

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС «ІНСТИТУТ ПРИКЛАДНОГО  
СИСТЕМНОГО АНАЛІЗУ» НАЦІОНАЛЬНОГО ТЕХНІЧНОГО  
УНІВЕРСИТЕТУ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ  
ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 6

Варіант №18

3 дисципліни: Програмування та алгоритмічні мови

Роботу виконав:

Студент 1 курсу групи КА-95

Петренко Денис

Перевірив:

Гуськова В.Г.

Київ-2020

## 1. Завдання:

Реалізувати шаблон функції для визначення кількості елементів контейнера, що задовольняють певній умові. Умову визначити як шаблону функцію, що передається як параметр.

### 2.1 Лістинг програми:

```
#include<string.h>
#include <iostream>
#pragma warning(disable : 4996)
using namespace std;
class Phrase {
private:
    char* phrase;
public:
    operator char* () { return phrase; }
    Phrase() {
        phrase = new char[50];
        strcpy(phrase, "nophrase");
    };
    Phrase(char* phr) {
        phrase = new char[strlen(phr) + 1];
        strcpy(phrase, phr);
    };
    Phrase(Phrase& b) {
        phrase = new char[strlen(b.phrase) + 1];
        strcpy(phrase, b.phrase);
        //cout << "\n-----Constructor copy Phrase";
    };
    ~Phrase() { if (phrase) delete[] phrase; };
    friend ostream& operator<<(ostream& out, Phrase& v) {
        return out << "'" << v.phrase << "'" << ';' << endl;
    };
    friend istream& operator>>(istream& in, Phrase& v) {
        return in >> v.phrase;
    };
    Phrase& setphrase(char* phr) {
```

```

        delete[]phrase;
        phrase = new char[strlen(phr) + 1];
        strcpy(phrase, phr);
        return*this;
    };
    char* getphrase() { return phrase; };
    virtual void view() {
        cout << "" << phrase << "" << ';' << ' ' << endl;
    }
};

class Text {
private:
    char* title;
    int size_arr;
    Phrase* Arr;
public:
    Phrase& operator[] (int j) { return Arr[j]; };
    Text() {};
    Text(char* tit, int size, Phrase* A) :title(tit), size_arr(size), Arr(A) {};
    Text(char* tit, int size) : title(tit), size_arr(size) {
        Arr = new Phrase[size];
        for (int i = 0; i < size; i++) {
            cin >> Arr[i];
        }
    };
    ~Text() {};
    friend ostream& operator<<(ostream& out, Text& v) {
        return out << "" << v.title << "" << ',' << ' ' << '[' << v.size_arr << ']' << ';' <<
endl;
    };
    Text& settitle(char* tit) {
        delete[]title;
        title = new char[strlen(tit) + 1];
        strcpy(title, tit);
        return*this;
    };
    Text& setsize(int size) {
        size_arr = size;

```

```

        return*this;
    };
    Text& setArr(Phrase* A) {
        Arr = A;
        return*this;
    }
    char* gettitle() { return title; };
    int getsize() { return size_arr; }
    Phrase* getArr() { return Arr; };
};

template <class T>
bool umova(T a, int n) {
    if (strlen(a) <= n) {
        return true;
    }
    else { return false; }
}

template <class T>
int number(T a, int n) {
    int k = 0;
    for (int i = 0; i < n; i++) {
        if (umova(a[i], 5))
        {
            k++;
        }
    }
    return k;
};

int main() {
    char phr1[50], phr2[50], tit[50];
    //-----
    cout << "With inline type" << endl;
    int n;
    char** arr;
    cout << "Enter number of array: "; cin >> n;
    arr = new char* [n];
    for (int i = 0; i < n; i++) {

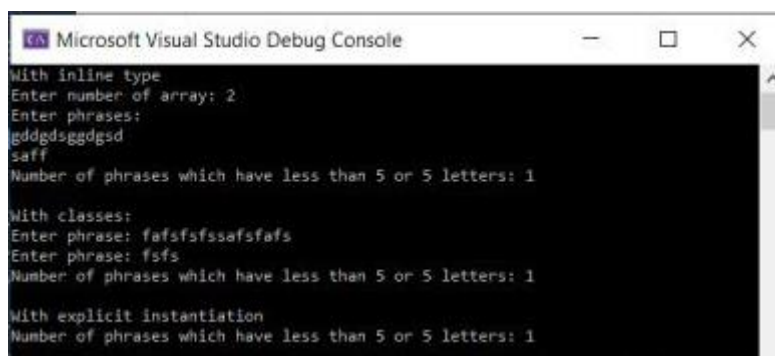
```

```

        arr[i] = new char[50];
    }
    cout << "Enter phrases:" << endl;
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    cout << "Number of phrases which have less than 5 or 5 letters: " <<
number(arr, n) << endl;
    //-----
    cout << "\nWith classes: " << endl;
    Phrase p1;
    cout << "Enter phrase: "; cin >> phr1;
    Phrase p2(phr1);
    Phrase p3 = p2;
    cout << "Enter phrase: "; cin >> phr2;
    p3.setphrase(phr2);
    Phrase* A = new Phrase[10];
    A[1] = p1; A[2] = p2; A[3] = p3;
    Text t1;
    t1.settitle(tit).setsize(10).setArr(A);
    cout << "Number of phrases which have less than 5 or 5 letters: " <<
number(A, 4) << endl;
    //-----
    cout << "\nWith explicit instantiation" << endl;
    cout << "Number of phrases which have less than 5 or 5 letters: " <<
number<Text>(t1, 4) << endl;
    return 0;
}

```

## 2.2 Результати:



```

Microsoft Visual Studio Debug Console
With inline type
Enter number of array: 2
Enter phrases:
gddgdsggdgsd
saff
Number of phrases which have less than 5 or 5 letters: 1

With classes:
Enter phrase: fafsfsfssafsfafs
Enter phrase: fsfs
Number of phrases which have less than 5 or 5 letters: 1

With explicit instantiation
Number of phrases which have less than 5 or 5 letters: 1

```

**Висновок:**

У даному практикумі було здобуто навички побудови власних шаблонів у мові C++.