**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА (САМАРСКИЙ УНИВЕРСИТЕТ)»**

ИНСТИТУТ ИНФОРМАТИКИ И КИБЕРНЕТИКИ

Кафедра программных систем

Дисциплина

**Объектно-ориентированное программирование**

**ОТЧЕТ**

по лабораторной работе № 1

**Объектно-ориентированное программирование обработки строк с  использованием функций стандартной библиотеки**

**языка С++ в стиле языка С.**

Вариант № 3

Студент: Боряков Н.С.

Группа: 6102-020302D

Преподаватель: Баландин А.В.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата:

Самара 2022

*ЗАДАНИЕ*

Ввести с клавиатуры произвольное абстрактное предложение длиной не более 80 символов, состоящее из абстрактных слов. Слова в предложении отделяются друг от друга как минимум одним символом «пробел». Предложение должно быть введено с помощью функции getchar(). Используя введённое предложение, построить новое предложение, в котором слова исходного предложения упорядочены по убыванию значений первых символов слов.

**ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ**

**Объектная модель предметной области**

**1. Информационный объект СЛОВО (**тип Сущность**)**

*Свойства:*

* Строка
* Длина\_Строки

*Процедуры*:

* Создание\_Слова
* Подсчет\_количества\_букв\_в\_результирующей\_строке

**2. Информационный объект СПИСОК\_СЛОВ (**тип Связь**)** *Свойства:*

* Список\_Слов
* Длина\_Списка\_Слов

*Процедуры:*

* Построение\_пустого\_Списка\_Слов
* Добавление\_Слова\_в\_Список\_Слов
* Построение\_Предложения\_по\_Списку\_Слов

**3. Информационный объект ПРЕДЛОЖЕНИЕ (**тип Сущность**)** *Свойства:*

* Символьная\_строка
* Список\_слов

*Процедуры:*

* Ввод\_Символьной\_строки\_и\_создание\_Исходного\_Предложения
* Вывод\_строки\_Исходного\_Предложения
* Создание\_пустого\_Результируещего\_Предложения
* Ввод\_Символьной\_строки\_в\_Результирующее\_Предложение\_копированием
* Сортировка\_Списков\_слов\_в\_Результирующем\_Предложение
* Вывод\_Строки\_Предложения

**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ**

**Спецификация классов**

**1. Класс СЛОВО**

*Свойства:*

o Строка – Массив символов с \0.

O Длина\_Строки – Целое число.

*Процедуры*:

o Создание\_слова – Метод.

o Подсчет\_количества\_букв\_в\_результирующей\_строке – Метод.

**2. Класс СПИСОК\_СЛОВ**

*Свойства:*

o Символьная\_строка - Массив символов.

o Список\_слов – Двумерный массив.

o Количество\_слов – Целое число.

*Процедуры:*

o Разделение\_заданной\_строки\_на\_лексемы – Метод.

**3. Класс ПРЕДЛОЖЕНИЕ**

*Свойства:*

o Символьная\_строка - Массив символов.

o Набор\_слов – Двумерный массив.

o Количество\_слов – Целое число.

*Процедуры:*

o Создание\_Предложения-Ввод\_Символьной\_строки – Конструктор без  параметров.

O Создание\_пустого\_предложения– Конструктор с параметром.

o Вывод\_Предложения\_посимвольно – Метод.

O Копирование\_заданной\_строки\_в\_Результирующее – Метод.

O Сортировка\_Списков\_слов\_в\_Результирующем\_Предложение

– Метод.

**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ**

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma once

#include <iostream>

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

#include <Windows.h>;

using namespace std;

class Sentence

{

private:

// предложение

char\*\* sent;

int count;

char\* str;

public:

Sentence(char\*\* oldsent, int counter)

{

count = counter;

//

sent = (char\*\*)malloc(count\* sizeof(char\*\*)); // создали память на новое предложение

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

{

sent[i] = (char\*)malloc(sizeof(char\*) \* strlen(oldsent[i]));

strcpy(sent[i], oldsent[i]);

}

}

Sentence()

{

int index = 0; // счётчик символов

str = (char\*)malloc(81); // выделение памяти для строки на 80 символов

char element; // определение пременной для символа

while ((element = getchar()) != '\n') // введенный элемент не равен символу перехода

{

if (index < 80) // заполнение массива символов при кол-ве элементов в нем меньше 80

{

str[index] = element; // добавление элемента в массив

index++; // обновление счётчика для перехода к новому элементу

}

}

str[index] = '\0'; // последний элемент

Divvvvv();

};

char\*\* Copy()

{

char\*\* \_sent = (char\*\*)malloc(sizeof(char\*\*) \* count);

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

{

\_sent[i] = (char\*)malloc(sizeof(char\*) \* strlen(sent[i]));

strcpy(\_sent[i], sent[i]);

}

return \_sent;

}

Sentence Sort()

{

char\*\* tempsent = Copy();

for (int j = count - 1; j > 0; j--)

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

// сравнение строк

if (strcmp(&tempsent[i][0], &tempsent[i+1][0]) < 0)

{

char\* flag = tempsent[i];

tempsent[i] = tempsent[i + 1];

tempsent[i + 1] = flag;

}

Sentence newSent(tempsent, count);

return newSent;

}

void Print()

{

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

{

printf("%s ", sent[i]);

}

}

private:

void Divvvvv()

{

sent = (char\*\*)malloc(sizeof(char\*\*) \* 81);

char\* word = strtok(str, " ");

int i = 0;

while (word != NULL)

{

sent[i] = word;

word = strtok(NULL, " "); //разделяем предложение на оставшиеся лексемы

i++;

count++;

}

sent = (char\*\*)realloc(sent, sizeof(char\*\*) \* count);

}

};

int main()

{

system("chcp 1251");//Использовать кириллицу для консольного ввода/вывода

Sentence s1;

printf("1ое исходное предложение:\n");

s1.Print();

Sentence s2 = s1.Sort();

printf("\n2ое исходное предложение:\n");

s2.Print();

system("pause"); //Задержать консоль на экране перед завершением программы

return 0;

}

**Спецификация алгоритма программы**

1. Создание Исходного\_Предложения;
2. Построение Списка\_Слов Исходного\_Предложения, деление его на лексемы;
3. Создание пустого Результатного\_Предложения;
4. Копирование в Результатное\_Предложение Исходное;
5. Сортировка Символьных\_Строк в Результатном\_Предложение;
6. Выводим Результатного\_Предложения на консоль;
7. Завершение работы.

Изображение выглядит как текст

Автоматически созданное описание

Пример работы программы