**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе № 1**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: Рекурсия**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6383 |  | Михеева Е.Е. |
| Преподаватель |  | Шолохова О.М. |

Санкт-Петербург

2017

Ознакомление с основными понятиями и приёмами рекурсивного программирования, получение навыков программирования рекурсивных процедур и функций на языке программирования С++

1. Постановка задачи

1. проанализировать полученное задание, выделив рекурсивно определяемые информационные объекты и (или) действия;
2. разработать программу, использующую рекурсию;
3. сопоставить рекурсивное решение с итеративным решением задачи;
4. сделать вывод о целесообразности и эффективности рекурсивного решения данной задачи.

Задание: Построить синтаксический анализатор для понятия *вещественное число*.

*вещественное\_число::= целое\_число . целое\_без\_знака |*

*целое\_число . целое\_без\_знака*Е*целое число |*

*целое\_число*Е*целое\_число*

*целое\_без\_знака::=цифра*| *цифра целое\_без\_знака*

*целое\_число::=целое\_без\_знака | + целое\_без\_знака | −целое\_без\_знака*

2. Основные теоретические положения.

В данной задаче требуется построить синтаксический анализатор для понятия вещественное число. Понятие определенно рекурсивно, так как оно зависит от понятия целое без знака, которое определенно рекурсивно следующим образом:

*целое\_без\_знака::=цифра*| *цифра целое\_без\_знака*

Вычисления заканчиваются в случае, если встречается некорректный символ или конец строки.

3. Спецификация программы. Программа предназначена для определения, является ли входная строка вещественным числом.

4. Описание программы.

Программа написана на языке С++ с использованием компилятора cl версии

17.00.60610.1 для x64.

Входными данными для программы является строка, вводимая пользователем с клавиатуры. Выходными данными является сообщение о том, является ли строка вещественным числом, и если нет, то выводится причина, по которой оно не является.

5. Анализ полученных результатов

Пример диалога с пользователем:

Введите строку: 12.3456

Число вещественное

Реализация.

Глобальные переменные:

bool INCORRECT\_SYMBOL – флаг встречи некорректного символа

Функции:

1. bool real(std::istream &in, int &points,int &expon);

Выясняет, является ли число вещественным

Входные параметры:

1) in – входной поток

Выходные параметры:

1) points – количество точек

2) expon – количество символов “e”

Возвращемое значение: true, если число вещественное и false в противном случае

2. bool integer(std::istream &in, int &points, int &expon)

Выясняет, является ли число целым

Входные параметры:

1) in – входной поток

2) points – количество точек

3) expon – количество символов “e”

Возвращемое значение: true, если число вещественное и false в противном случае

Входные параметры:

1) in – входной поток

2) points – количество точек

3) expon – количество символов “e”

3. bool u\_integer(std::istream &in, int &points, int &expon)

Входные параметры:

1) in – входной поток

Выходные параметры:

1) points – количество точек

Возвращемое значение: true, если число беззнаковое целое или false в противном случае

4. int figure(char c)

Выясняет, какой символ

Входные параметры:

1) c – символ, считанный с потока

Возвразаемые значения:

1) 0 – символ – цифра

2) 2 – символ конца строки

3) 3 – символ экспонента

4) 4 – символ точка

5) некорректный символ

5. void what\_fail(int points, int expon)

Выводит тип ошибки.

Входные параметры:

1) points – количество точек

2) количество экспонент

Ничего не возвращает.

6. Тестирование

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 12.54 | Число вещественное |
| 234 | Число целое |
| 12е5 | Число вещественное |
| 23а4 | Некорректный символ |
| 12е5.6 | Число иррациональное |
| 12е6е4 | Некорректный символ |
| 12.56е6 | Число вещественное |

Выводы:

1. Программа корректно определяет, является ли число вещественным

2. В ходе выполнения работы была освоена рекурсия и работа с рекурсивными функциями.

Приложение A. Исходный код.

//

// Лабораторная работа №1

// lab1.cpp

// Михеева Екатерина, группа 6383

// 16.09.17

// Программа для определения, является ли введенная строка вещественным //числом

// вещественное\_число::= целое\_число . целое\_без\_знака |

целое\_число . целое\_без\_знакаЕцелое число |

целое\_числоЕцелое\_число

//целое\_без\_знака::=цифра | цифра целое\_без\_знака

//целое\_число::=целое\_без\_знака | + целое\_без\_знака | −целое\_без\_знака

#include "stdafx.h"

#include "iostream"

using namespace std;

bool INCRORRECT\_SYMBOL = 0;

void what\_fail(int points, int expon)

{

if (points == 0 && expon == 1)

{

cout << "Число иррациональное" << endl;

return;

}

if (points == 0 && expon == 0 && INCRORRECT\_SYMBOL == 0) {

cout << "Число целое" << endl;

return;

}

cout << "Некорректный символ" << endl;

return;

}

bool real(std::istream &in, int &points,int &expon);

int figure(char c)

{

if (c <= '9' && c >= '0')

return 0;

if (c == '\n')

return 2;

if (c == 'e')

return 3;

if (c == '.')

return 4;

return 5;

}

bool u\_integer(std::istream &in, int &points, int &expon);

bool integer(std::istream &in, int &points, int &expon)

{

switch (in.peek())

{

case '+':

in.get();

if (u\_integer(in, points, expon) == false) return false;

break;

case '-':

in.get();

if (u\_integer(in,points,expon) == false) return false;

break;

case '\n': {

if (points == 1 || expon == 1)return true;

return false;

}

default:

if (u\_integer(in, points,expon) == false) return false;

}

}

bool u\_integer(std::istream &in, int &points, int &expon)

{

char c = cin.peek();

switch (figure(c))

{

//цифра

case 0:

in.get();

u\_integer(in,points,expon);

break;

//точка

case 4:

if (expon == 1 || points == 1) return false;

points++;

in.get();

if (figure(in.peek()) != 0 )

{

INCRORRECT\_SYMBOL = 1;

return false;

}

u\_integer(in,points,expon);

break;

//конец строки

case 2:

if (points == 1 || expon == 1)

return true;

return false;

//экспонента

case 3:

if (expon == 1)

{

expon++;

return false;

}

expon++;

in.get();

if (figure(in.peek()) != 0 )

{

INCRORRECT\_SYMBOL = 1;

return false;

}

integer(in,points,expon);

break;

case 5:

INCRORRECT\_SYMBOL = 1;

return false;

}

}

bool real(std::istream &in, int &points, int &expon)

{

char c = in.peek();

if (c == '.' || c == 'e') {

INCRORRECT\_SYMBOL = 1;

return false;

}

if(integer(in, points,expon) == false) return false;

return true;

}

int main()

{

int points = 0, expon = 0;

setlocale(0, "rus");

cout << "Введите строку" << endl;

cin;

if (real(cin, points,expon) == true)

cout << "Число вещественное\n";

else what\_fail(points, expon);

system("pause");

return 0;

}

Приложение Б. Блок-схема

