**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:**

5) Наследование.

**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:

// 5) Наследование.

#include "string.h"

#include "iostream"

#include <sstream>

#include <vector>

#include <array>

#include <stdio.h>

#include <conio.h>

#include <math.h>

#include <stdlib.h>

using namespace std;

class Sentence {

protected:

string sentence;

public:

Sentence() {

sentence = "Hello!, wo/rld!";

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

}

Sentence(string STR) {

sentence = STR;

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

}

Sentence(const Sentence& other) {

sentence = other.sentence;

}

virtual void Number\_Of\_Letters();

void Print() {

cout << this -> sentence << endl;

}

void Add\_Word() {

cout << endl << "Добавление слова:" << endl;

string word\_to\_add;

int begin\_end = 1;

cout << "Введите слово которое вы хотите добавить: " << endl;

cin >> word\_to\_add;

cout << "Добавить слово в начало текста - 1, если в конец - 2\n" << endl;

cin >> begin\_end;

switch(begin\_end) {

case 1:

word\_to\_add = word\_to\_add;

sentence = word\_to\_add + " " + sentence;

cout << sentence << endl;

break;

case 2:

sentence += " " + word\_to\_add;

cout << sentence << endl;

break;

default:

cout << "Неправильно введено число" << endl;

break;

}

}

void Delete\_Word() {

cout << endl << "Удаление слова: " << endl;

string new\_sentence = sentence;

char word\_to\_delete(' ');

int count\_spaces = 0;

int total\_spaces = 0;

cout << "Введите номер слова которое вы хотите удалить: " << endl;

cin >> total\_spaces;

total\_spaces - 2;

for (int i = 0; i < new\_sentence.length(); i++) {

if (new\_sentence[i] != word\_to\_delete) {

count\_spaces == count\_spaces;

} else {

count\_spaces++;

if (count\_spaces == total\_spaces) {

new\_sentence[i++];

}

while (new\_sentence[i] != word\_to\_delete && new\_sentence[i] != NULL) {

new\_sentence.erase(i, 1);

}

}

}

cout << "Исходное предложение: " << sentence << endl;

cout << "Предложение с удаленным словом: " << new\_sentence << endl;

}

void Trailing\_sign() {

char Tr\_sign('!');

int stop = 0;

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

{

if (sentence[i] == Tr\_sign)

{

stop = i;

i = sentence.length();

}

}

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

cout << sentence[i];

}

};

void Sentence::Number\_Of\_Letters() {

cout << endl << "Подсчет количества букв:" << endl;

int count\_letters = 0;

char Space(' ');

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

if (sentence[i] != Space) {

count\_letters++;

}

}

cout << "Исходное предложение: " << sentence << endl;

cout << "Количество букв: " << count\_letters << endl;

}

class Inheritance\_From\_Sentence : public Sentence {

char punctuation;

public:

Inheritance\_From\_Sentence(char pun = ('!')) {

punctuation = pun;

};

Inheritance\_From\_Sentence(Inheritance\_From\_Sentence& M) : Sentence(M) {

punctuation = M.punctuation;

};

void Trailing\_sign() {

switch(punctuation) {

case '.':

cout << "Белый " << sentence << endl;

break;

case '!':

cout << "Желтый " << sentence << endl;

break;

case '?':

cout << "Зеленый " << endl;

break;

default:

cout << "Неизвестный символ" << endl;

break;

}

}

};

class Do\_Have\_Letters : public Sentence {

public:

void Number\_Of\_Letters() override {

cout << "В предложении есть буквы!" << endl;

}

};

int main() {

setlocale(LC\_ALL, "RUS");

Sentence withoutParams;

Sentence withParams("One1 Two2 Three333");

Sentence ab = withParams;

cout << endl;

ab.Print();

withParams.Print();

cout << endl;

withParams.Number\_Of\_Letters();

cout << endl;

withoutParams.Trailing\_sign();

cout << endl;

Inheritance\_From\_Sentence nasl;

nasl.Trailing\_sign();

cout << endl;

Do\_Have\_Letters conc\_letters;

Sentence\* pointer = &conc\_letters;

pointer->Number\_Of\_Letters();

cout << endl;

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

cout << endl;

return 0;

};

**Результат:**

