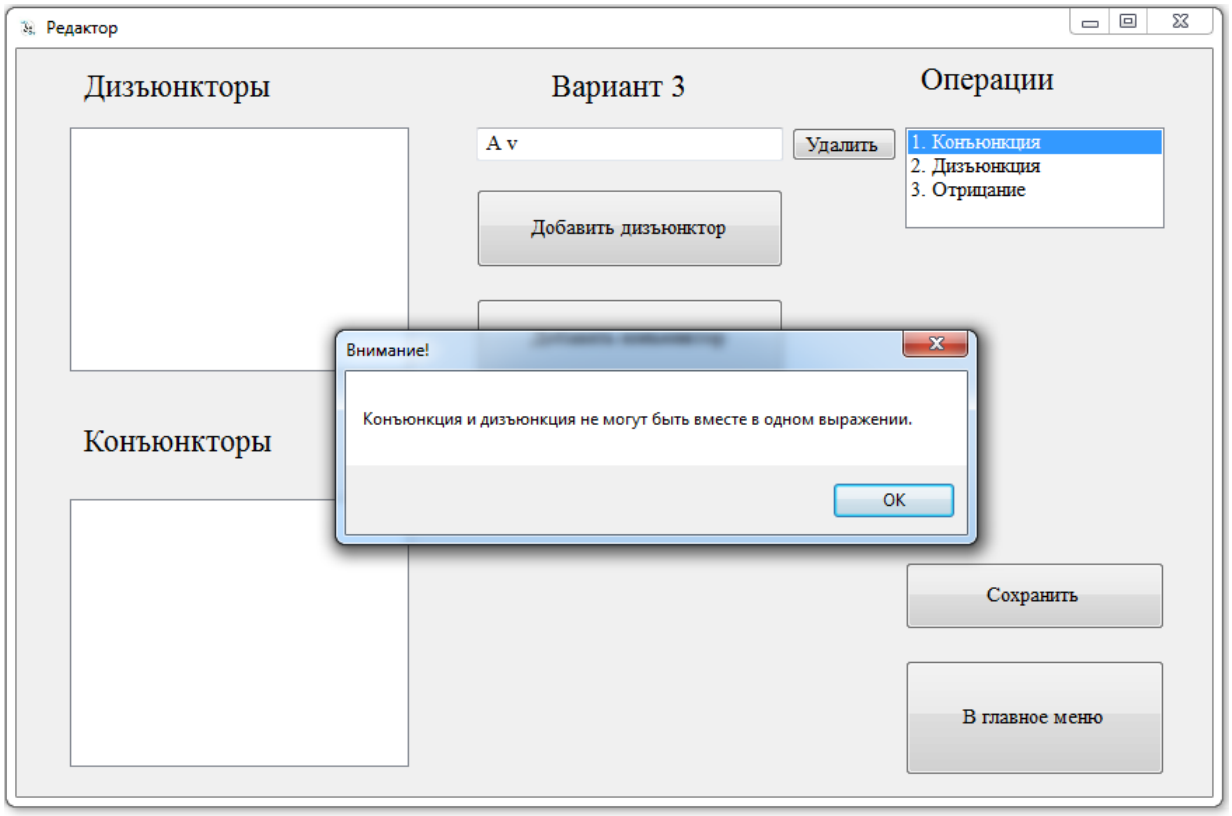

Рисунок 5 – Окно **Редактор** приложения преподавателя: выбор операции

Приложение не даёт ошибиться при выборе списка, в который нужно добавить выражение (см. рисунок 6):

```
1   // кнопка "Добавить дизъюнктор"
2   private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e)
3   {
4      String  ^s;
5      s = textBox1->Text->ToString(); // сохранение строки из поля редактирования текста
6
7      kate = what_kate();  // определение текущей операции
8      oper = what_oper();  // определение последнего символа
9
10     if (oper > 0)
```

```
11      {
12        MessageBox::Show(this,"Выражение не может оканчиваться знаком логической
              операции.", "Внимание!");
13        textBox1->Focus();
14        textBox1->Select(textBox1->Text->Length, 0);
15        return;
16      }
17      if  (kate  == 2)
18      {
19        MessageBox::Show(this,"Выражение содержит конъюнкцию.", "Внимание!");
20        textBox1->Focus();
21        textBox1->Select(textBox1->Text->Length, 0);
22        return;
23      }
24      if  (  (textBox1->Text != "")  && (s[0] !=  '  ')  )
25      {
26        if  (listBox1->SelectedIndex > -1)
27        {
28          listBox1->Items[listBox1->SelectedIndex] = textBox1->Text;
29          listBox1->SelectedIndex = -1;
30          textBox1->Text = "";
31        }
32        else
33        {
34          listBox1->Items->Add(textBox1->Text);
35          textBox1->Text = "";
36          n1++;
37        }
38      }
39      oper  = -1;
40      kate  = 0;
41      listBox1->SelectedIndex = -1;
42      listBox2->SelectedIndex = -1;
43      is_changes_ed = true;  // произошли изменения
44  }
```

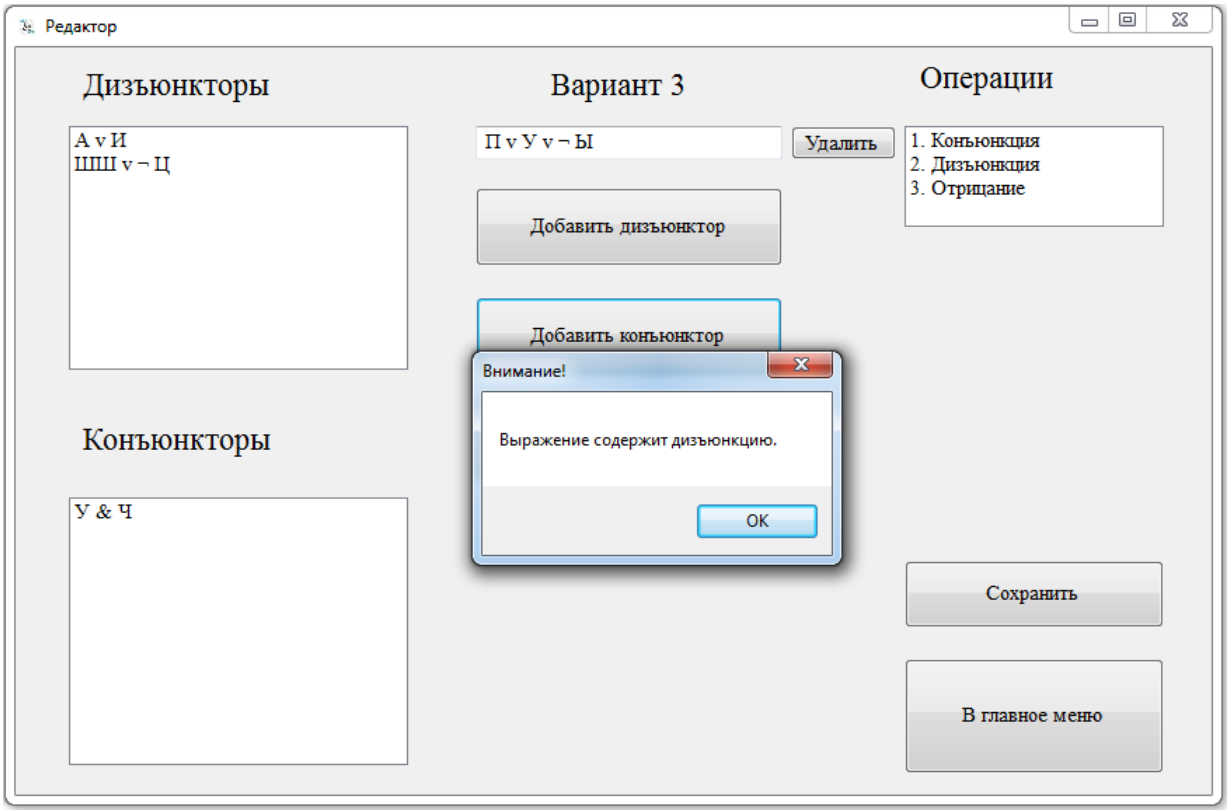

Рисунок 6 – Окно **Редактор** приложения преподавателя: добавление выражения в нужный список

Создано окно **Редактор заданий**, содержащее один элемент `Label`, один элемент `ListBox` и четыре элемента `Button`. Вид окна представлен на рисунке 7.

При нажатии кнопки «Редактировать» происходит переход к окну **Редактор**:

```
1   // кнопка "Редактировать"
2   private: System::Void button2_Click(System::Object^  sender, System::EventArgs^  e)
3   {
4     int x = listBox1->SelectedIndex;
5     Form1::var = x + 1;
6     Form1::now_edit = true;
7
8     if (Form1::var == 0)
9     {
10      MessageBox::Show(this,"Выберите задание.", "Внимание!");
11      return;
12    }
13
14    if ( is_changes_sel )
15    {
```

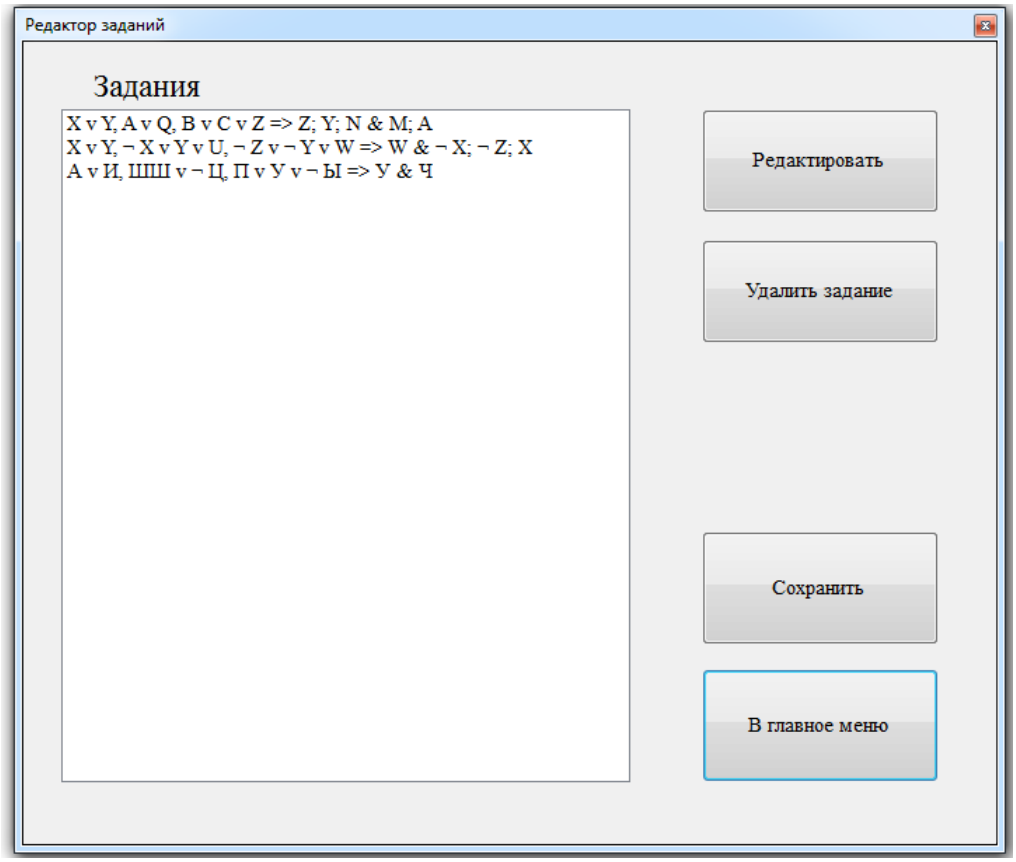

Рисунок 7 – Окно **Редактор заданий** приложения преподавателя

```
        MessageBoxButtons buttons = MessageBoxButtons::YesNo;
        System::Windows::Forms::DialogResult  result ;

         result  = MessageBox::Show(this, "Сохранить изменения?", "Внимание!", buttons);

        if  ( result  == System::Windows::Forms::DialogResult::Yes)
        {
          save();  // функция сохранения
        }
        else
        {
           if_not_save ();  // функция, если не выбрано сохранение
        }
    }
```

Реализация функции сохранения:

```
// функция сохранения
private:  void  save()
{
   ofstream  out(Form1:: fileEditor );
   for  (int  i  =  0;  i  <  Form1::kol_zad;  i++)
```

```
6       {
7           string s;
8           using namespace Runtime::InteropServices;
9           const char* chars = (const char*)(Marshal::StringToHGlobalAnsi(listBox1->Items[i
                ]->ToString())).ToPointer();
10          s = chars;
11          Marshal::FreeHGlobal(IntPtr((void*)chars));
12          out << i+1 << ". " << s << endl;
13      }
14      out.close();
15      is_changes_sel = false; // отсутствие изменений
16  }
```

Создано окно **Распределение заданий**, содержащее четыре элемента `Label`, три элемента `ListBox`, один элемент `ComboBox`, один элемент `TextBox` и четыре элемента `Button`. Вид окна представлен на рисунке 8.

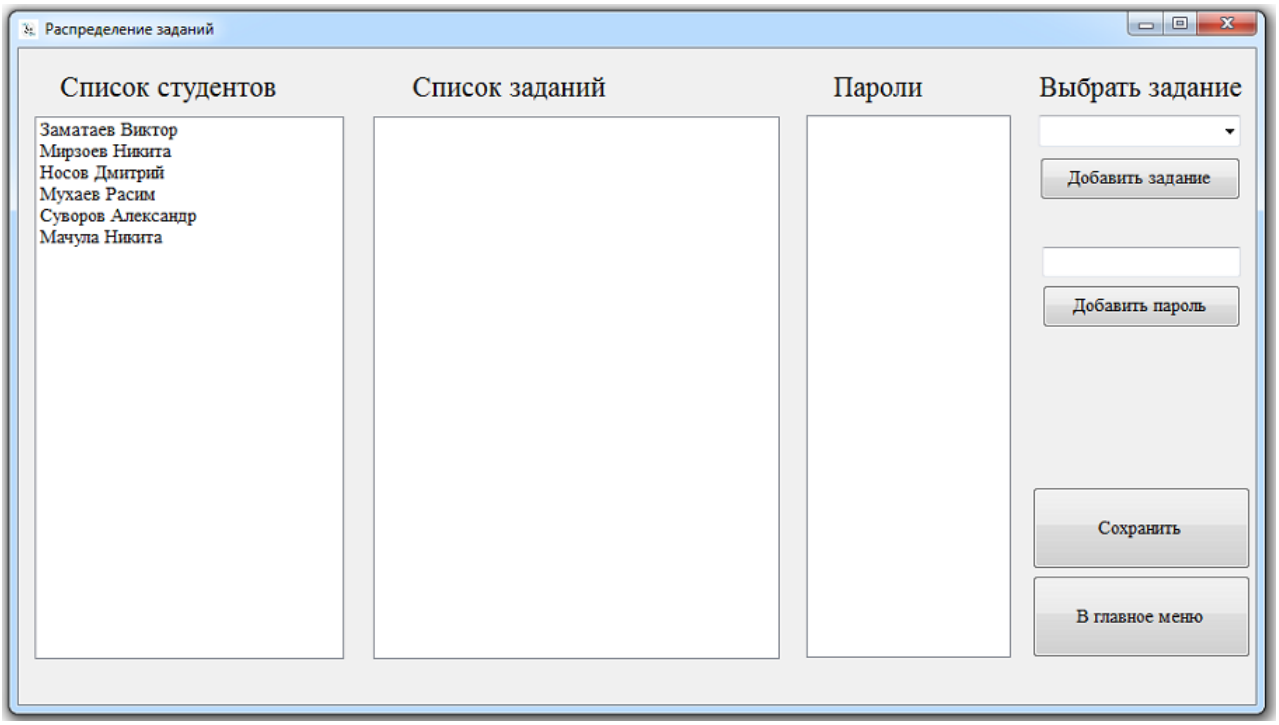

Рисунок 8 – Окно **Распределение заданий** приложения преподавателя

После нажатия кнопки «Добавить задание» выполняется следующий фрагмент кода:

```
// кнопка "Добавить задание"
private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e)
{
  if (listBox1->SelectedIndex == -1)
    MessageBox::Show(this,"Выберите студента.", "Внимание!");
  else
  {
    if (comboBox1->SelectedIndex == -1)
      MessageBox::Show(this,"Выберите задание.", "Внимание!");
    else
    {
      listBox2->Items[listBox1->SelectedIndex] = comboBox1->Text;
      listBox1->SelectedIndex = -1;
      listBox2->SelectedIndex = -1;
      listBox3->SelectedIndex = -1;
      comboBox1->Text = "";
      is_changes_ass = true; // произошли изменения
```

```
18              }
19          }
20      }
```

Создано окно **Проверка задания**, содержащее четыре элемента `Label`, три элемента `ListBox`, один элемент `TextBox` и один элемент `Button`. Вид окна представлен на рисунке 9.

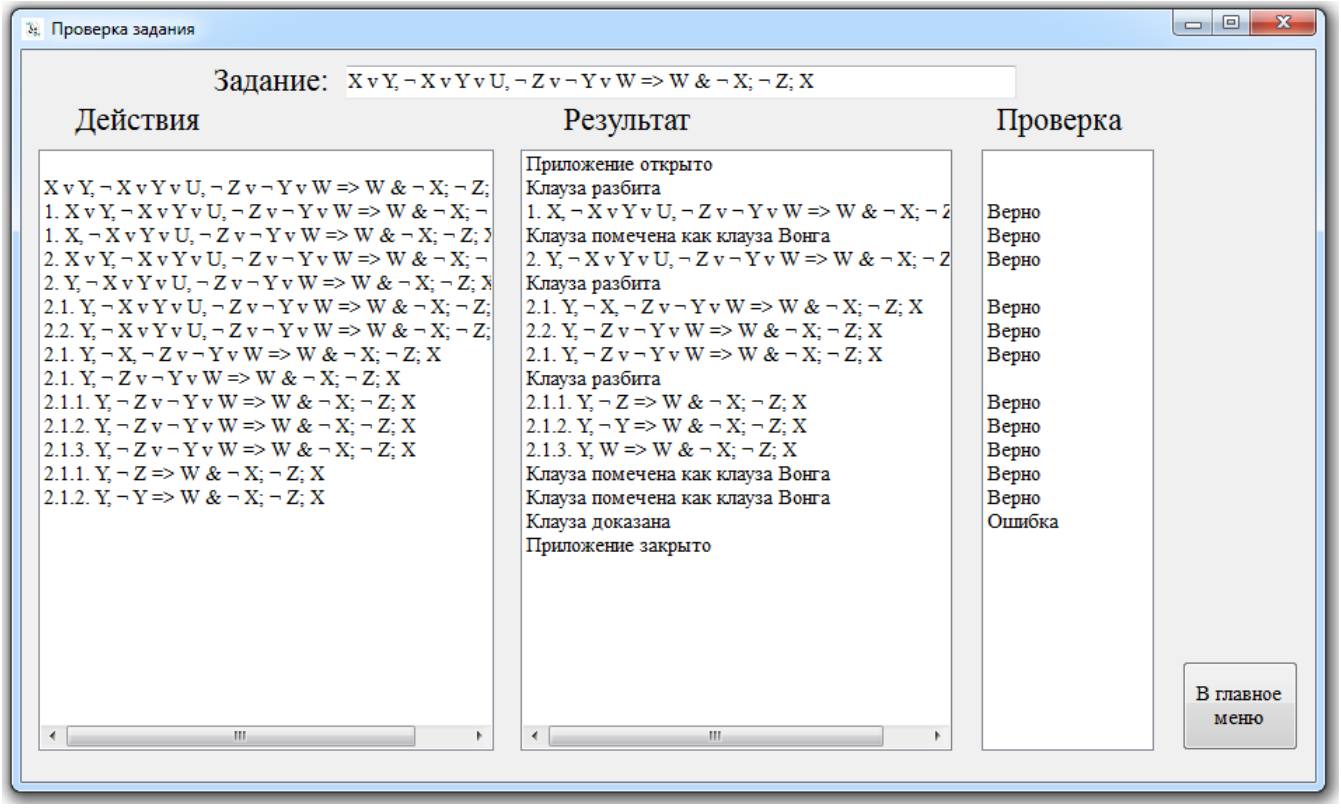

Рисунок 9 – Окно **Проверка задания** приложения преподавателя

Функция проверки на правильность разбиения клаузы:

```
1   // проверка разбиения клаузы
2   private: bool check_crash(string s1, string s2, string &new_str)
3   {
4     int num = (int)s1[s1.find(" ") - 2] - 48;
5     int x = s1.find("v");
6     if (x == string::npos)
7       x = s1.find("&");
8     int i = 0, j = 0;
9     for (i = x; i >= 0; i--)
10    {
11      if (i == 0 || s1[i] == '.' || s1[i] == ',' || s1[i] == ';' || s1[i] == '>')
12        break;
13    }
14    if (i > 0)
15      i += 2;
16    for (j = x + 1; j < s1.size(); j++)
17    {
```

```
18          if (j + 1 == s1.size() || s1[j] == ',' || s1[j] == ';' || s1[j + 1] == '=')
19              break;
20      }
21
22      int k = 0;
23      string st = "";
24      x = i + 1;
25      int t = i;
26      while (x < j)
27      {
28          if (x + 1 == j || s1[x + 1] == 'v' || s1[x + 1] == '&')
29          {
30              k++;
31              if (k == num)
32              {
33                  if (x + 1 == j)
34                      x++;
35                  st = s1.substr(t, x - t);
36                  break;
37              }
38              t = x + 3;
39              x = t - 1;
40          }
41          x++;
42      }
```

Полный код программы находится на CD-диске в папке `Projects of applications/App for CSiIT` (Приложение А).

2 Создание приложения для решения задания по методу резолюций

ЗАДАНИЕ. Реализовать приложение для решения заданий по методу резолюций, используемое студентом.

Создано главное окно приложения, содержащее два элемента `Label`, один элемент `ComboBox`, один элемент `TextBox`, один элемент `Button`, один элемент `MenuStrip`, один элемент `ToolStripMenuItem`, и один элемент `OpenFileDialog`. Вид окна представлен на рисунке 10.

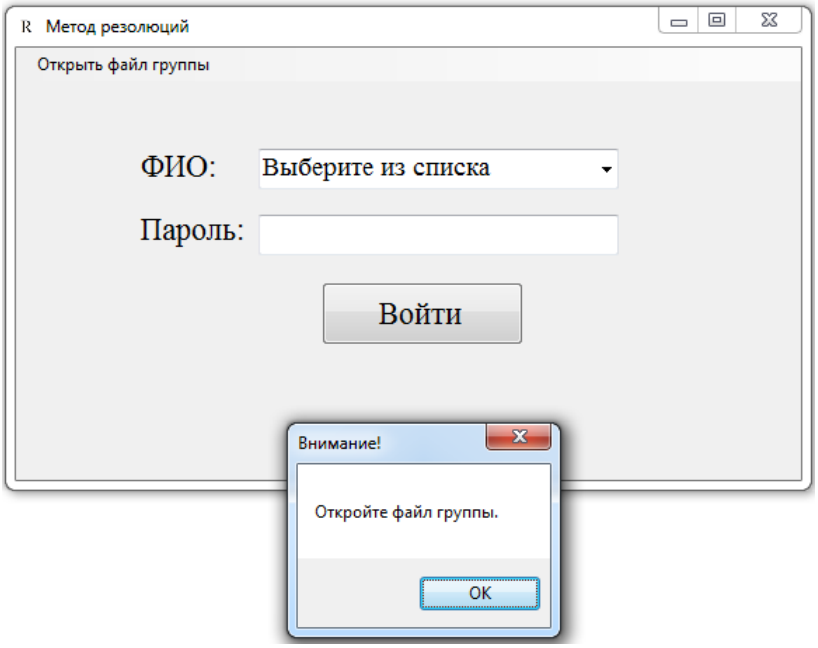

Рисунок 10 – Главное окно приложения по методу резолюций

На нажатие кнопки «Войти» установлено выполнение следующего кода:

```
1  // кнопка "войти"
2  System::Void Form1::button1_Click(System::Object^  sender, System::EventArgs^  e)
3  {
4    if (! is_init (textBox1->Text)) // проверка правильности пароля
5    {
6      if (textBox1->Text->Length == 0)
7        MessageBox::Show(this,"Введите пароль.", "Внимание!");
8      else
9        MessageBox::Show(this,"Неправильно введён пароль.", "Внимание!");
10     return;
11   }
12
13   // перейти к окну "Метод резолюций"
```

```
14      resolution ^res = gcnew resolution ( this ) ;
15      res ->Show();
16      this ->Hide();
17  }
```

Создано окно **Метод резолюций**, содержащее три элемента `Label`, два элемента `ListBox`, один элемент `TextBox` и пять элементов `Button`. Вид окна представлен на рисунке 11.

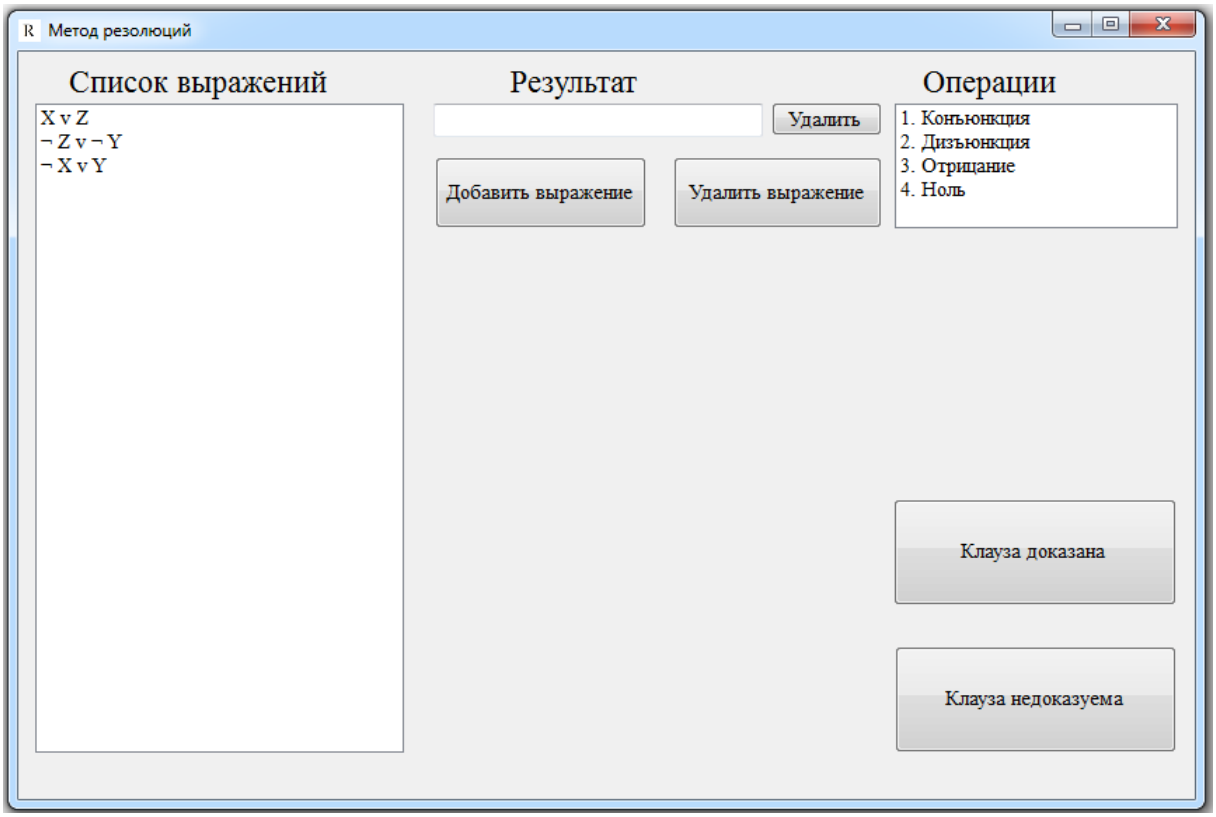

Рисунок 11 – Окно **Метод резолюций** приложения по методу резолюций

Функция обработки нажатия кнопки «Добавить выражение»:

```
// кнопка "Добавить выражение"
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e)
{
  kate = what_kate(); // определение текущей операции
  oper = what_oper();  // определение последнего символа
  if (listBox1->SelectedIndices->Count != 2)
  {
    MessageBox::Show(this,"Выберите ровно два выражения.", "Внимание!");
    textBox1->Focus();
    textBox1->Select(textBox1->Text->Length, 0);
    return;
  }
  if (oper > 0 && oper < 4)
  {
    MessageBox::Show(this,"Выражение не может оканчиваться знаком логической
        операции.", "Внимание!");
```

```
16          textBox1->Focus();
17          textBox1->Select(textBox1->Text->Length, 0);
18          return;
19      }
20      String ^ s;
21      s = textBox1->Text->ToString();
22      if ( (textBox1->Text != "") && (s[0] != ' ') )
23      {
24          listBox1->Items->Add(textBox1->Text);
25          textBox1->Text = "";
26
27          string fileStudent;
28          create_fileStudent ( fileStudent ); // создание файла студента
29
30          ofstream out( fileStudent , ios::app);
31          string s0 = convert_to_string (listBox1->SelectedItems[0]->ToString());
32          string s1 = convert_to_string (listBox1->SelectedItems[1]->ToString());
33          string s2 = convert_to_string (s);
34
35          out << s0 << ", " << s1 << " !=> " << s2 << endl;
36          out.close();
37      }
38      oper = -1;
39      kate = 0;
40      listBox1->ClearSelected();
41  }
```

Полный код программы находится на CD-диске в папке `Projects of applications/Resolution methode` (Приложение А).

3 Создание приложения для решения задания по методу Вонга

ЗАДАНИЕ. Реализовать приложение для решения заданий по методу Вонга, используемое студентом.

Создано главное окно приложения идентичное главному окну приложения для решения задачи по методу резолюций. Вид окна представлен на рисунке 12.

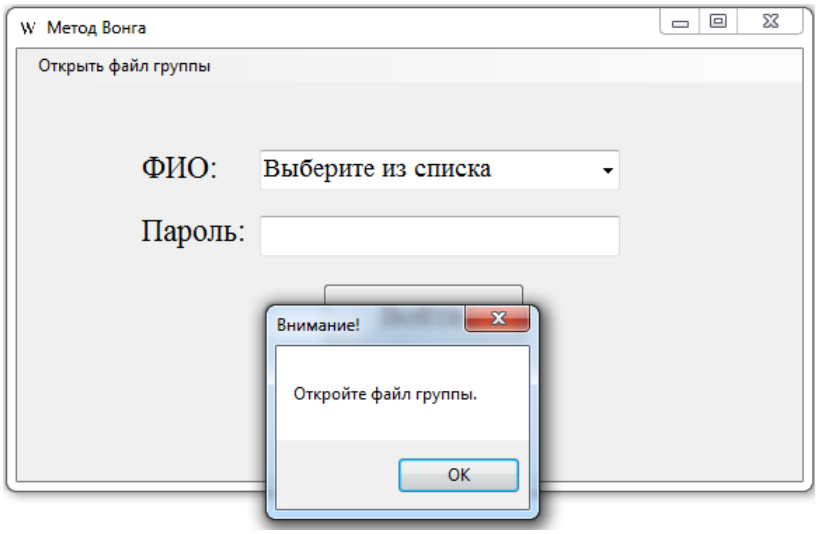

Рисунок 12 – Главное окно приложения по методу Вонга

Создано окно **Метод Вонга**, содержащее один элемент `TextBox`, один элемент `CheckedListBox` и пять элементов `Button`. Вид окна представлен на рисунке 13.

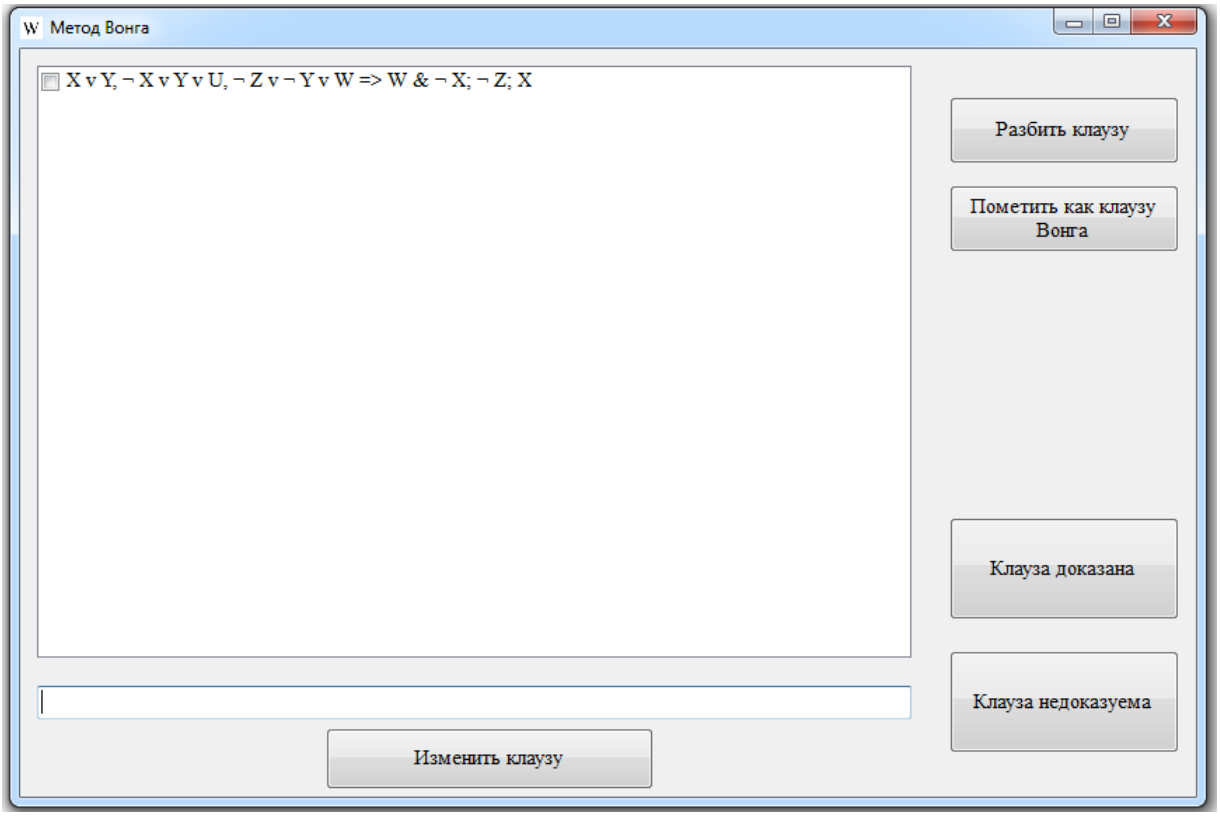

Рисунок 13 – Окно **Метод Вонга** приложения по методу Вонга

При нажатии на кнопку «Разбить клаузу» воспроизводится следующий код:

```
1   // кнопка "Разбить клаузу"
2   private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e)
3   {
4     if (checkedListBox1->SelectedIndex == -1)
5       MessageBox::Show(this,"Выберите клаузу.", "Внимание!");
6     else
7     {
8       String ^s;
9       s = checkedListBox1->SelectedItem->ToString(); // создание файла студента
10      if (s->IndexOf('v') == -1 && s->IndexOf('&') == -1)
11      {
12        MessageBox::Show(this,"Клаузу нельзя разбить.", "Внимание!");
13      }
14      else
15      {
```

```
16          string  s1 =  convert_to_string (s);
17
18          string   fileStudent ;
19           create_fileStudent  ( fileStudent );  // создание файла студента
20
21          // протоколирование действия студента
22          ofstream  out( fileStudent ,  ios :: app);
23          out  << s1 << " !=> !Crash" << endl;
24          out . close () ;
25          changes[checkedListBox1->SelectedIndex] = true;
26
27          int  x = s1 . find ("v");
28          if  (x == string :: npos)
29            x = s1 . find ("&");
30
31          int  kol = 1;  // количество разбиений
32          x++;
33          while  (x + 1 != s1 . size ()  && s1[x] != ',' && s1[x] != ';' && s1[x + 1] != '=')
34          {
35             if  (s[x] == 'v'  ||  s[x] == '&')
36               kol++;
37            x++;
38          }
39          kol++;
40
41          int  kol_points  = 0;  // количество точек
42          int  i = 0;
43          while  (s[i] != ' ')
44          {
45             if  (s[i] == '.')
46               kol_points++;
47            i++;
48          }
49
50          int  tek_n = 1;
51          String  ^s2 = "",  ^s3 = "";
52          if  ( kol_points  > 0)
53          {
54            s2 = s->Substring(0,  2 * kol_points );
55            s3 = s->Substring(2 * kol_points  + 1,  s->Length - (2 * kol_points + 1));
56          }
```

```
57            else
58            {
59              s3 = s;
60            }
61
62            while (tek_n <= kol)
63            {
64              checkedListBox1->Items->Add(s2 + tek_n + ". " + s3);
65              changes[checkedListBox1->Items->Count - 1] = false;
66              tek_n++;
67            }
68          }
69        }
70
71        checkedListBox1->SelectedIndex = -1;
72        textBox1->Text = "";
73    }
```

Полный код программы находится на CD-диске в папке `Projects of applications/Methode Wong` (Приложение А).

ЗАКЛЮЧЕНИЕ

В ходе практики была создана проверочная система, готовая к использованию в учебных целях. Эта система состоит из нескольких приложений, написанных на языке C++ в среде Microsoft Visual Studio. На практике разработаны приложения, содержащие такие элементы интерфейса как `Button`, `OpenFileDialog`, `ToolStripMenuItem`, `MenuStrip`, `Label`, `ListBox`, `TextBox`, `ComboBox`, `CheckedListBox`.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Создание приложения Windows Forms с помощью .NET Framework (C++) [Электронный ресурс]. — URL: http://msdn.microsoft.com/ru-ru/library/vstudio/ms235634(v=vs.100).aspx (Дата обращения 12.07.2015). Загл. с экр. Яз. рус.

ПРИЛОЖЕНИЕ А

CD-диск с отчетом о выполненной работе

На приложенном диске можно ознакомиться со следующими файлами:

Папка `LaTeX folder` — LaTeX- вариант отчета о практике;

Папка `Projects of applications/App for CSiIT` — проект приложения преподавателя;

Папка `Projects of applications/Methode Wong` — проект приложения по методу Вонга;

Папка `Projects of applications/Resolution method` — проект приложения по методу резолюций;

Папка `Examples` — примеры работы с программами;

`App for CSiIT.exe` — приложение преподавателя;

`Methode Wong.exe` — приложение по методу Вонга;

`Resolution method.exe` — приложение по методу резолюций;

`Отчёт по практике.pdf` — отчет по практике.