

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	5
1.2 Описание выходных данных.....	5
2 МЕТОД РЕШЕНИЯ.....	6
3 ОПИСАНИЕ АЛГОРИТМОВ.....	7
3.1 Алгоритм конструктора класса muchacha.....	7
3.2 Алгоритм деструктора класса muchacha.....	7
3.3 Алгоритм функции main.....	7
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	9
5 КОД ПРОГРАММЫ.....	10
5.1 Файл main.cpp.....	10
5.2 Файл muchacha.cpp.....	10
5.3 Файл muchacha.h.....	11
6 ТЕСТИРОВАНИЕ.....	12
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	13

1 ПОСТАНОВКА ЗАДАЧИ

Сконструировать систему, которая демонстрирует последовательность выполнения этапов ЖЦ объекта, возможность динамического изменения состава элементов системы.

Для построения системы использовать один объект. У объекта нет свойств. Объект на этапе ЖЦ «создание» сообщает об отработке метода конструктора. На этапе ЖЦ «демонтаж» сообщает об отработке метода деструктора.

Алгоритм отработки системы.

1. Построение системы: создать объект.
2. Уничтожить объект.
3. Завершить работу систему.

1.1 Описание входных данных

Отсутствует.

1.2 Описание выходных данных

Первая строка, с первой позиции:

CONSTRUCTOR

Вторая строка, с первой позиции:

DESTRUCTOR

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект `gracias` класса `muchacha` предназначен для проверки работы конструктора и деструктора;
- объект стандартного потока вывода данных `cout`;
- библиотека `iostream`;
- оператор-функция `new`;
- оператор-функция `delete`.

Класс `muchacha`:

- функционал:
 - метод `muchacha` — конструктор для создания объекта класса;
 - метод `~muchacha` — деструктор для удаления объекта класса.

3 ОПИСАНИЕ АЛГОРИТМОВ

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

3.1 Алгоритм конструктора класса *tuchacha*

Функционал: Проверка работоспособности конструктора.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса *tuchacha*

№	Предикат	Действия	№ перехода
1		Вывод "CONSTRUCTOR"	2
2		Переход на новую строчку	Ø

3.2 Алгоритм деструктора класса *tuchacha*

Функционал: Проверка работоспособности деструктора.

Параметры: нет.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса *tuchacha*

№	Предикат	Действия	№ перехода
1		Вывод "DESTRUCTOR"	Ø

3.3 Алгоритм функции *main*

Функционал: Главная функция программы.

Параметры: нет.

Возвращаемое значение: Целое число, индикатор работоспособности программы.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Создание объекта <i>gracias</i> класса <i>michacha</i> с помощью оператора <i>new</i>	2
2		Удаление объекта <i>gracias</i> с помощью оператора <i>delete</i>	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-1.

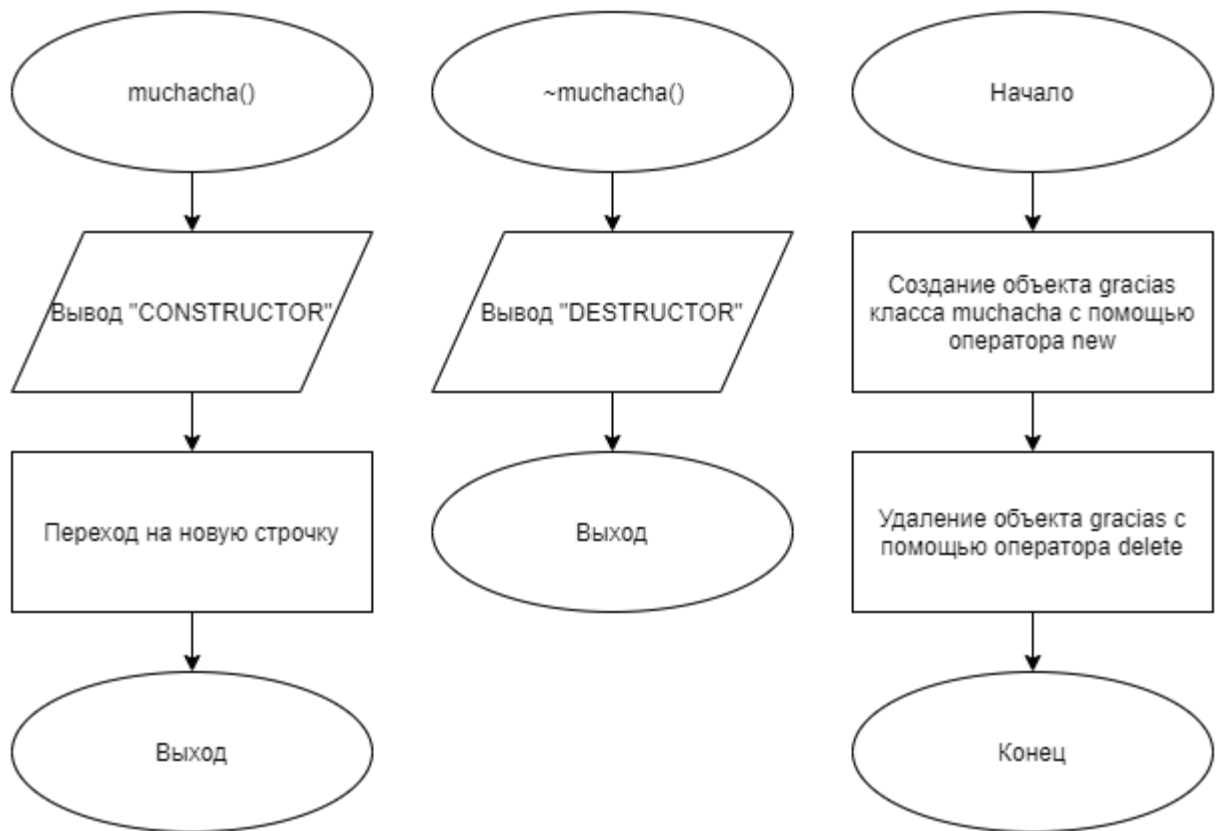


Рисунок 1 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include "muchacha.h"
#include <iostream>

int main()
{
    muchacha * gracias = new muchacha(); // muchacha - имя класса, gracias -
метод объекта
    delete(gracias); // удаление
    return(0);
}
```

5.2 Файл muchacha.cpp

Листинг 2 – muchacha.cpp

```
#include "muchacha.h" // подключение файла muchacha.h
#include <iostream>
using namespace std;

muchacha::muchacha() // перед двоеточием - класс, после - метод
{
    cout << "CONSTRUCTOR" << endl;
};

muchacha::~muchacha()
{
    cout << "DESTRUCTOR";
};
```


5.3 Файл muchacha.h

Листинг 3 – muchacha.h

```
#ifndef __MUCHACHA__H
#define __MUCHACHA__H

class muchacha
{
public:

muchacha(); // конструктор
~muchacha(); // деструктор

};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 4.

Таблица 4 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	CONSTRUCTOR DESTRUCTOR	CONSTRUCTOR DESTRUCTOR

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19 Единая система программной документации.
2. Методическое пособие студента для выполнения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avvora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).
3. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avvora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).
4. Шилдт Г. С++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2019. — 624 с.
5. Видео лекции по курсу «Объектно-ориентированное программирование» [Электронный ресурс]. АСО «Аврора».
6. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие /Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).