МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт Информационных Технологий

Кафедра МПО ЭВМ

Дисциплина «Теория автоматов и формальных языков»

Лабораторная работа №9-10

«Обработка ошибок на этапе лексического анализа. Построение КС-грамматики»

Выполнил:

студент группы 1ПИб-02-3оп-22

Маркелов Сергей Александрович

Проверил:

доцент, к.т.н.

Ганичева Оксана Георгиевна

Череповец, 2024 год

Задания

**Часть 1:** Написать часть лексического анализатора/функцию по обработке лексических ошибок. Принцип обработки ошибки: обнаружение ошибки, указание номера строки с ошибкой, идентификация ошибки

Отчет по работе включает код программы, описание типов ошибок, скриншоты с типичными ошибками

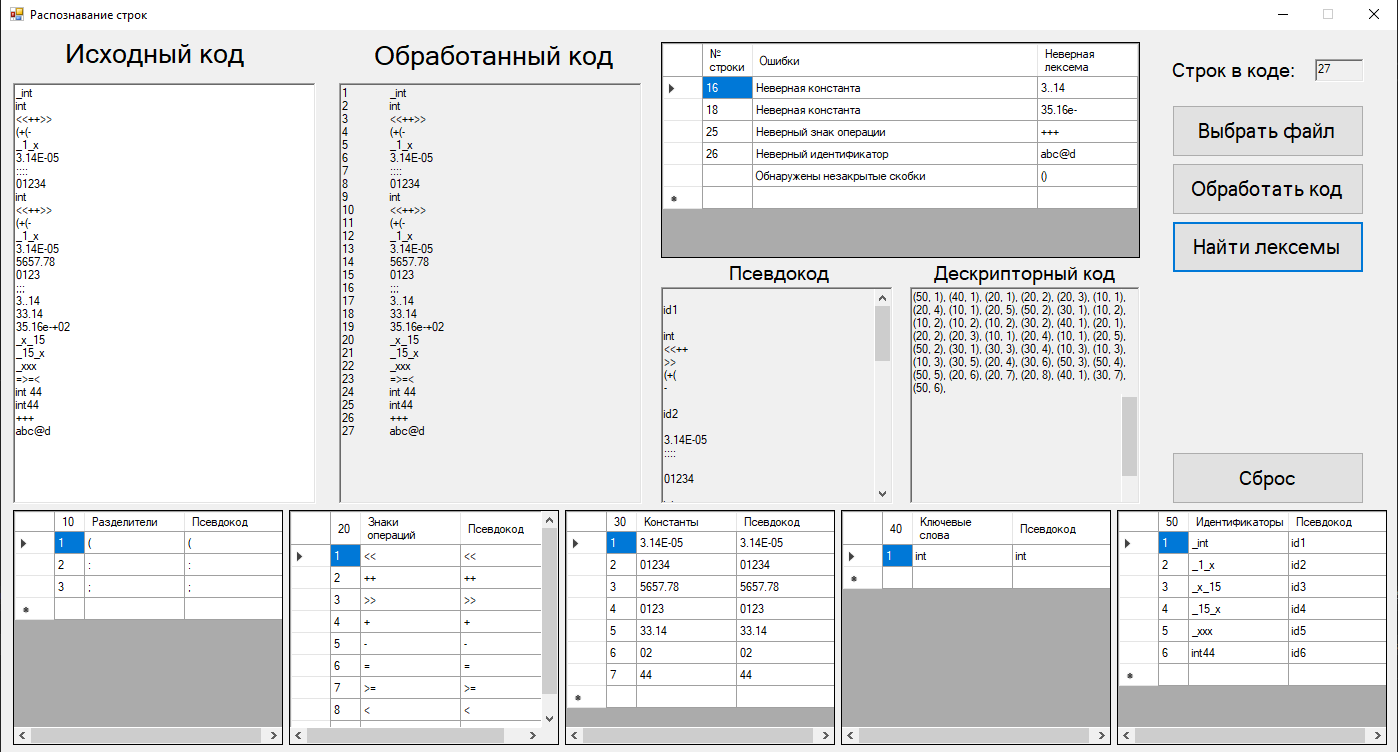
**Часть 2:** Построить КС-грамматику для заданного в курсовой работе варианта задания. Для заданного программного кода построить порождение и дерево вывода.

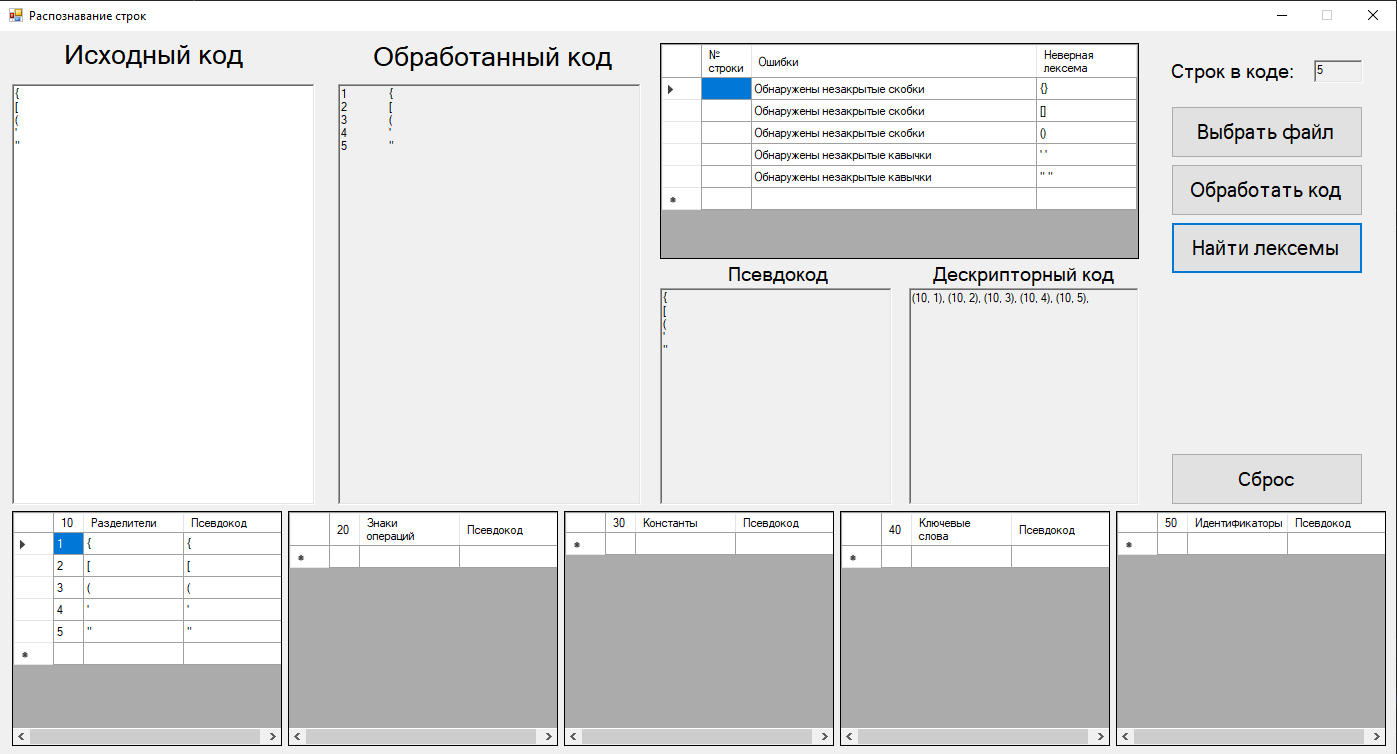
|  |  |
| --- | --- |
| Вариант | Задание |
| 14 | Оператор switch и if языка С++ |

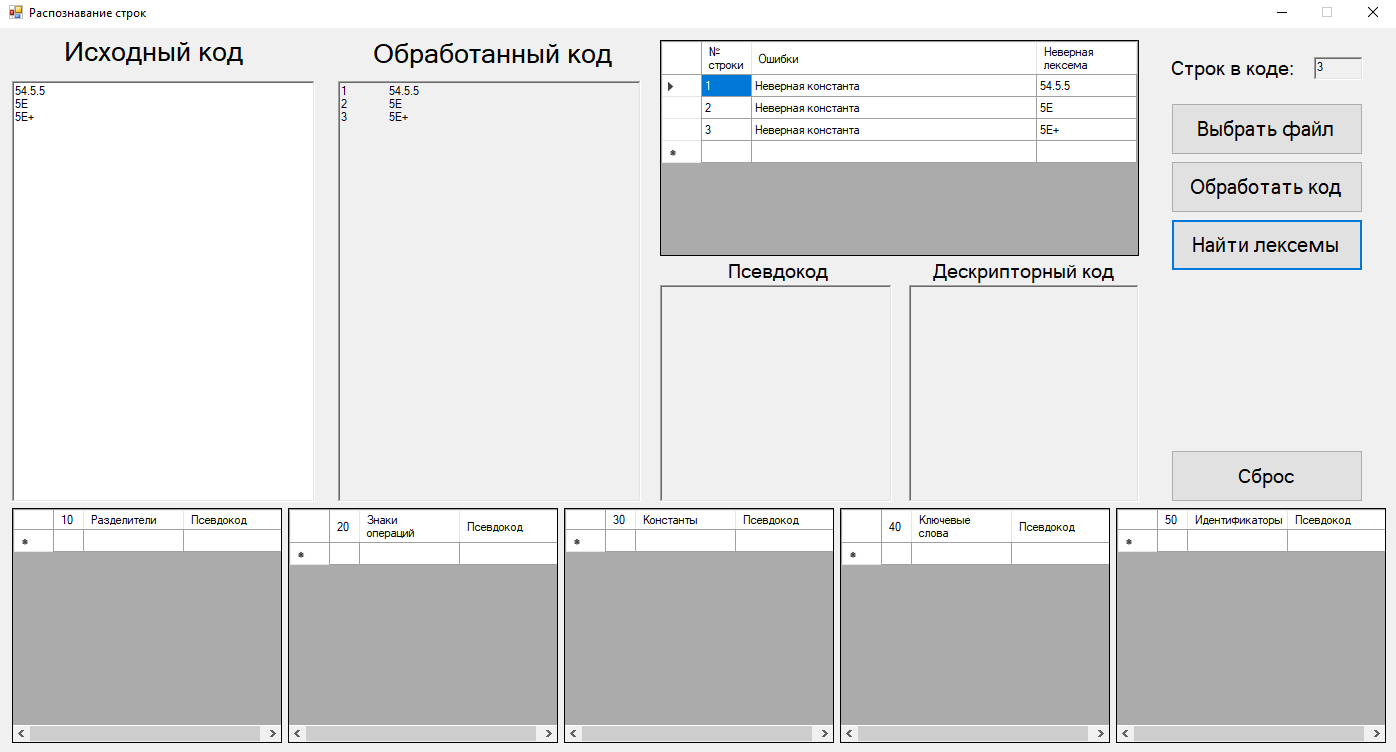
1. Типы ошибок

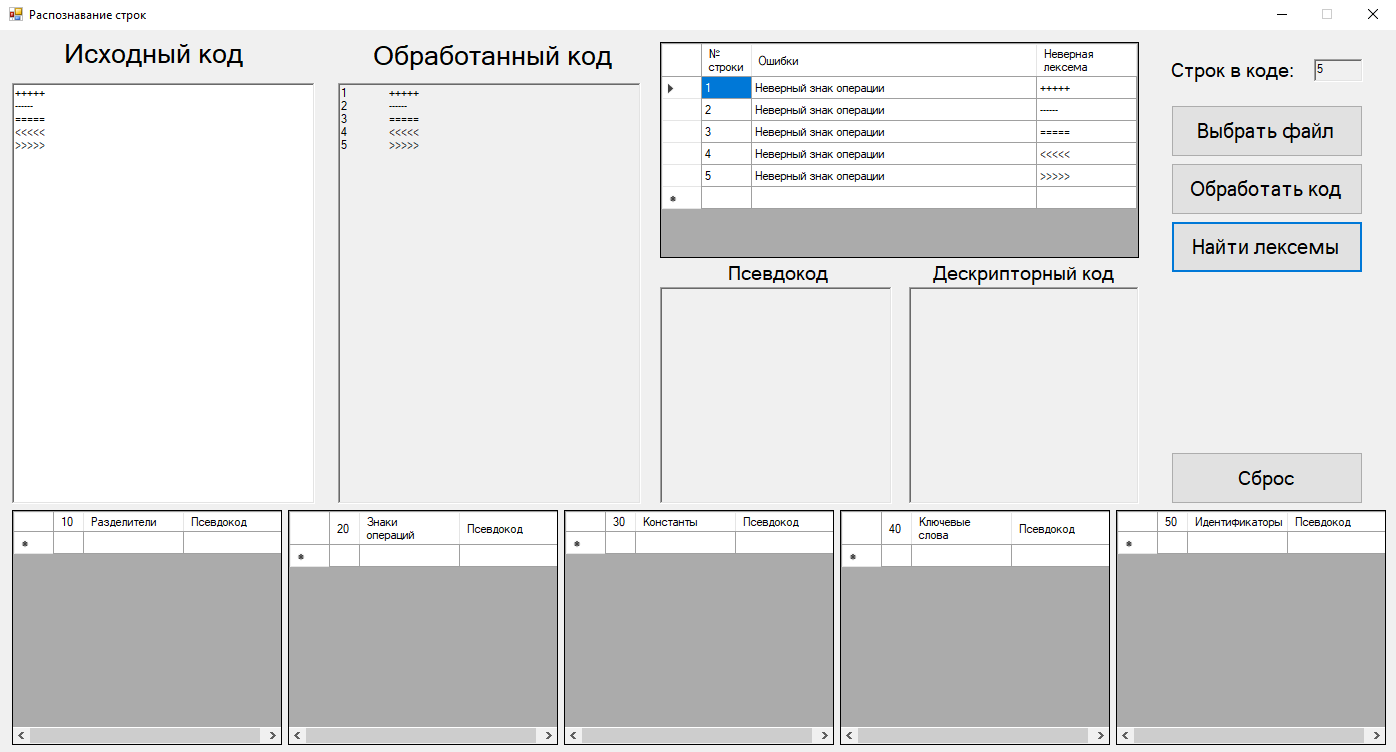
* Неверный знак операции (++++++, =====)
* Неверная константа (1.4.5, 5.3E+-02, 5.)
* Неверное ключевое слово (whi@le, i^f)
* Неверный идентификатор (1abc, abc@d)
* Неверный символ (@, №, $, ^, ~, `)
* Незакрытый многострочный комментарий (/\*…)

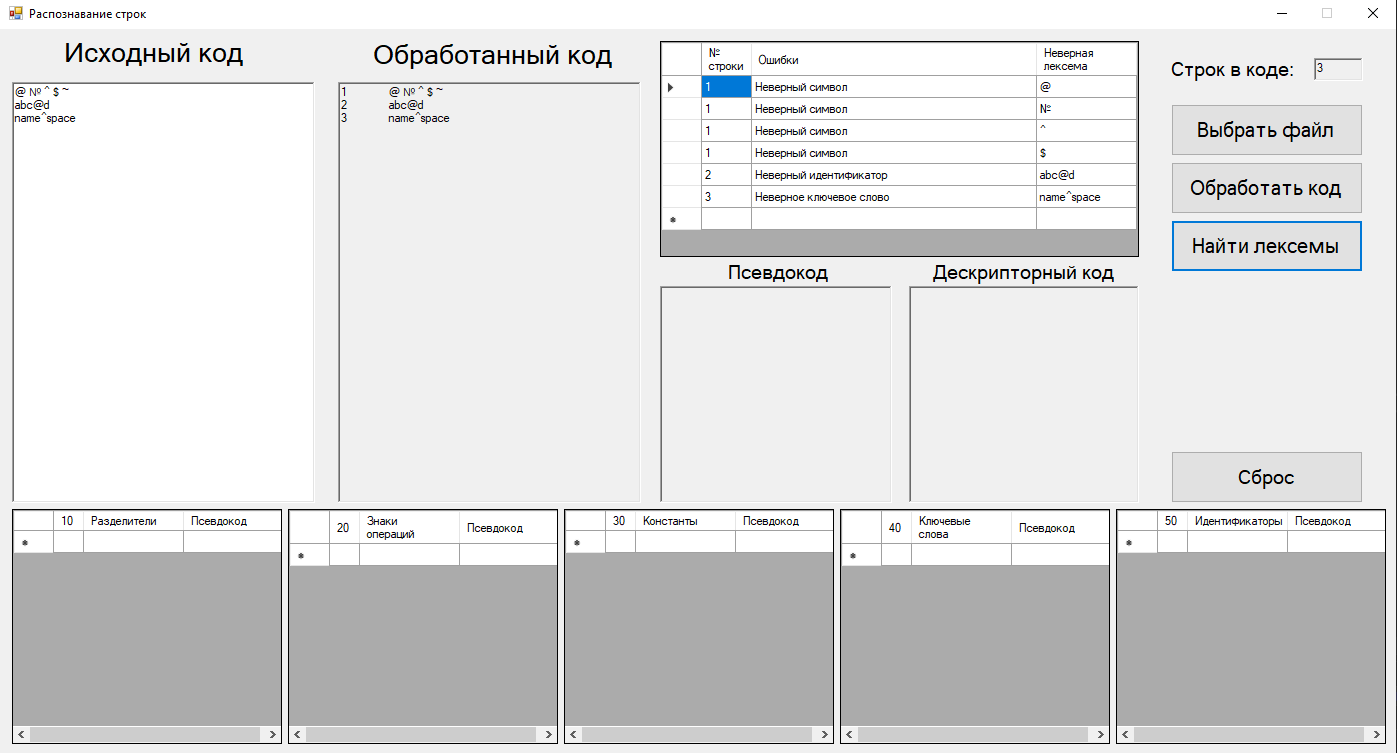
1. Тестирование











2. Программный код

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

String^ str0 = prog;

String^ symbol;

String^ count;

String^ pk;

string str00 = marshal\_as<string>(str0), str = "";

bool f = false;

char c;

int s = 0;

for (int i = 0; i < str00.length(); i++) {

c = str00[i];

Automat2(c, s, str, i, f);

if (c == ' ' || c == '\n') richTextBox4->Text += Char::ToString(c);

if (s == 0) { //ошибки

if (f) {

symbol = "Неверный символ: ";

symbol += marshal\_as<String^>(str);

cli::array<String^>^ row = gcnew cli::array<String^> { symbol };

dataGridView6->Rows->Add(row);

symbol = "";

str = "";

f = false;

}

}

if (s == 1) { //разделители

if (f) {

symbol += marshal\_as<String^>(str);

if (!search(dataGridView1, symbol)) {

count\_r++;

count = Convert::ToString(count\_r);

pk = symbol;

cli::array<String^>^ row = gcnew cli::array<String^> { count, symbol, pk };

dataGridView1->Rows->Add(row);

richTextBox4->Text += pk;

richTextBox5->Text += "(10, " + count + "), ";

}

else {

richTextBox4->Text += psevdo;

richTextBox5->Text += "(10, " + id + "), ";

}

symbol = "";

str = "";

f = false;

s = 0;

count = "";

}

}

if (s >= 2 && s <= 9) { //знаки

if (f) {

for (int i = 0; i < str.length(); i++)

symbol += Char::ToString(str[i]);

if (!search(dataGridView2, symbol)) {

count\_z++;

count = Convert::ToString(count\_z);

pk = symbol;

cli::array<String^>^ row = gcnew cli::array<String^> { count, symbol, pk };

dataGridView2->Rows->Add(row);

richTextBox4->Text += pk;

richTextBox5->Text += "(20, " + count + "), ";

}

else {

richTextBox4->Text += psevdo;

richTextBox5->Text += "(20, " + id + "), ";

}

symbol = "";

str = "";

f = false;

s = 0;

count = "";

}

}

if (s >= 10 && s <= 13) { //константы

if (f) {

for (int i = 0; i < str.length(); i++)

symbol += Char::ToString(str[i]);

if (symbol[symbol->Length - 1] == '.' || symbol[symbol->Length - 1] == 'E' || symbol[symbol->Length - 1] == 'e' || symbol[symbol->Length - 1] == '+' || symbol[symbol->Length - 1] == '-') {

symbol = "Неверная константа: " + symbol;

cli::array<String^>^ row = gcnew cli::array<String^> { symbol };

dataGridView6->Rows->Add(row);

}

else if (!search(dataGridView3, symbol)) {

count\_const++;

count = Convert::ToString(count\_const);

pk = symbol;

cli::array<String^>^ row = gcnew cli::array<String^> { count, symbol, pk };

dataGridView3->Rows->Add(row);

richTextBox4->Text += pk;

richTextBox5->Text += "(30, " + count + "), ";

}

else {

richTextBox4->Text += psevdo;

richTextBox5->Text += "(30, " + id + "), ";

}

symbol = "";

str = "";

f = false;

s = 0;

count = "";

}

}

if (s == 17) { //ключевые слова

if (f) {

for (int i = 0; i < str.length(); i++)

symbol += Char::ToString(str[i]);

if (!search(dataGridView4, symbol)) {

count\_key++;

count = Convert::ToString(count\_key);

pk = symbol;

cli::array<String^>^ row = gcnew cli::array<String^> { count, symbol, pk };

dataGridView4->Rows->Add(row);

richTextBox4->Text += pk;

richTextBox5->Text += "(40, " + count + "), ";

}

else {

richTextBox4->Text += psevdo;

richTextBox5->Text += "(40, " + id + "), ";

}

symbol = "";

str = "";

f = false;

s = 0;

count = "";

pk = "";

}

}

if (s == 999) { //идентификаторы

if (f) {

for (int i = 0; i < str.length(); i++)

symbol += Char::ToString(str[i]);

if (!search(dataGridView5, symbol)) {

count\_id++;

count = Convert::ToString(count\_id);

pk = "id" + count;

cli::array<String^>^ row = gcnew cli::array<String^> { count, symbol, pk };

dataGridView5->Rows->Add(row);

richTextBox4->Text += pk;

richTextBox5->Text += "(50, " + count + "), ";

}

else {

richTextBox4->Text += psevdo;

richTextBox5->Text += "(50, " + id + "), ";

}

symbol = "";

str = "";

f = false;

s = 0;

count = "";

}

}

}

if (richTextBox5->Text != "") richTextBox5->Text = richTextBox5->Text->Substring(0, richTextBox5->Text->Length - 1);

string pcode = marshal\_as<string>(richTextBox4->Text);

s = 0;

for (int i = 0; i < pcode.length(); i++) {

Automat1(pcode[i], s);

}

richTextBox4->Text = marshal\_as<String^>(pcode);

Вывод

В ходе выполнения лабораторной работы была написана часть лексического анализатора по обработке лексических ошибок. Ошибки обрабатываются по следующему принципу: обнаружение ошибки, указание номера строки с ошибкой, идентификация ошибки.