**Основное задание:**

Предложение. Хранит строку переменной длины представляющую собой предложение.

Методы: добавить слово, удалить слово, вставить слово, количество букв, количество слов, самое длинное слово,самое короткое слово, есть ли в предложении заданное слово, слово под заданным номером, равны ли два предложения.

**Выполнил:** Пазов В.С.

**Проверил:** Дзюба Е.А.

**Выполненное:**

9) Контейнеры и Итераторы.

**Код:**

// Предложение.

// Хранит строку переменной длины представляющую собой предложение.

// Методы: добавить слово, удалить слово, вставить слово, количество букв,

// количество слов, самое длинное слово, самое короткое слово,

// есть ли в предложении заданное слово, слово под заданным номером, равны ли два предложения.

// Выполненное:

// 9) Контейнеры и Итераторы.

#include "string.h"

#include "iostream"

#include <sstream>

#include <vector>

#include <array>

#include <string>

using namespace std;

class Sentence {

protected:

string sentence;

public:

Sentence() {

sentence = "Hello, world";

cout << "Конструктор без параметров: " << sentence << endl;

}

Sentence(string STR) {

sentence = STR;

cout << "Конструктор с параметрами: " << sentence << endl;

}

Sentence(const Sentence& other) {

sentence = other.sentence;

}

void Print() {

cout << this->sentence << endl;

}

};

int main() {

setlocale(LC\_ALL, "RUS");

Sentence withoutParams;

Sentence withParams("One1 Two2 Three333");

Sentence sentence1("Text");

Sentence sentence2("Text!\_/2");

Sentence copy\_sentence = sentence1;

cout << endl;

copy\_sentence.Print();

sentence2.Print();

vector<string> sentence3 = { "текст 1", "текст 2", "текст 3", "текст 4" };

vector<string>::iterator text;

cout << endl << "Вывод исходных предложений" << endl;

for (vector<string>::iterator text = sentence3.begin(); text != sentence3.end(); ++text) {

cout << \*(text) << endl;

}

cout << endl << "Добавление в начало" << endl << endl;

vector<string>::iterator Insert1=sentence3.begin();

sentence3.insert(Insert1, "Предложение добавлено");

for (vector<string>::iterator Insert1 = sentence3.begin(); Insert1 != sentence3.end(); ++Insert1) {

cout << \*(Insert1) << endl;

}

cout << endl << "Удаление начального предложения" << endl <<endl;

vector<string>::iterator Erase1 = sentence3.begin();

sentence3.erase(Erase1);

for (vector<string>::iterator Erase1 = sentence3.begin(); Erase1 != sentence3.end(); ++Erase1) {

cout << \*(Erase1) << endl;

}

cout << endl << "Вывод предложений через одно!" << endl << endl;

vector<string>::iterator Conclusion\_Through\_One = sentence3.begin();

for (vector<string>::iterator Conclusion\_Through\_One = sentence3.begin(); Conclusion\_Through\_One != sentence3.end(); Conclusion\_Through\_One+=2) {

cout << \*(Conclusion\_Through\_One) << endl;

}

cout << endl;

cout << endl <<"Завершение программы" << endl;

cout << endl;

return 0;

}

**Результат:**

