**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 |  | Астафьева У.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

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 пользователю выводится меню доступных команд и ожидается ввод команды с клавиатуры.

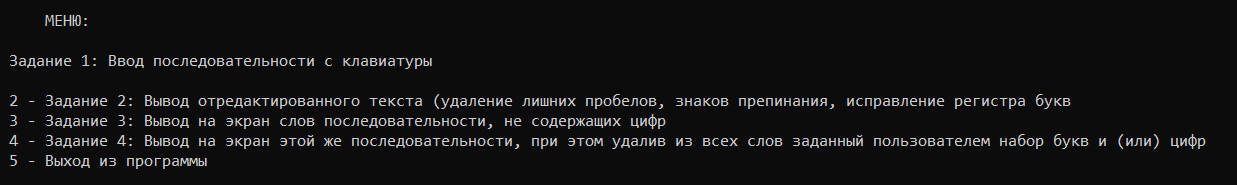


Рисунок . – Запуск программы, меню

1. Далее пользователю сразу предлагается ввести последовательность с клавиатуры

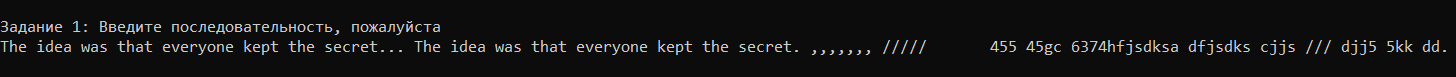


Рисунок 2. Задача 1 - Ввод пользователем последовательности с клавиатуры

1. Следующий шаг зависит от введенной команды, если пользователь ввёл:

3.1. “2”, то выведется отредактированная последовательность с удалёнными пробелами, знаками препинания, изменённым регистром.

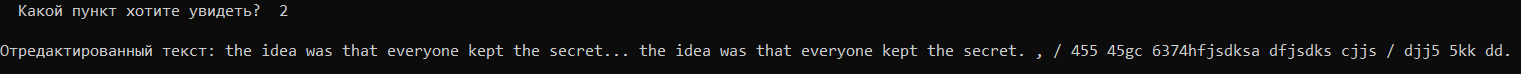


Рисунок . Задача 2 - Вывод отредактированного текста

3.2. “3”, то выведутся слова последовательности, в которых нет цифр. Так же – исходный текст для сравнения.

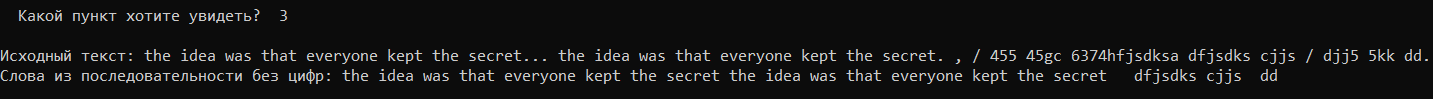


Рисунок 4. Задача 3 – Слова последовательности без цифр

3.3. “4”, то сначала пользователю предлагается ввести набор букв и (или) цифр, который необходимо удалить из последовательности. Далее выводится отредактированная последовательность.

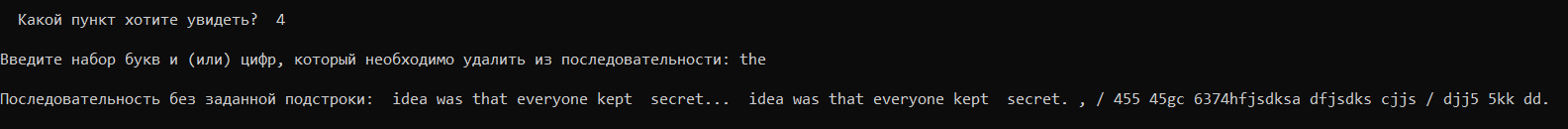


Рисунок 5. Задача 4 – Последовательность без введённой подстроки

* 1. “0, 1”, от “6” до любого другого числа, то пользователю предлагается ввести другое число, так как команды с таким номером нет.

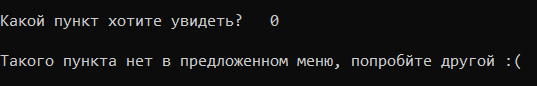


Рисунок 6 – Вывод пункта, не предложенного в меню

* 1. “5”, то происходит выход из программы.

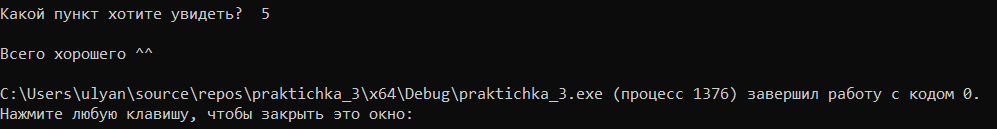
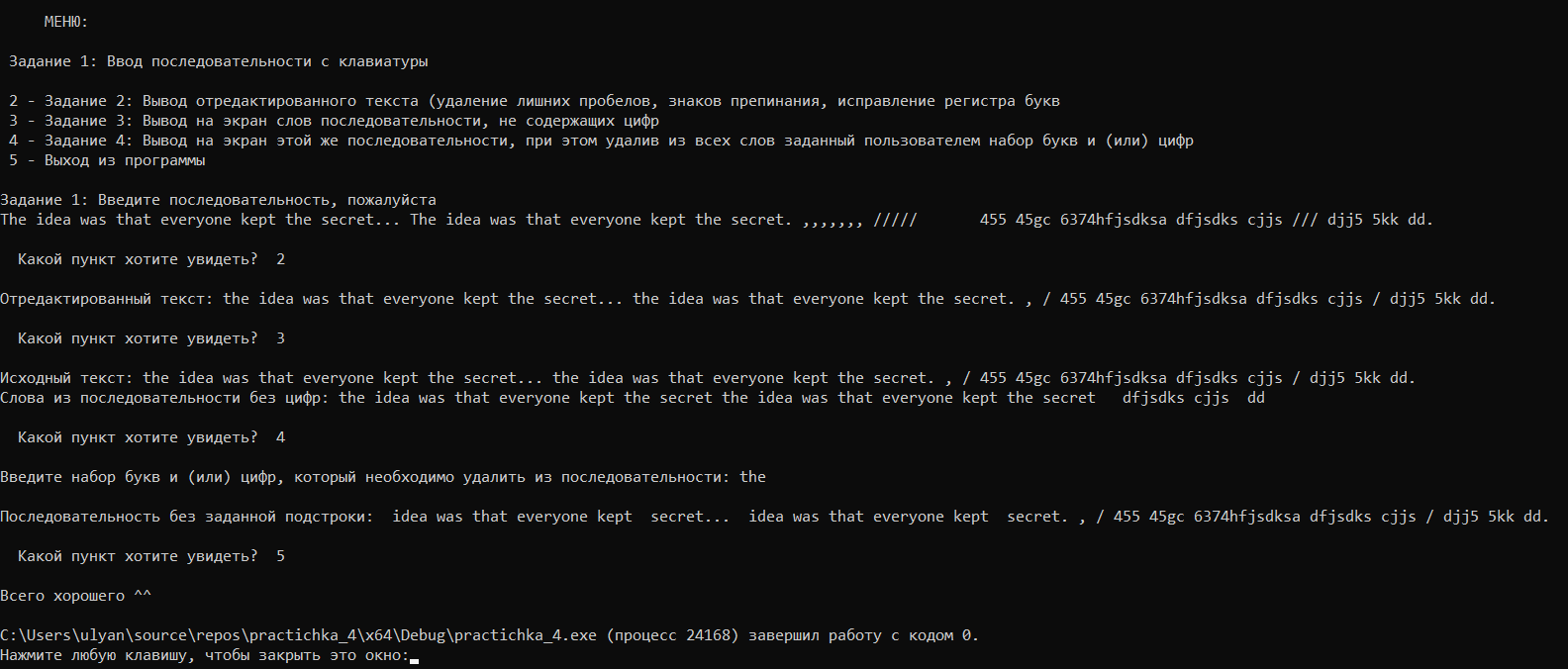


Рисунок 7 – Выход из программы

**Общий результат кода с примерами:**



**Выводы.**

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

Приложение А

рабочий код

#include <iostream>

#include <string>

using namespace std;

void delete\_simvols(string &text, char simvol, int &n)

{

n = text.length();

int index = text.find(simvol);

if (index > -1)

{

for (int i = index; i < n; i++)

{

int count = 0;

if (text[i] == simvol)

{

while (text[i + count] == simvol)

{

++count;

--n;

}

if (count > 0)

text = text.erase(i + 1, count - 1);

}

}

}

}

void checking\_numbers(string text, string clearWords, int &i, bool flag\_number)

{

int n = text.length();

char si = '...';

while (((text[i] != ' ') && (text[i] != ',') && (text[i] != '!') && (text[i] != '?') && (text[i] != ';') && (text[i] != ':') && (text[i] != ')') && (text[i] != si) && (text[i] != '/')) && (i != n))

{

if (text[i] >= 48 && text[i] <= 57)

flag\_number = true;

clearWords += text[i];

++i;

}

if (flag\_number == false)

cout << clearWords << ' ';

}

int main()

{

setlocale(LC\_ALL, "Russian");

bool flag = true;

int point = 0;

cout << "\n МЕНЮ:\n\n Задание 1: Ввод последовательности с клавиатуры\n\n" \

" 2 - Задание 2: Вывод отредактированного текста (удаление лишних пробелов, знаков препинания, исправление регистра букв\n" \

" 3 - Задание 3: Вывод на экран слов последовательности, не содержащих цифр\n" \

" 4 - Задание 4: Вывод на экран этой же последовательности, при этом удалив из всех слов заданный пользователем набор букв и (или) цифр\n" \

" 5 - Выход из программы\n\n";

cout << "Задание 1: Введите последовательность, пожалуйста\n";

string text = {};

getline(cin, text);

while (flag)

{

cout << "\n Какой пункт хотите увидеть? ";

cin >> point;

switch (point)

{

case 2:

{

int n = text.length();

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

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

{

char simvol = simvols[i];

delete\_simvols(text, simvol, n);

}

char simvol = '.';

int index = text.find(simvol);

for (int i = index; i < n; i++)

{

int count\_points = 0;

int count\_threetochie = 0;

while (text[i + count\_points] == simvol)

++count\_points;

count\_threetochie = count\_points / 3;

if (count\_threetochie != 0)

{

text = text.erase(i + 3, count\_points - 3);

n -= count\_points - 3;

i += 3;

}

if (count\_points == 2)

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

}

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

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

cout << "\nОтредактированный текст: " << text;

cout << endl;

break;

}

case 3:

{

int n = text.length();

cout << "\nИсходный текст: " << text;

cout << "\nСлова из последовательности без цифр: ";

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

{

string clearWords{};

bool flag\_number = false;

if (i == 0)

checking\_numbers(text,clearWords, i, flag\_number);

if (text[i] == ' ')

{

++i;

checking\_numbers(text,clearWords, i, flag\_number);

--i;

}

}

cout << endl;

break;

}

case 4:

{

int n = text.length();

string mini\_text = "";

cout << "\nВведите набор букв и (или) цифр, который необходимо удалить из последовательности: ";

cin >> mini\_text;

int index = -1;

while ((index = text.find(mini\_text)) != -1)

{

text = text.erase(index, mini\_text.length());

n -= mini\_text.length();

}

cout << "\nПоследовательность без заданной подстроки: " << text;

cout << endl;

break;

}

case 5:

{

cout << "\nВсего хорошего ^^\n";

flag = false;

break;

}

default:

cout << "\nТакого пункта нет в предложенном меню, попробуйте другой :(\n";

}

}

return 0;

}