

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЁТ
по лабораторной работе №1
по дисциплине «Алгоритмы и структуры данных»
Тема: Рекурсия

Студентка гр. 7383

Иолшина В.

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2018

Содержание

1. ЦЕЛЬ РАБОТЫ	3
2. РЕАЛИЗАЦИЯ ЗАДАЧИ.	4
3. ТЕСТИРОВАНИЕ	6
4. ВЫВОД	7
5. ПРИЛОЖЕНИЕ А: ИСХОДНЫЙ КОД ПРОГРАММЫ	8
6. ПРИЛОЖЕНИЕ Б: ТЕСТОВЫЕ СЛУЧАИ	15

1. ЦЕЛЬ РАБОТЫ

Цель работы: познакомиться с основными понятиями и приемами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

Формулировка задачи: Построить синтаксический анализатор для понятия вещественное число:

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

2. РЕАЛИЗАЦИЯ ЗАДАЧИ.

В данной работе используется функция `main` и 4 функции:

```
bool isInt(ifstream &infile, char &ref);  
bool isUnsInt(ifstream &infile, char &ref, int i);  
void Error(short k);  
bool isNumber(ifstream &infile, char s).
```

В функции `main` на консоль выводится циклическое меню, где пользователь может выбрать, ввести вещественное число самостоятельно, выбрав «1» или использовать число, хранящееся в файле `ex.txt`, выбрав «2». В случае выбора «3», произойдет выход из программы. Далее для введенного числа или успешного считанного числа из файла происходит вызов функции `isNumber(ifstream &infile, char s)`, которая, считывая число посимвольно, проверяет каждый символ на соответствие формулировке задачи. Для первого символа вызывается функция `isInt(ifstream &infile, char &ref)`, которая проверяет, является ли символ плюсом или минусом и вызывает для этого же или следующего символа функцию `isUnsInt(ifstream &infile, char &ref, int i)`. Эта функция, в свою очередь, проверяет, является ли символ цифрой и далее для каждого последующего символа рекурсивно вызывает саму себя. В случае, когда последовательность цифр заканчивается, функция передает по ссылке `&ref` символ, на котором остановилась проверка и возвращается в функцию `isNumber(ifstream &infile, char s)`. Далее символ проверяется на соответствие точке или экспоненте и снова вызывается функция `isUnsInt(ifstream &infile, char &ref, int i)`, чтобы проверить следует ли за точкой или экспонентой беззнаковое число. На этом этапе число может быть считано полностью и функция `main` может дать ответ, является ли оно вещественным. Если число было считано не полностью, функция `isNumber(ifstream &infile, char s)` будет продолжать считывание и проверку, является ли следующий символ экспонентой, а последующий целым числом.

На каждом этапе посимвольного считывания числа предусмотрен вызов функции `Error(short k)` в случае считывания символа, не соответствующего поставленной задаче. Параметром, переданный функции, содержит код

возникшей ошибки. Функция `Error(short k)` в зависимости от кода, переданного ей, выводит описание ошибки. Функция `main` завершается выводом информации о том, было ли число вещественным.

3. ТЕСТИРОВАНИЕ

3.1. ПРОЦЕСС ТЕСТИРОВАНИЯ

Программа собрана в операционной системе Ubuntu 16.04.2 LTS", с использованием компилятора g++ (Ubuntu 5.4.0-6ubuntu1~16.04.5). В других ОС и компиляторах тестирование не проводилось.

3.2. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Тестовые случаи представлены в Приложении А.

Во время тестирования была обнаружена ошибка: если целая и дробная части вещественного числа состояли из более чем 1 цифры, функция `bool isUnsInt(ifstream &infile, char &ref, int i)`, считывающая последующие символы и проверяющая их на соответствие цифре, не возвращала символ, на котором закончилось считывание, и дальнейшие операции в функции `bool isNumber(ifstream &infile, char s)` проходили с уже обработанными символами, соответственно программа работала неверно. Для устранения ошибки было принято решение передавать в функции `bool isInt(ifstream &infile, char &ref)`, `bool isUnsInt(ifstream &infile, char &ref, int i)` обрабатываемый символ по ссылке. После исправления, тестовые случаи не выявили неверной работы программы. Это говорит о том, что поставленная задача была выполнена.

4. ВЫВОД

В ходе работы были изучены основные понятия рекурсивного программирования на языке C++. Был реализован синтаксический анализатор для понятия «вещественное число», а также разработано циклическое меню взаимодействия с пользователем.

5. ПРИЛОЖЕНИЕ А: ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>
#include <fstream>
#include <cctype>

using namespace std;

bool isInt(ifstream &infile, char &ref);

bool isUnsInt(ifstream &infile, char &ref, int i);

void Error(short k);

bool isNumber(ifstream &infile, char s)
{
    bool b;
    if(infile >> s)
    {
        cout << s;
        b = isInt(infile, s);
        if(b)
        {
            if(s == '.')
            {
                if(infile >> s)
                {
                    cout << s;
                    int i=0;
                    b = isUnsInt(infile, s, i);
                    if(b)
                    {
                        if (s == 'E' || s == 'e')
                        {
                            if(infile >> s)
                            {
                                cout << s;
                                b = isInt(infile, s);
                                if(b)
                                    return true;
                            }
                        }
                    }
                }
            }
        }
    }
}
```



```

        else
        {
            Error(3);
            return false;
        }

    }
    else
        return true;
}
else
    return true;
}
else
{
    Error(6);
    return false;
}
}
else
{
    Error(4);
    return false;
}
}
else if (s == 'E' || s == 'e')
{
    if(infile >> s)
    {
        cout << s;
        b = isInt(infile, s);
        if(b)
            return true;
        else
        {
            Error(3);
            return false;
        }
    }
}

```

```

        else
            return true;
    }
    else
    {
        Error(5);
        return false;
    }
}
else
{
    Error(3);
    return false;
}
}
else
{
    Error(2);
    return false;
}
}

```

```

bool isInt(ifstream &infile, char &ref)
{
    bool b;
    int i=0;
    if(ref == '+' || ref == '-')
    {
        if(infile >> ref)
        {
            cout << ref;
            b = isUnsiInt(infile, ref, i);
        }
        else return false;
    }
    else b = isUnsiInt(infile, ref, i);
    if(b) return true;
    else return false;
}

```

```
}
```

```
bool isUnsInt(ifstream &infile, char &ref, int i)
```

```
{
```

```
    bool b;
```

```
    b = isdigit(ref);
```

```
    i++;
```

```
    if(b)
```

```
    {
```

```
        if(infile >> ref)
```

```
        {
```

```
            cout << ref;
```

```
            isUnsInt(infile, ref, i);
```

```
        }
```

```
        else return true;
```

```
    }
```

```
    else if (i == 1)
```

```
        return false;
```

```
    else return true;
```

```
}
```

```
void Error (short k)
```

```
{
```

```
    cout << endl << "err#" << k << endl;
```

```
    switch (k)
```

```
    {
```

```
        case 1: cout << "! - Лишние символы во входной строке" << endl; break;
```

```
        case 2: cout << "! - Отсутствует начальный символ" << endl; break;
```

```
        case 3: cout << "! - Символ не является целым числом" << endl; break;
```

```
        case 4: cout << "! - Отсутствует нужный символ" << endl; break;
```

```
        case 5: cout << "! - Отсутствует точка или экспонента" << endl; break;
```

```
        case 6: cout << "! - Символ не является целым беззнаковым" << endl;
```

```
    break;
```

```

        case 7: cout << "! - Символ не является экспонентой" << endl; break;

        case 8: cout << "! - " << endl; break;

        default : cout << "! - ...";break;
    };
}

int main()
{
    char s;
    char ss[100];
    char &ref = s;
    bool b, exit = true;
    int n;
    while(exit)
    {
        cout << "Введите 1, если хотите ввести вещественное число" << endl
        << "Введите 2, если хотите использовать число из файла ex.txt" << endl <<
        "Введите 3, чтобы завершить работу" << endl;
        cin >> n;
        switch(n)
        {
            case 1:
            {
                ofstream tempfile("ex.txt");
                cout << "Введите вещественное число:" << endl;
                cin >> ss;
                tempfile << ss;
                tempfile.close();
                ifstream infile ("ex.txt");
                if(!infile)
                cout << "Входной файл не открыт" << endl;
                cout << "Анализатор для вещественного числа:" << endl;
                b = isNumber(infile, s);
                infile >> s;
                if(b && !infile.eof())
                {

```

```

        Error(1);
        return false;
    }
    b = (b && infile.eof());
    infile.close();
    if(b)
        cout << endl << "Это вещественное число" << endl;
    else
        cout << endl << "Это НЕ вещественное число" << endl;
    }
    break;
case 2:
    {
        ifstream infile ("ex.txt");
        if(!infile)
            cout << "Входной файл не открыт" << endl;
        cout << "Анализатор для вещественного числа:" << endl;
        b = isNumber(infile, s);
        infile >> s;
        if(b && !infile.eof())
        {
            Error(1);
            return false;
        }
        b = (b && infile.eof());
    }
    if(b)
        cout << endl << "Это вещественное число" << endl;
    else
        cout << endl << "Это НЕ вещественное число" << endl;
    break;
case 3:
    {
        exit = false;
        return 0;
    }
default:
    cout << "Повторите попытку" << endl;
    break;

```

```
    }  
    }  
    return 0;  
}
```

6. ПРИЛОЖЕНИЕ Б: ТЕСТОВЫЕ СЛУЧАИ.

Ввод	Вывод	Верно
65436.29043	Это вещественное число	Да
-23451.9473678	Это вещественное число	Да
64365e-39245	Это вещественное число	Да
+43475E+23947	Это вещественное число	Да
34258e-7412734	Это вещественное число	Да
94875.23875e-23874	Это вещественное число	Да
-32478.2347E32984	Это вещественное число	Да
+23848.10239e+82747	Это вещественное число	Да
238	Это НЕ вещественное число	Да
34295.	Это НЕ вещественное число	Да
.3485	Это НЕ вещественное число	Да
234987e	Это НЕ вещественное число	Да
-3249.e	Это НЕ вещественное число	Да
E.234	Это НЕ вещественное число	Да