

Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»
Институт компьютерных наук и технологий
Высшая школа киберфизических систем и управления

УДК 004.421

УТВЕРЖДАЮ

« ____ » _____ г.

ОТЧЕТ

по дисциплине «Теория и технологии программирования»

Лабораторная работа №10

Выполнил:

студент гр. 3532703/00001

_____ Д.П. Кимельфельд

подпись, дата

Санкт-Петербург 2021 г.

1. Цель работы

Цель задания – изучить методы работы с классом CString.

2. Задание

Вариант № 13.

Необходимо, чтобы в работе с текстом были представлены следующие команды, в соответствии с моим вариантом:

1. Добавление в конец.
2. Замена всех первых строчных букв слов прописными.
3. Вставка текста из файла.

3. Текст программы

```
#include <fstream>
#include <iostream>
#include <Windows.h>
#include <vector>
#include <string>
#include <conio.h>
using namespace std;

enum action { Add_to_the_end = 1, Replace_all_first_lowercase_letters = 2, Insert_text_from_file
= 3, Exit = 5, Text_mode = 4 };

bool checkSubString(string s, string s1, int g) { // s - строка с точкой , s1 подстрока
    int sizeSubstring = size(s1);
    if (g < 5 && sizeSubstring >= 4) {
        g = 3;
        while (s[g] == s1[g]) {
            --g;
            if (g == -1) break;
        }
        if (g == -1) return 1;
        else return 0;
    }
    if (5 <= g && g <= 11 && sizeSubstring >= 5) {
        g = 4;
        while (s[g] == s1[g]) {
            --g;
            if (g == -1) break;
        }
        if (g == -1) return 1;
        else return 0;
    }
    if (g == 12 && sizeSubstring >= 7) {
        g = 6;
        while (s[g] == s1[g]) {
            --g;
            if (g == -1) break;
        }
        if (g == -1) return 1;
        else return 0;
    }
    return 0;
}

bool checkFile(string path) {
    int sizePath = size(path);
    char c;
    int sizeSubPath = 0, save = 0;
    char sym[] = { '*', '<', '?', '>', '"', '|' };
    for (int i = 0; i < sizePath; i++) {
        for (int j = 0; j < 6; j++)
            if (path[i] == sym[j]) return 1;
    }
    //Уменьшаем регистр
    for (int i = 0; i < sizePath; i++) {
        c = path[i];
        path[i] = tolower(c);
    }
    sizePath = size(path);
    //Удаляем пробелы
    for (int i = sizePath - 1; i >= 0; --i) {
        if (path[i] == ' ') {
            path.erase(i, 1);
        }
    }
}
```

```

    }
}
sizePath = size(path);
int j = 0, g = 0;
for (int i = sizePath - 1; i >= 0; --i) {
    if (path[i] == '/') {
        j++;
    }
    if (path[i] == '\\') {
        g++;
    }
}
//
string delim;
if (j > g) {
    delim = "/";
}
else delim = "\\";
//
vector<string> arr;
int n = 0, next, delta = 1;
// Деление от слешей
string reserved[] = {
"con", "aux", "clock$", "com1", "com2", "com3", "com4", "lpt1", "lpt2", "lpt3", "nul", "prn", "com" };
string reserved1[] = { "con.", "aux.", "nul.", "prn.", "com.", "com1.",
"com2.", "com3.", "com4.", "lpt1.", "lpt2.", "lpt3.", "clock$." };
int l = 1;
// Отделение на подстроки
while ((next = path.find(delim, n)) != string::npos) {
    string tmp = path.substr(n, next - n);
    arr.push_back(path.substr(n, next - n));
    n = next + delta;
    l++;
}
//Проверка после последнего слеша
string tmp = path.substr(n);
arr.push_back(path.substr(n));
//
for (int k = 0; k < l; k++) {
    for (int s = 0; s < 13; s++) {
        if (reserved[s] == arr[k]) return 1;
        if (reserved1[s] == arr[k] || checkSubString(reserved1[s], arr[k], s)) return
1;
    }
}
return 0;
}
void outputVector(vector<string> a) {
    for (int i = 0; i < a.size(); i++) {
        cout << i+1 << ". " << a[i] << endl;
    }
}
void toAdd(vector<string> &text, int n) {
    string addStr;
    cout << "Enter the addition for end: " << endl;
    getline(cin, addStr);
    //
    if (n >= 1 && text.size() > 1 && n <= text.size() || (text.size() == 1 && n == 1))
text[n-1] += addStr;
    //if (n >= 1 && text.size() > 1 && n < text.size()) text.push_back(text[n-1]); // size =
2,3,4,5 n = 1,2,3,4,5
    //else if (text.size() == 1 || n <= 0 || n >= text.size()) cout << endl << "Sorry..But
it is not impossibly..." << endl;
    // (n < 1 or size <= 1 ) n =
0, size = 1,0 | -1,-2

```

```

        //system("pause");
    }
    int eqv(char a) {
        if (a == '9') return 9;
        else if (a == '8') return 8;
        else if (a == '7') return 7;
        else if (a == '6') return 6;
        else if (a == '5') return 5;
        else if (a == '4') return 4;
        else if (a == '3') return 3;
        else if (a == '2') return 2;
        else if (a == '1') return 1;
        else if (a == '0') return 0;
        else return 666;
    }
    int Mypow(int a, int b) {
        int res = 1;
        if (b == 0) return res;
        for (int i = b; i > 0; i--) res = res * a;
        return a;
    }
    bool my_isdigit(char ch)
    {
        if (ch == '1') return true;
        else if (ch == '2') return 1;
        else if (ch == '3') return 1;
        else if (ch == '4') return 1;
        else if (ch == '5') return 1;
        else if (ch == '6') return 1;
        else if (ch == '0') return 1;
        else if (ch == '7') return 1;
        else if (ch == '8') return 1;
        else if (ch == '9') return 1;
        else return 0;
    }
    int checkInput(string s, int low, int high) {
        int n = 0;
        for (int i = s.size() - 1; i >= 0; --i) {
            if (s[i] == ' ') {
                s.erase(i, 1);
            }
        }
        for (int i = s.size() - 1, k = 0; i >= 0; i--) {
            if (my_isdigit(s[i])) {
                n += Mypow(10, k)*eqv(s[i]);
                k++;
            }
            else {
                if ((s[i] == ')') || s[i] == '.') && (i == s.size() - 1)) {
                    continue;
                }
                else {
                    return high + 1;
                }
            }
        }
        return n;
    }
}

void toInputTxt(vector<string> &text, int txt) { // нахуя изменить все начиная с 183 строки
    string path = "example.txt";
    int selection;

```

```

string s;
cout << "Do u wanna input way to file?" << endl << "1.Yes" << endl << "2.No" << endl;
getline(cin, s);
selection = checkInput(s, 1, 2);
switch (selection) {
case 1:
    do {
        cout << "Enter the way to file:" << endl;
        cin.clear();
        getline(cin, path);
    } while (checkFile(path));

case 2: {
    ifstream file(path);
    int n;
    if (!file) {
        cout << "File does not exist..." << endl;
        system("pause");
        break;
    }
    else {
        cout << "File open!" << endl;
        system("pause");
        int number = 0;
        if (text.size() != 0) { // txt - порядковый номер строки, начиная с 0
            do {
                cout << "Enter after which line to insert the addition" <<
endl;

                string s;
                getline(cin, s);
                number = checkInput(s, 1, text.size());
                /*if (number <= 0 || number > text.size()) {
                    cout << "Error! Number does not exist.///." << endl;
                    system("pause");
                    break;
                }*/
                } while (number > ( text.size() + 1 ) || number <= 0);
            }
            else number = 1;
            number++;
            char end = '0';
            int k = 0;
            do {
                string j;
                getline(file, j);
                if (text.size() >= number + k) text.insert(text.begin() + number + k -
1, j);

                else text.push_back(j);
                k++;
                end = file.peek();
            } while (end != EOF);
        }
        break;
    }
default: {
    cout << "Error! Action does not exist.!. " << endl;
    system("pause");
}

    // cin.clear();
}
}

```

```

void toUpper(vector<string> &text,int n) {

    for (int i = n - 1; i < n ; i++) {                // i - Строка
        int save = 0;
        for (int j = 0; j < text[i].size() ; j++) {    // j - Символ в i-ой строке
            if (!(isalpha(text[i][j]))) save = j;
            if (save != 0) {
                if (isalpha(text[i][j]) && islower(text[i][j]) && (j - save == 1 ))
text[i][j] = toupper(text[i][j]);
            }
            else
                if (isalpha(text[i][j]) && islower(text[i][j]) && j == 0) text[i][j] =
toupper(text[i][j]);
        }
    }
}

void input(vector <string> &textInput, int &txt) {
    cout << "Enter text (Type 'Y' when finished typing keys): " << endl;
    string str;
    int symbol = 13;
    int n = 1 + textInput.size();
    bool end = false;
    do {
        if (symbol == 13) cout << n << ". ";
        symbol = _getch();
        if (symbol == 13) { // 13 - enter
            textInput.push_back(str);
            str = "";
            cout << endl;
            n++;
        }
        else if ((char)symbol == 'Y') {
            if (txt == 0) break;
            textInput.push_back(str);
            end = true;
        }
        else {
            str = str + (char)symbol;
            cout << (char)symbol;
        }
        txt++;
    } while (!end);
}

```

```

int main() {
    vector<string> textInput;
    int txt = 0;
    input(textInput,txt);
    cout << endl;
    while (1) {
        system("cls");
        cout << "Select the action: " << " | 1.Add to the end " << " | 2.Replace all first
lowercase letters of words with uppercase ones " << " | 3.Insert text from file " << " | 4.Text
mode | 5.Exit |" << endl;
        outputVector(textInput);
        cin.clear();
        string s;
        cout << endl << "Action: " << endl;
        int selection;
        do {
            getline(cin, s);
            selection = checkInput(s, 1, 5);

```

```

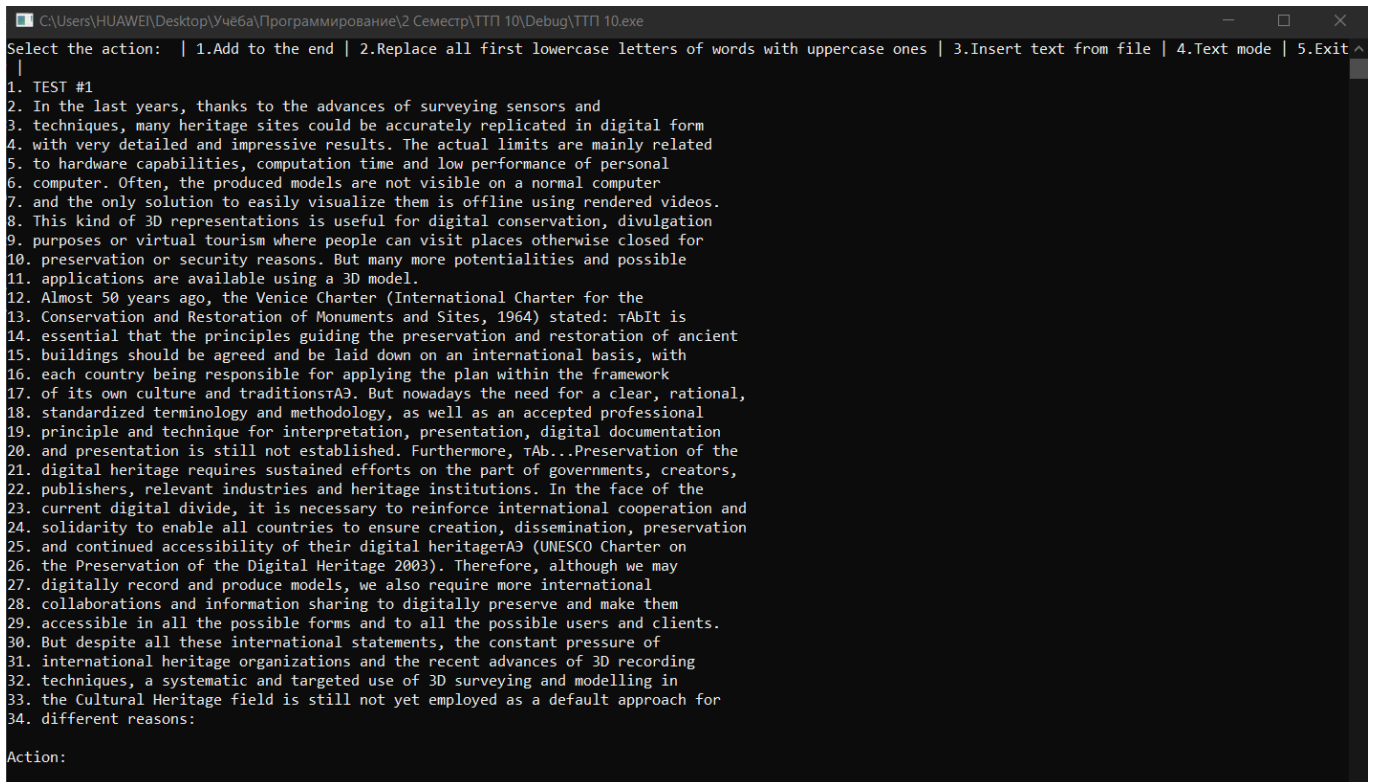
    } while (s == "");

    //
    switch (selection) {
    case Add_to_the_end: { // 1.Добавление в конец
        if (txt == 0) {
            cout << "There are no lines, you are put into text mode" << endl;
            input(textInput, txt);
            break;
        }
        cout << "Input number of line: ";
        string s;
        getline(cin, s);
        int number = checkInput(s, 1, textInput.size());
        if (number <= 0 || number > textInput.size()) {
            cout << "Error! Number does not exist.." << endl;
            system("pause");
            break;
        }
        toAdd(textInput, number);
        break;
    }
    case Replace_all_first_lowercase_letters: { // 2.Замена всех первых строчных букв
        // слов прописными.
        cout << "Input number of line: ";
        string s;
        getline(cin, s);
        int number = checkInput(s, 1, textInput.size());
        if (number <= 0 || number > textInput.size()) {
            cout << "Error! Number does not exist.." << endl;
            system("pause");
            break;
        }
        toUpper(textInput, number);
        break;
    }
    case Insert_text_from_file: { // 3.Вставка текста из файла
        toInputTxt(textInput, txt);
        break;
    }
    case Text_mode: { // 4.Режим ввода
        input(textInput, txt);
        break;
    }
    case Exit: {
        return 0;
    }
    default: {
        cout << "Error! Action does not exist.." << endl;
        system("pause");
    }
    }
}
return 0;
}

```


4. Пример работы программы

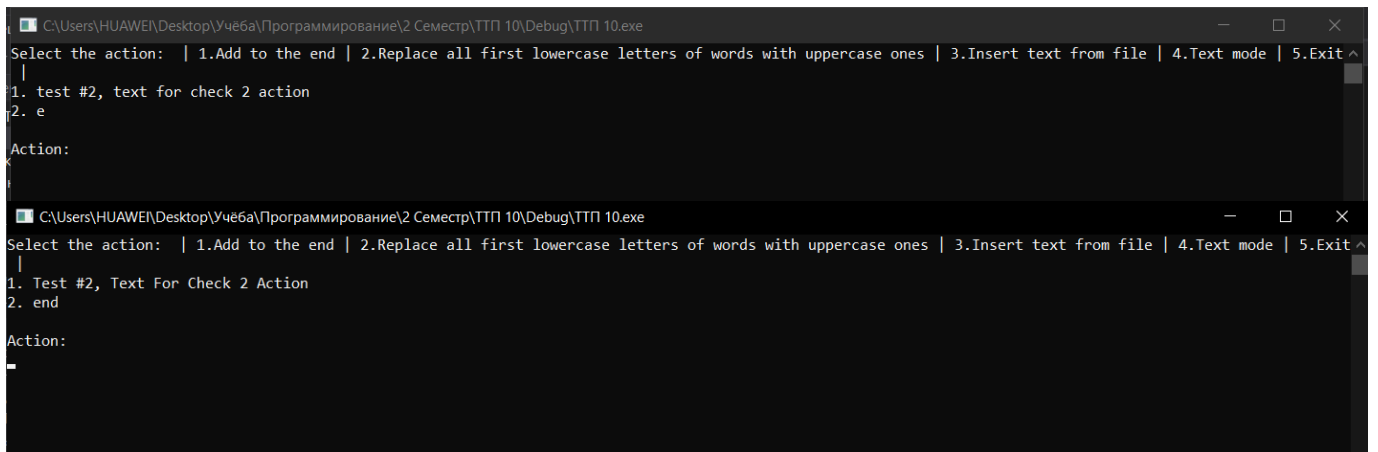
Пример №1



```
C:\Users\HUAWEI\Desktop\Учеба\Программирование\2 Семестр\ТПП 10\Debug\ТПП 10.exe
Select the action: | 1.Add to the end | 2.Replace all first lowercase letters of words with uppercase ones | 3.Insert text from file | 4.Text mode | 5.Exit ^
|
1. TEST #1
2. In the last years, thanks to the advances of surveying sensors and
3. techniques, many heritage sites could be accurately replicated in digital form
4. with very detailed and impressive results. The actual limits are mainly related
5. to hardware capabilities, computation time and low performance of personal
6. computer. Often, the produced models are not visible on a normal computer
7. and the only solution to easily visualize them is offline using rendered videos.
8. This kind of 3D representations is useful for digital conservation, divulgation
9. purposes or virtual tourism where people can visit places otherwise closed for
10. preservation or security reasons. But many more potentialities and possible
11. applications are available using a 3D model.
12. Almost 50 years ago, the Venice Charter (International Charter for the
13. Conservation and Restoration of Monuments and Sites, 1964) stated: тAbIt is
14. essential that the principles guiding the preservation and restoration of ancient
15. buildings should be agreed and be laid down on an international basis, with
16. each country being responsible for applying the plan within the framework
17. of its own culture and traditionsтАЭ. But nowadays the need for a clear, rational,
18. standardized terminology and methodology, as well as an accepted professional
19. principle and technique for interpretation, presentation, digital documentation
20. and presentation is still not established. Furthermore, тAb...Preservation of the
21. digital heritage requires sustained efforts on the part of governments, creators,
22. publishers, relevant industries and heritage institutions. In the face of the
23. current digital divide, it is necessary to reinforce international cooperation and
24. solidarity to enable all countries to ensure creation, dissemination, preservation
25. and continued accessibility of their digital heritageтАЭ (UNESCO Charter on
26. the Preservation of the Digital Heritage 2003). Therefore, although we may
27. digitally record and produce models, we also require more international
28. collaborations and information sharing to digitally preserve and make them
29. accessible in all the possible forms and to all the possible users and clients.
30. But despite all these international statements, the constant pressure of
31. international heritage organizations and the recent advances of 3D recording
32. techniques, a systematic and targeted use of 3D surveying and modelling in
33. the Cultural Heritage field is still not yet employed as a default approach for
34. different reasons:
Action:
```

Рисунок 1 – вставка текста из файла

Пример №2



```
C:\Users\HUAWEI\Desktop\Учеба\Программирование\2 Семестр\ТПП 10\Debug\ТПП 10.exe
Select the action: | 1.Add to the end | 2.Replace all first lowercase letters of words with uppercase ones | 3.Insert text from file | 4.Text mode | 5.Exit ^
|
1. test #2, text for check 2 action
2. e
Action:
x

C:\Users\HUAWEI\Desktop\Учеба\Программирование\2 Семестр\ТПП 10\Debug\ТПП 10.exe
Select the action: | 1.Add to the end | 2.Replace all first lowercase letters of words with uppercase ones | 3.Insert text from file | 4.Text mode | 5.Exit ^
|
1. Test #2, Text For Check 2 Action
2. end
Action:
-
```

Рисунок 2 – использование повышения регистра первых букв и вставка текста в конец

Выводы

В ходе выполнения программы я узнал, что язык C имеет достаточно много устаревших плохих команд, таких как `isdigit`, которая возвращает небулевы значения и может расценить `char` ('1') как число, так как функция основывается на сравнение кодового представления знака ASCII. Также следует отметить, что из-за этого мне пришлось написать свои команды, такие как `eqv(char a)` и `my_isdigit(char ch)`. Также я вспомнил, что математическая функция `pow` возвращает значения типа `double`, что было непозволительно, поскольку я работал с натуральными числами, поэтому мне также пришлось написать свою функцию возведения числа 10 в неотрицательную целую степень (`myPow(int a,int b)`). Также я узнал о методах классах `string`, таких как `insert ()`, `begin ()`, `size ()`, `push_back ()`.

С помощью них выполнена вся работа, в чем и состояла цель этой работы.