МИНИСТЕРСТВО НАУКИ и высшего образования

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**Отчёт по лабораторной работе № 3**

**по учебной дисциплине «Конструкторы и деструкторы»**

**Тема: «Структура консольного приложения в C#»**

Выполнил(а) студент(ка)

специальности 09.02.03

Программирование в компьютерных

системах

IV курса группы 419/7

Косовский Семен

Андреевич

Преподаватель

Молькова Лолита Юрьевна

Санкт-Петербург,

2024

Цель работы:

Изучение возможности инициализации объектов класса с помощью конструкторов и уничтожение их с помощью деструкторов.

Задание:

Пользовательский класс MyString должен содержать необходимые элементыданные, которые создаются в динамической области памяти. • Конструкторы (без параметров, с параметрами, копирования) для создания строк: MyString (…); • Деструктор: ~MyString(); • Метод ввода исходной строки: set(); • Метод изменения исходной строки согласно варианту (исходная и измененная строка должны сохраняться в файле): update() ; • Метод вывода на экран: print(…); • Каждый вызов методов (в том числе конструкторов и деструктора) сопровождается выдачей соответствующего сообщения; Код методов – вне пространства определения класса. Написать демонстрационную программу, в которой показать использование объектов созданного класса  
Вариант 8.

**Ход работы:**

На рисунке 1 изображена вся работы программы

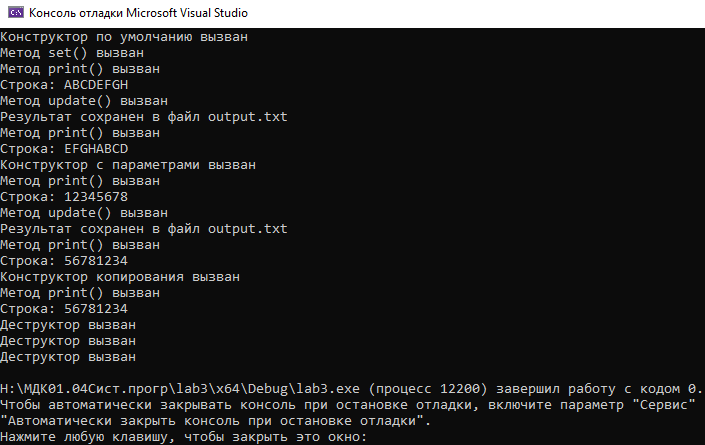


Рисунок 1 – Вся работы программы

На рисунке 2 изображен файл output.txt

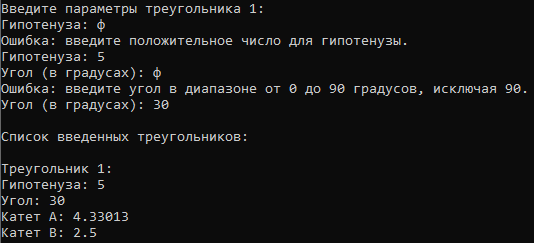


Рисунок 2 – файл output.txt

Код:

Заголовочный файл MyString.h  
#ifndef MYSTRING\_H

#define MYSTRING\_H

#include <iostream>

// Определение класса MyString

class MyString {

private:

char\* str; // Указатель на строку в динамической памяти

size\_t length; // Длина строки

public:

// Конструкторы

MyString(); // Конструктор по умолчанию

MyString(const char\* s); // Конструктор с параметрами

MyString(const MyString& other); // Конструктор копирования

// Деструктор

~MyString();

// Методы

void set(const char\* s); // Метод ввода строки

void update(); // Метод изменения строки согласно заданию

void print() const; // Метод вывода строки на экран

};

#endif // MYSTRING\_H

Файл реализации методов класса MyString.cpp  
#include "MyString.h"

#include <cstring>

#include <fstream>

#include <utility>

// Конструктор по умолчанию

MyString::MyString() : str(nullptr), length(0) {

std::cout << "Конструктор по умолчанию вызван" << std::endl;

}

// Конструктор с параметрами

MyString::MyString(const char\* s) {

std::cout << "Конструктор с параметрами вызван" << std::endl;

length = strlen(s);

str = new char[length + 1];

strcpy\_s(str, length + 1, s); // Используем безопасную функцию strcpy\_s

}

// Конструктор копирования

MyString::MyString(const MyString& other) {

std::cout << "Конструктор копирования вызван" << std::endl;

length = other.length;

str = new char[length + 1];

strcpy\_s(str, length + 1, other.str); // Используем безопасную функцию strcpy\_s

}

// Деструктор

MyString::~MyString() {

std::cout << "Деструктор вызван" << std::endl;

delete[] str;

}

// Метод для ввода строки

void MyString::set(const char\* s) {

std::cout << "Метод set() вызван" << std::endl;

delete[] str; // Освобождаем старую строку, если она была

length = strlen(s);

str = new char[length + 1];

strcpy\_s(str, length + 1, s); // Используем безопасную функцию strcpy\_s

}

// Остальная часть кода остаётся без изменений...

// Метод для изменения строки

void MyString::update() {

std::cout << "Метод update() вызван" << std::endl;

// Проверка, что длина строки кратна 4

if (length % 4 == 0 && length > 0) {

// Сохраняем исходную строку для записи в файл

std::string originalStr(str);

// Меняем первую половину строки с второй

size\_t half\_length = length / 2;

for (size\_t i = 0; i < half\_length; ++i) {

std::swap(str[i], str[half\_length + i]);

}

// Открываем файл для записи

std::ofstream outFile("output.txt");

if (outFile.is\_open()) {

outFile << "Исходная строка: " << originalStr << std::endl;

outFile << "Измененная строка: " << str << std::endl;

outFile.close();

std::cout << "Результат сохранен в файл output.txt" << std::endl;

}

else {

std::cerr << "Ошибка при открытии файла!" << std::endl;

}

}

else {

std::cout << "Длина строки не кратна 4, изменения не произведены." << std::endl;

}

}

// Метод для вывода строки на экран

void MyString::print() const {

std::cout << "Метод print() вызван" << std::endl;

if (str != nullptr) {

std::cout << "Строка: " << str << std::endl;

}

else {

std::cout << "Строка пуста" << std::endl;

}

}

3. Основной файл main.cpp  
#include "MyString.h"

int main() {

setlocale(LC\_ALL, "rus");

MyString s1; // Вызов конструктора по умолчанию

s1.set("ABCDEFGH"); // Ввод строки, кратной 4

s1.print(); // Вывод строки

s1.update(); // Изменение строки

s1.print(); // Вывод измененной строки

MyString s2("12345678"); // Вызов конструктора с параметрами

s2.print();

s2.update();

s2.print();

MyString s3 = s2; // Вызов конструктора копирования

s3.print();

return 0;

}

**MyString**

| **Название метода** | **Описание** |
| --- | --- |
| MyString() | Конструктор по умолчанию класса MyString. |
| MyString(const char\* s) | Конструктор с параметрами, принимает C-строку и сохраняет её. |
| MyString(const MyString& other) | Конструктор копирования, создает копию строки. |
| ~MyString() | Деструктор, освобождает динамическую память. |
| void set(const char\* s) | Метод ввода строки, устанавливает новую строку. |
| void update() | Изменяет строку согласно условию: если длина кратна 4, меняет первую половину строки со второй. |
| void print() const | Выводит строку на экран. |