Факультет ИУ «Информатика и системы управления»

Кафедра ИУ-3 «Информационные системы и телекоммуникации»

**Отчет к лабораторной работе № 2**

по дисциплине «Алгоритмы и структуры данных»

«Строки»

Продолжительность работы 2 часа

|  |  |  |
| --- | --- | --- |
| Выполнил |  | Нгуен Чыонг Жанг ИУ3-31Б |
| Принял |  | Сакулин С.А. |

Москва 2020

**Цель работы**: В лабораторной работе необходимо разработать класс String, предоставляющий более удобные и безопасные средства для операций со строками нежели встроенный тип char\*.

**Задание**: Строка, содержащая произвольный русский текст, состоит не более чем из 200 символов. Написать, какие буквы и сколько раз встречаются в этом тексте. Ответ должен приводиться в грамматически правильной форме: например: **а** — 25 раз, **к** — 3 раза и т.д.

**Код**

#include <cstdint>

#include <fstream>

#include <iostream>

#include <string>

#include <utility>

#include <vector>

using namespace std;

int Len(char \*\_str) {

int size = 0;

while (\_str[size] != '\0') {

size++;

}

return size;

}

void Cpy(char \*in\_str, char \*src\_str) {

for (int i = 0; i < Len(in\_str); i++)

in\_str[i] = src\_str[i];

}

bool Cmp(char \*str\_f, char \*str\_s) {

int i = 0;

int max = Len(str\_f) > Len(str\_s) ? Len(str\_f) : Len(str\_s);

while (i < max && str\_f[i] == str\_s[i]) {

i++;

}

return i == max ? true : false;

}

class String {

private:

char \*str;

int len;

public:

String() {

str = new char[500];

\*str = '\0';

len = 0;

}

friend istream &operator>>(istream &in, String &str) {

in >> str.str;

return in;

}

friend ostream &operator<<(ostream &out, String &str) {

out << str.str;

return out;

}

String &operator=(const String &rhs) {

if (this != &rhs) {

this->str = new char[Len(rhs.str) + 1];

Cpy(this->str, rhs.str);

}

return \*this;

}

String &operator+=(const String &rhs) {

int size = Len(this->str) + Len(rhs.str);

char \*ts = new char[size + 1];

for (int i = 0; i < Len(this->str); i++)

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

for (int ii = Len(this->str), j = 0; ii <= size; ii++, j++)

ts[ii] = rhs.str[j];

delete[] this->str;

this->str = ts;

return \*this;

}

friend bool operator==(const String &lhs, const String &rhs) {

return Cmp(lhs.str, rhs.str);

}

friend bool operator!=(const String &lhs, const String &rhs) {

return !(lhs.str == rhs.str);

}

const char &operator[](int i) const { return this->str[i]; }

int StrLen() { return Len(this->str); }

void Insert(String &rhs, int pos) {

int sz = Len(this->str) + Len(rhs.str);

char \*td = new char(sz + 1);

\*td = '\0';

int j = 0;

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

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

j++;

}

for (int i = pos; i < pos + Len(rhs.str); i++) {

td[i] = rhs.str[i - pos];

}

for (int i = pos + Len(rhs.str); i <= sz; i++) {

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

j++;

}

delete[] str;

str = td;

}

~String();

};

String operator+(const String &lhs, const String &rhs) {

String ts = lhs;

return ts += rhs;

}

String::~String() { delete[] str; }

int Find\_Real\_Index(const char \*c) {

int index = 0;

while (c[index] != '\0')

index++;

return index;

}

void CounterChar(const char \*str) {

int l = Find\_Real\_Index(str);

int len = 'z' - 'a' + 1;

int \*res = new int[len];

for (int i = 0; i < l; i++)

res[str[i] - 'a']++;

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

if (res[i] != 0) {

cout << char(i + 'a') << " " << res[i] << endl;

}

}

}

int main() {

char str[1000];

ifstream file;

file.open("Lab2.txt", ios::in);

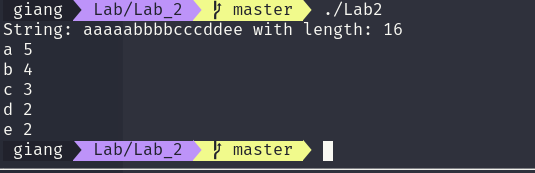
file >> str;

file.close();

CounterChar(str);

return 0;

}



**Вывод**: Класс string предоставляет более удобные возможности для работы со строками в C++ , нежели встроенный тип char\*, потому что он позволяет работать с динамической памятью, что позволяет легко складывать строки.