**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**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 2372 |  | Тубшинов В.Т. |
| 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

2.1) Удаляет лишние пробелы  
2.2) Удаляет лишние знаки препинания

2.3) Исправляет регистр букв

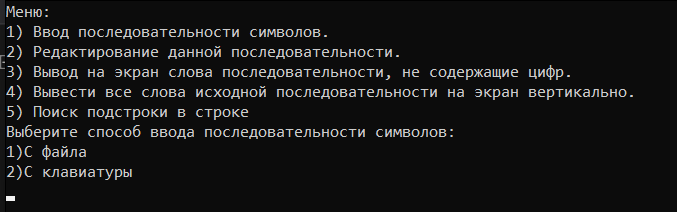
3) Вывести на экран слова последовательности, не содержащие цифр

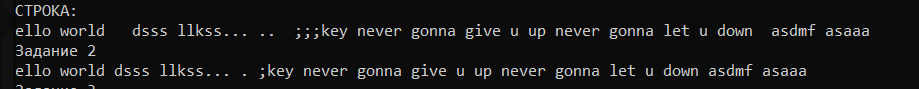
4) Вывести на экран ту же последовательность, удалив из всех слов заданный набор букв и (или) цифр

**Выполнение работы.**

Код программы представлен в приложении А.

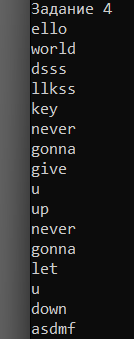
1. При запуске программы всплывает меню с заданиями. Первое задание ввод последовательности с клавиатуры, либо из файла.



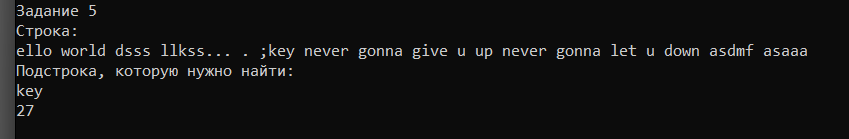
1. При выборе одного из двух вариантов ввода последовательности далее выполняется удаление лишних символов.
2. Далее, нужно было вывести на экран слова, не содержащие цифр.



1. Далее, нужно было вывести все слова вертикально.



1. И последнее задание, нахождение подстроки в строке. Нужно ввести строку, которую найдет программа и выведет индекс первого элемента строки.



**Выводы.**

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

Приложение А

рабочий код

#include <iostream>

#include <cstring>

#include <string>

#include <fstream>

using namespace std;

void delChar(char ch, char \*arr, int index)

{

for (int i = index; i < strlen(arr); i++)

{

arr[i] = arr[i + 1];

}

}

bool isHere(char ch, char \*arr)

{

int f = 0;

for (int i = 0; i < strlen(arr); i++)

{

if (ch == arr[i])

{

return true;

}

}

return false;

}

int main()

{

setlocale(0, " ");

setlocale(LC\_ALL, "ru");

char S[700];

char text[50][10];

char let[] = "QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm";

char symbols[] = { ' ',',','.',':','"', '?', '!', '-', '(', ')', ';', '\0' };

char numbers[] = "1234567890";

char tempChar[20];

int count = 0;

string path;

string str;

ifstream fin;

int answer;

for (int i = 0; i < 50; i++)

{

text[i][0] = '\0';

}

cout << "Меню: " << endl;

cout << "1) Ввод последовательности символов.\n2) Редактирование данной последовательности. \n3) Вывод на экран слова последовательности, не содержащие цифр. \n4) Вывести все слова исходной последовательности на экран вертикально. \n5) Поиск подстроки в строке \n";

cout << "Выберите способ ввода последовательности символов:\n1)С файла\n2)С клавиатуры" << endl;

cin >> answer;

switch (answer)

{

case 1:

cout << "Введите путь до файла:" << endl;

cin >> path;

fin.open(path);

if (fin.is\_open())

{

string str1;

while (!fin.eof())

{

getline(fin, str1);

str += str1;

}

}

fin.close();

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

{

S[i] = (char)str[i];

}

S[str.length()] = '\0';

break;

case 2:

cout << "Введите последовательность символов(последовательность должна заканчиваться на enter)" << endl;

cin.getline(S, 999, '\n');

cin.getline(S, 999, '\n');

delChar(S[0], S, 0);

break;

default:

break;

}

cout << "СТРОКА:\n" << S << endl;

//Задание 2

cout << "Задание 2" << endl;

for (int i = 0; i < strlen(S); i++)

{

for (int j = 0; j < (strlen(let) - 26); j++)

{

if (i != 0 && S[i] == let[j])

{

S[i] = let[j + 26];

}

}

}

for (int i = 0; i < strlen(S) - 1; i++)

{

if (isHere(S[i], symbols) && isHere(S[i + 1], symbols) && S[i] == S[i + 1])

{

if (S[i] == '.' && S[i + 1] == '.' && S[i + 2] == '.')

{

i += 2;

}

else

{

delChar(S[i], S, i);

i--;

}

}

}

cout << S << endl;

//Задание 3

cout << "Задание 3" << endl;

count = 0;

for (int i = 0, j = 0; i < strlen(S); i++, j++)

{

tempChar[j] = S[i];

if (isHere(S[i], symbols))

{

tempChar[j] = '\0';

for (int k = 0; k < strlen(tempChar); k++)

{

if (isHere(tempChar[k], symbols))

{

delChar(tempChar[k], tempChar, k);

}

}

int f = 1;

j = -1;

for (int k = 0; k < strlen(tempChar); k++)

{

if (strlen(tempChar) > 0)

{

text[count][k] = tempChar[k];

text[count][k + 1] = '\0';

if (isHere(tempChar[k], numbers))

{

f = 0;

}

}

}

if (f && strlen(tempChar) > 0)

{

cout << tempChar << " ";

}

count += 1;

}

}

cout << endl;

//Задание 4

cout << "Задание 4" << endl;

for (int i = 0; i < 50; i++)

{

if (strlen(text[i]) > 0)

{

cout << text[i] << endl;

}

}

//Задание 5

char F[20];

cout << "Задание 5" << endl;

cout << "Строка: " << endl;

cout << S << endl;

cout << "Подстрока, которую нужно найти: " << endl;

cin >> F;

unsigned indices[30];

for (int i = 0; i < 30; i++)

{

indices[i] = 0;

}

for (int i = 0, j = 0, k = 0; i < strlen(S); i++)

{

if (F[j] == S[i])

{

j++;

if (j == strlen(F))

{

indices[k] = i - strlen(F);

j = 0;

k++;

}

}

else

{

j = 0;

}

}

count = 0;

for (int i = 0; indices[i] != 0; i++)

{

count += 1;

cout << indices[i] << " ";

}

if (count == 0)

{

cout << "Такой подстроки не найдено";

}

cout << endl;

}