**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 |  | Полещук А.В. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2021

Исходная формулировка задания

Преобразовать заданную строку следующим образом: удалить группы пробелов, которыми, возможно, начинается и заканчивается строка.

1. Постановка задачи:
2. Разработать способ хранения строки символов с маркером
3. Организовать в памяти динамическое размещение данных в виде массива символов
4. Обеспечить эффективное выполнение операция по преобразованию строки
5. Реализовать чтение строки из файла любого размера

Этапы решения задачи:

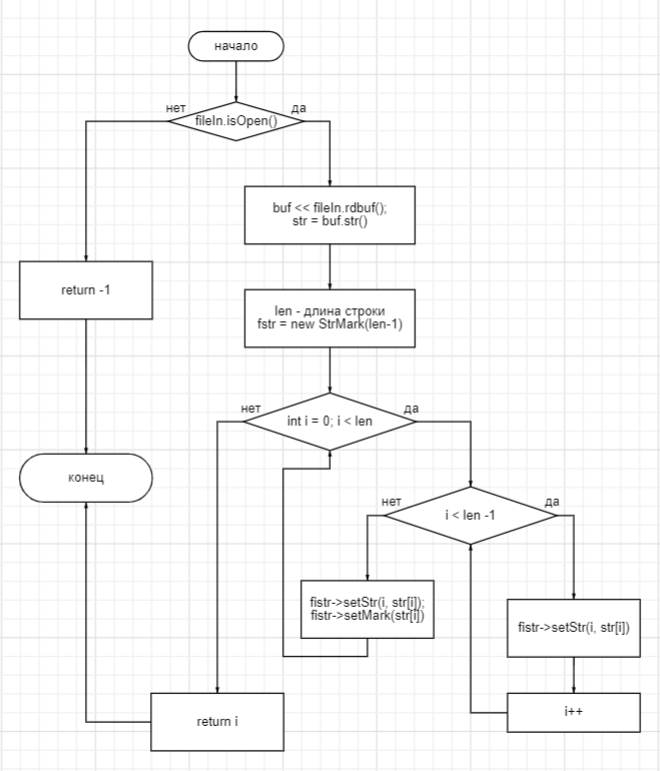
* Постановка задачи и определение особенностей реализации;
* Разработка класса для ввода и вывода строки;
* Проверка работы;
* Разработка основного алгоритма программы;
* Тестирование.

1. Анализ вариантов хранения и обработки информации.

Целью является реализация задачи представления строки символов с маркером. Для этого в программе используется массив символов char, который инициализируется в зависимости от размера входящей строки.

Задача: реализовать класс, представляющий строку символов с маркером в виде динамического массива символов char и записать строку и маркер в структуру для хранения.

1. Описание методов.
2. Для чтения строки из файла используется метод класса LabIO: int LabIO::loadFile(). После того, как файл был открыт мы записываем содержимое всего файла в локальную переменную типа string, дальше идет ее обработка(определение длины строки, нахождение строки и маркера) для записи в объект класса StrMark. Возвращаемым значением метода будет длина полученной строки, либо -1, если не удалось открыть файл. Таким образом строка любого размера будет без изменений храниться в объекте.



1. Конструктор класса LabIO: вызывает функцию load() для загрузки
2. Метод для вывода в файл: по своей структуре практически аналогична функции ввода из файла. Используется поток вывода ofstream. В цикле while проверяется явялется ли элемент маркером. После чего он передается в поток вывода.
3. Метод StrMark\* LabIO::getStr() возвращает указатель на объект класса StrMark, в который были записаны данные из файла.
4. Класс StrMark реализован для хранения строки, его методы(StrMark::setStr, StrMark::setMark, StrMark::getStr, StrMark::getMark) используются для записи и вывода полученной строки в классе LabIO.
5. В классе EditStr реализован метод void EditStr::delSpace(StrMark \*strM). Данный метод проверяет строку на наличие пробелов в начале и в конце строки, после чего удаляет их.
6. Программа на C++:

Файл main.cpp:

#include "labio.h"  
#include "editstr.h"  
  
int main(){  
 StrMark \*fstr;  
 LabIO \*io = new LabIO();  
 fstr = io->getStr();  
 EditStr \*edStr = new EditStr();  
 edStr->delSpace(fstr);  
 io->saveFile(fstr);  
 return 0;  
}

Файл LabIO.cpp:

#include "labio.h"  
  
using namespace std;  
  
LabIO::LabIO(){  
 loadFile();  
}  
  
LabIO::~LabIO(){  
 delete fistr;  
}  
  
int LabIO::loadFile(){  
 int i;  
 int len;  
 fileIn.open("C:\\lab1\\lab1.txt", ios\_base::in);  
 if(fileIn.is\_open()) {  
 cout << "File is open to load." << endl;  
 }  
 else {  
 cout << "File isn't open." << endl;  
 return -1;  
 }  
 stringstream buf;  
 string str;  
 buf << fileIn.rdbuf();  
 str = buf.str();  
 len = str.length();  
 cout << "len is " << len << endl;  
 fistr = new StrMark(len-1);

cout << "original string is:" << endl << "(";  
 for(i = 0; i < len; i++){  
 if(i < len-1) {fistr->setStr(i, str[i]); cout << fistr->getStr(i) << endl;}  
 else {  
 fistr->setStr(i, str[i]);  
 fistr->setMark(str[i]);  
 cout << "mark is: " << fistr->getMark() << endl;  
 }  
 }

cout << ")" << endl;  
 fileIn.close();  
 return i;  
}  
  
int LabIO::saveFile(StrMark \*strM){  
 fileOut.open("C:\\lab1\\lab1.txt", ios\_base::out);  
 if(fileOut.is\_open()) {  
 cout << "File is open to save." << endl;  
 }  
 else {  
 cout << "File isn't open." << endl;  
 return -1;  
 }  
 int i = 0;  
 while(strM->getStr(i) != strM->getMark()){  
 cout << strM->getStr(i);  
 fileOut << strM->getStr(i);  
 i++;  
 }  
 fileOut << strM->getMark();  
 fileOut.close();  
 return i;  
}  
  
StrMark\* LabIO::getStr(){  
 return fistr;  
}

Файл LabIO.h:

#include "strmark.h"  
#include <iostream>  
#include <fstream>  
#include <sstream>  
#include <string>  
  
using namespace std;  
  
class LabIO{  
  
public:  
 LabIO();  
 ~LabIO();  
 int loadFile();  
 int saveFile(StrMark \*strM);  
 StrMark\* getStr();  
  
 private:  
 StrMark \*fistr;  
 ifstream fileIn;  
 ofstream fileOut;  
 char c;  
};

Файл StrMark.cpp:

#include "strmark.h"  
  
using namespace std;  
  
StrMark::StrMark(unsigned len){  
 strLen = len;  
 strM = new char[strLen];  
}  
  
StrMark::~StrMark(){  
 delete strM;  
}  
  
void StrMark::setMark(char m){  
 mark = m;  
}  
  
char StrMark::getMark(){  
 return mark;  
}  
  
void StrMark::setStr(unsigned i, char str){  
 strM[i] = str;  
}  
  
char StrMark::getStr(unsigned i){  
 return strM[i];  
}  
  
void StrMark::strOut(){  
 int i = 0;  
 while(strM[i] != mark){  
 cout << strM[i];  
 i++;   
 }  
 cout << endl;  
}

Файл StrMark.h:

#ifndef **STRMARK\_H**#define **STRMARK\_H**#include <iostream>  
  
class StrMark{  
  
public:   
 StrMark(unsigned len);  
 ~StrMark();  
 void setMark(char mark);  
 char getMark();  
 void setStr(unsigned i, char str);  
 char getStr(unsigned i);  
 int lastInd();  
 void strOut();  
   
private:  
 char \*strM;  
 char mark;  
 unsigned strLen;  
};  
  
#endif

Файл EditStr.cpp:

#include "editstr.h"  
  
using namespace std;  
  
EditStr::EditStr(){  
  
}  
  
EditStr::~EditStr(){  
  
}  
  
void EditStr::delSpace(StrMark \*strM){  
 char str0;  
 char c;  
 char mark;  
 bool firstSign = false;  
 int i = 0; *//счетчик для всей строки* int ni = 0; *//счетчик для строки после нахождения первого символа не*

*//равного ' '* for(;;){  
 str0 = strM->getStr(i);  
 mark = strM->getMark();  
 if(str0 != ' ') firstSign = true;  
 if(firstSign) {  
 strM->setStr(ni, str0);  
 ni++;  
 }  
 if(str0 == mark){  
 str0 = strM->getStr(ni-2);  
 while(str0 == ' '){  
 ni--;  
 strM->setStr(ni, mark);  
 str0 = strM->getStr(ni-1);  
 }  
 break;  
 }  
 i++;  
 }  
}

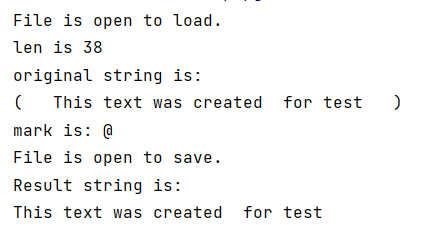
Файл EditStr.h:

#include "strmark.h"  
#include <cstring>  
  
class EditStr{  
  
public:   
 EditStr();  
 ~EditStr();  
  
 void delSpace(StrMark \*strM);  
};

1. Тестирование
2. Файл строк *lab1.txt*

*« This text was created fo test @»*

Результат:



1. Результат записан в файл lab1.txt
2. Вывод:
3. Разработан способ хранения строки символов с маркером.
4. Организовано динамическое размещение данных в виде массива символов (char \*).
5. Реализовано чтение строки из файла любого размера.