**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**КАФЕДРА САПР**

отчет

**по лабораторной работе №1**

**по дисциплине «Программирование»**

Тема: **«Представление строки символов с маркером, ее обработка»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 0336 |  | Насекин С.А. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2021

Основное задание: удалить из заданной строки символы, стоящие между скобками. Вид скобок задаётся.

Условия:

1) Реализация на С++

2) Использовать только примитивные типы и операции ввода/вывода.

Дано:

Текстовый файл input.txt содержащий строку:

“Tom did play hookey, and he had a very good time. [Bla, Bla, Bla, Bla, Bla, Bla, Bla, Bla, Bla, Bla, Bla, Bla] He got back home barely in season to help Jim, the small colored boy, saw next-day's wood and split the kindlings before supper--at least he was there in time {Useless information} to tell his adventures to Jim while Jim did three-fourths of the work. Tom's younger brother (or rather half-brother) Sid was already through with his part of the work (picking up chips), for he was a quiet boy, and had no adventurous, trouble-some ways.”

Необходимо преобразовать строку таким образом, чтобы все символы между заданным видом скобок (Включая скобки) были удалены, а результат записан в файл result.txt

Способ решения:

1) Осуществить загрузку строки из файла.

2) Преобразовать данную строку в строку с маркером.

3) Создать алгоритм поиска и удаления символов в скобках, с сохранением целостности строки.

4) Вывести результат в файл.

Для хранения строки символов с маркером, используется класс MyString. Который содержит поля length, str, MARKER, bracketsType.

class MyString {

public:

int length = 0;

char \*str = nullptr;

const char MARKER = '@';

int bracketsType;

Для загрузки строки из файла используется функция ifstream из библиотеки <fstream>

ifstream input("..\\input.txt");

if (input.is\_open()){

char\* temp = new char[1000];

input.getline(temp, 1000);

str.addString(temp);

} else{

cout << "File didn't find" << endl;

return 1;

} input.close();

Для внесения строки в класс MyString используется метод addString с параметром const char\* str – принимаемая текстовая строка.

void addString(const char \*str) {

for (int i = 0; str[i] != '\000'; ++i) {

length++;

}

this->str = new char[length + 1];

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

this->str[i] = str[i];

}

this->str[length] = MARKER;

}

Для выбора типа скобок используется стандартный ввод\вывод, доступные виды скобок (), [], {}. Данные виды скобок являются наиболее используемыми.

cout << "Chise type of brackets: " << endl;

cout << "1 - () " << endl;

cout << "2 - [] " << endl;

cout << "3 - {} " << endl;

cin >> str.bracketsType;

Для удаления символов между выбранными скобками используется метод DeleteTextInBrackets() без параметров, поскольку тип скобок был выбран и внесён в класс MyString ранее.

void DeleteTextInBrackets() {

char first;

char second;

if (bracketsType == 1) {

first = '(';

second = ')';

}

if (bracketsType == 2) {

first = '[';

second = ']';

}

if (bracketsType == 3) {

first = '{';

second = '}';

}

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

if (str[i] == first) {

for (int j = i; j < length; j++) {

if (str[j] == second) {

int tempL = length - ((j - i) + 1);

char \*temp = new char[tempL];

int f = 0;

for (; f < i; f++) {

temp[f] = str[f];

}

for (int d = j + 1; d <= length; d++, f++) {

temp[f] = str[d];

}

length = tempL;

str = temp;

}

}

}

}

}

Для вывода на экран строки с маркером используется метод Print.

void Print() {

for (int i = 0; str[i] != MARKER; i++) {

cout << str[i];

}

}

Файл main:

int main() {

setlocale(LC\_ALL, "ru");

cout << "Laboratory work number 1" << endl;

cout << "Completed by a student of the group 0336" << endl;

cout << "Nasekin S.A." << endl << endl;

MyString str;

ifstream input("..\\input.txt");

if (input.is\_open()){

char\* temp = new char[1000];

input.getline(temp, 1000);

str.addString(temp);

} else{

cout << "File didn't find" << endl;

return 1;

}

input.close();

cout << "current marker: " << str.MARKER << endl;

cout << "Original text:" << endl;

str.Print();

cout << "Select bracket type: " << endl;

cout << "1 - () " << endl;

cout << "2 - [] " << endl;

cout << "3 - {} " << endl;

cout << "Your reply: ";

cin >> str.bracketsType;

cout << endl << "Modified text:" << endl;

str.DeleteTextInBrackets();

str.Print();

ofstream result("..\\result.txt",ios::out);

for (int i = 0; str.str[i]!=str.MARKER; ++i) {

result << str.str[i];

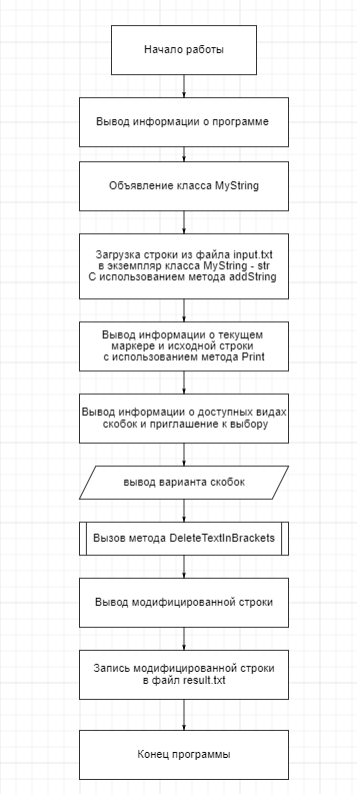
}

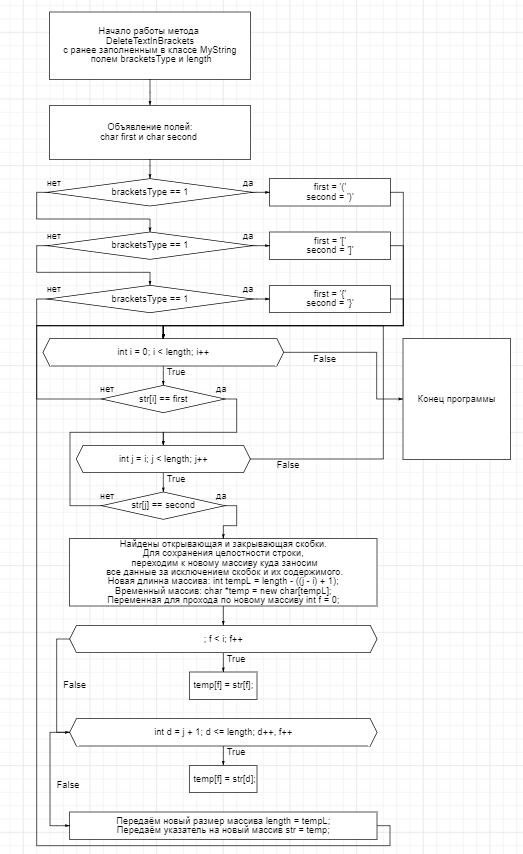
result.close();

cout << endl << endl << "Modified text written to file." << endl;

return 0;

}

Общий вид программы:

 Метод DeleteTextInBrackets

Код программы:

#include <iostream>

#include <fstream>

using namespace std;

class MyString {

public:

int length = 0;

char \*str = nullptr;

const char MARKER = '@';

int bracketsType;

void addString(const char \*str) {

for (int i = 0; str[i] != '\000'; ++i) {

length++;

}

this->str = new char[length + 1];

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

this->str[i] = str[i];

}

this->str[length] = MARKER;

}

void Print() {

for (int i = 0; str[i] != MARKER; i++) {

cout << str[i];

}

}

/\* void СhoiceBrackets() \*/

void DeleteTextInBrackets() {

char first;

char second;

if (bracketsType == 1) {

first = '(';

second = ')';

}

if (bracketsType == 2) {

first = '[';

second = ']';

}

if (bracketsType == 3) {

first = '{';

second = '}';

}

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

if (str[i] == first) {

for (int j = i; j < length; j++) {

if (str[j] == second) {

int tempL = length - ((j - i) + 1);

char \*temp = new char[tempL];

int f = 0;

for (; f < i; f++) {

temp[f] = str[f];

}

for (int d = j + 1; d <= length; d++, f++) {

temp[f] = str[d];

}

length = tempL;

str = temp;

}

}

}

}

}

};

int main() {

setlocale(LC\_ALL, "ru");

cout << "Laboratory work number 1" << endl;

cout << "Completed by a student of the group 0336" << endl;

cout << "Nasekin S.A." << endl << endl;

MyString str;

ifstream input("..\\input.txt");

if (input.is\_open()){

char\* temp = new char[1000];

input.getline(temp, 1000);

str.addString(temp);

} else{

cout << "File didn't find" << endl;

return 1;

}

input.close();

cout << "current marker: " << str.MARKER << endl;

cout << "Original text:" << endl;

str.Print();

cout << "Select bracket type: " << endl;

cout << "1 - () " << endl;

cout << "2 - [] " << endl;

cout << "3 - {} " << endl;

cout << "Your reply: ";

cin >> str.bracketsType;

cout << endl << "Modified text:" << endl;

str.DeleteTextInBrackets();

str.Print();

ofstream result("..\\result.txt",ios::out);

for (int i = 0; str.str[i]!=str.MARKER; ++i) {

result << str.str[i];

}

result.close();

cout << endl << endl << "Modified text written to file." << endl;

return 0;

}