**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №4**

**по дисциплине «Программирование»**

Тема: **ТЕКСТОВЫЕ СТРОКИ**

|  |  |  |
| --- | --- | --- |
| Студент(ка) гр. | Минин М. А. |  |
| Преподаватель | Глущенко А. Г. |  |

Санкт-Петербург

2022

**Цель работы.**

Изучение текстовых строк как массивов символов и их обработке, а также

структуры «строка» из области имен std. Изучение потоков, используемых для работы с файлами и непосредственно чтения текстовой строки из файла. Написание программы, способной отредактировать входной текст, исключив из него наиболее очевидные ошибки.

**Основные теоретические положения.**

Класс string предназначен для работы со строками типа char, которые представляют собой строчку с завершающим нулем (символ ‘\0’). Класс string был введен как альтернативный вариант для работы со строками типа char. Чтобы использовать возможности класса string, нужно подключить библиотеку <string> и пространство имен std. Объявление же переменной типа string осуществляется схоже с обычной переменной.

Основными задачами при обработке текстовых строк являются следующие: определение фактической длины текста, копирование текста из одной строки в другую, объединение строк, сравнение (больше, меньше, равно) строк и т.п.

**Постановка задачи.**

Необходимо написать программу, которая:

1. Считывает строку с клавиатуры
2. Редактирует входной текст
   1. Удаляет лишние пробелы
   2. Удаляет лишние знаки препинания
   3. Исправляет регистр букв
3. Вывести на экран слова последовательности, не содержащие цифр
4. Вывести на экран ту же последовательность, заменив во всех словах первую букву соответствующей прописной буквой.

**Выполнение работы.**

Код программы представлен в приложении А.

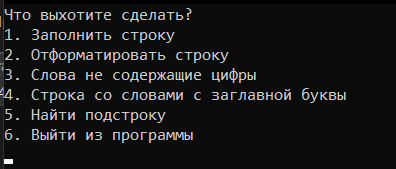
1. При запуске программы пользователю выводится меню доступных команд и ожидается ввод команды с клавиатуры.

Рис. 1 – Меню программы

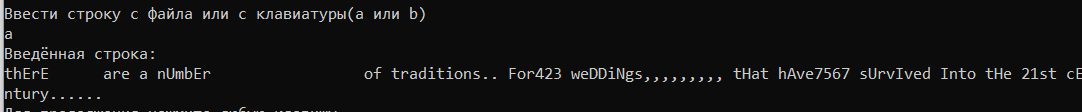
1. Выбор ввода строки.

Рис. 2 – Ввод строки

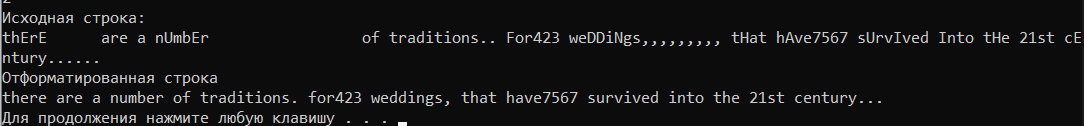
1. Форматирование строки.

Рис.3 – Форматирование строки

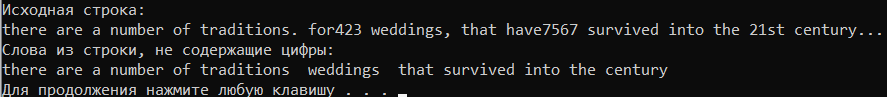
1. Вывод слов, не содержащих цифры

Рис.4 – Слова без цифр

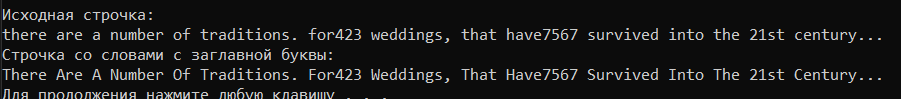
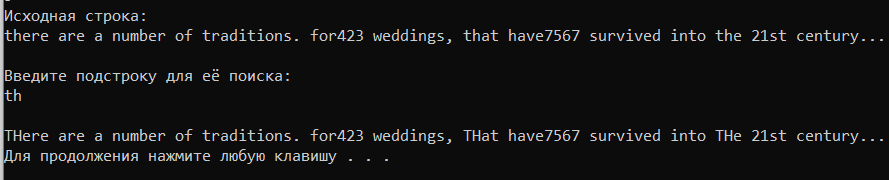
1. Строка со словами с заглавной буквы

Рис.5 – Слова с заглавной буквы

1. Поиск подстроки

**Вывод.**

В ходе работы были изучены текстовые строки как массивы, класс string и методы обработки строк.

.

Приложение А

рабочий код

#include <iostream>

#include <string>

#include <windows.h>

#include <fstream>

using namespace std;

// thErE are a nUmbEr of traditions.. For423 weDDiNgs,,,,,,,,, tHat hAve7567 sUrvIved Into tHe 21st cEntury......

void StrInput(string& str) {

ifstream File(R"(C:\Программирование\4\_прак\Test.txt)");

if (!File.is\_open()) {

cout << "Файл не открыт";

}

string str2;

while (!File.eof())

getline(File, str);

cout << "Введённая строка:" << '\n';

cout << str << '\n';

File.close();

}

void DeleteSymbols(string& str, char Symbol){

int Len = str.length();

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

{

int cnt = 0;

if (str[i] == Symbol)

{

while (str[i + cnt] == Symbol)

{

++cnt;

--Len;

}

if (cnt > 0)

str = str.erase(i + 1, cnt - 1);

}

}

}

void DeletePoints(string& str) {

char Symbol = '.';

int Len = str.length();

if (str.find(Symbol) >= 0) {

for (int i = str.find(Symbol); i < Len; i++) {

int cntPoints = 0;

int cntEllipsis = 0;

while (str[i + cntPoints] == Symbol)

++cntPoints;

cntEllipsis = cntPoints / 3;

if (cntEllipsis != 0) {

str = str.erase(i + 3, cntPoints - 3);

Len -= cntPoints - 3;

i += 3;

}

if (cntPoints == 2) {

str = str.erase(i + 1, 1);

Len--;

}

if (cntEllipsis > 1) {

str = str.erase(i, 3);

Len -= 3;

}

}

}

}

void DeleteNumbers(string& str, int& i) {

int Len = str.length();

string str2;

bool Number = false;

while (((str[i] != ' ') and (str[i] != ',')

and (str[i] != '!') and (str[i] != '?')

and (str[i] != ';') and (str[i] != ':')

and (str[i] != ')') and (str[i] != '.')

and (str[i] != '/')) and (i != Len))

{

if (str[i] >= 48 and str[i] <= 57)

Number = true;

str2 += str[i];

++i;

}

if (Number == false)

cout << str2 << ' ';

}

void UppercaseLetters(string str) {

int Len = str.length();

bool space = false;

for (int i = 0; i < Len-1; i++) {

if (str[i] == ' ')

str[i + 1] = toupper(str[i + 1]);

}

str[0] = toupper(str[0]);

cout << str << '\n';

}

void LinearSearch(string str) {

string Substring;

cout << "Исходная строка:" << '\n';

cout << str << '\n';

getline(cin, Substring);

cout << '\n' << "Введите подстроку для её поиска:" << '\n';

cin >> Substring;

int Len = str.length();

int LenSub = Substring.length();

string str2;

bool flag=false;

for (int i = 0; i < Len - LenSub + 1; i++) {

int j = i, cnt = 0;

str2.clear();

while (cnt != LenSub) {

str2 += str[j];

j++;

cnt++;

}

if (str2 == Substring) {

for (int k = i; k <j; k++)

str[k] = toupper(str[k]);

flag = true;

}

}

if (flag == false)

cout << '\n' << "Данная подстрака не найдена" << '\n';

else

cout << '\n' << str << '\n';

}

void Task2(string& str) {

string Symbols = " ,/)(!;:?";

for (int i = 0; i < Symbols.length(); i++)

{

char Symbol = Symbols[i];

DeleteSymbols(str, Symbol);

}

DeletePoints(str);

int Len = str.length();

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

str[i] = tolower(str[i]);

}

void Task3(string& str) {

cout << "Исходная строка:" << '\n';

cout << str << '\n';

cout << "Слова из строки, не содержащие цифры:" << '\n';

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

DeleteNumbers(str, i);

}

cout << '\n';

}

void Task4(string& str) {

cout << "Исходная строчка:" << '\n';

cout << str << '\n';

cout << "Строчка со словами с заглавной буквы:" << '\n';

str[0] = tolower(str[0]);

UppercaseLetters(str);

}

void PrintMenu4() {

system("cls");

cout << "Что выхотите сделать?" << '\n';

cout << "1. Заполнить строку" << '\n';

cout << "2. Отформатировать строку" << '\n';

cout << "3. Слова не содержащие цифры" << '\n';

cout << "4. Строка со словами с заглавной буквы" << '\n';

cout << "5. Найти подстроку" << '\n';

cout << "6. Выйти из программы" << '\n';

}

void Menu4(string& str) {

int Variant;

do {

PrintMenu4();

cin >> Variant;

switch (Variant) {

case 1:

char perm;

cout << "Ввести строку с файла или с клавиатуры(a или b)" << '\n';

cin >> perm;

while (perm != 'a' and perm != 'b') {

cout << "Неверно введён вид вывода, попробуйте снова: ";

cin >> perm;

}

if (perm == 'a')

StrInput(str);

else {

cout << "Введите строку:" << '\n';

getline(cin, str);

}

break;

case 2:

cout << "Исходная строка:" << '\n';

cout << str;

Task2(str);

cout << '\n' << "Отформатированная строка" << '\n';

cout << str << '\n';

break;

case 3:

Task3(str);

break;

case 4:

Task4(str);

break;

case 5:

LinearSearch(str);

break;

}

if (Variant != 6)

system("pause");

} while (Variant != 6);

}

int main() {

setlocale(0, "");

string str;

Menu4(str);

}