Здесь будет титульник, листай ниже

**СОДЕРЖАНИЕ**

[1 ПОСТАНОВКА ЗАДАЧИ 5](#__RefHeading___Toc1305_94810477)

[1.1 Описание входных данных 5](#__RefHeading___Toc1307_94810477)

[1.2 Описание выходных данных 5](#__RefHeading___Toc1309_94810477)

[2 МЕТОД РЕШЕНИЯ 6](#__RefHeading___Toc1311_94810477)

[3 ОПИСАНИЕ АЛГОРИТМОВ 7](#__RefHeading___Toc1313_94810477)

[3.1 Алгоритм конструктора класса MyClass 7](#__RefHeading___Toc1315_94810477)

[3.2 Алгоритм деструктора класса MyClass 7](#__RefHeading___Toc1317_94810477)

[3.3 Алгоритм функции main 7](#__RefHeading___Toc1319_94810477)

[4 БЛОК-СХЕМЫ АЛГОРИТМОВ 9](#__RefHeading___Toc1321_94810477)

[5 КОД ПРОГРАММЫ 10](#__RefHeading___Toc1323_94810477)

[5.1 Файл main.cpp 10](#__RefHeading___Toc1325_94810477)

[5.2 Файл MyClass.cpp 10](#__RefHeading___Toc1327_94810477)

[5.3 Файл MyClass.h 11](#__RefHeading___Toc1329_94810477)

[6 ТЕСТИРОВАНИЕ 12](#__RefHeading___Toc1331_94810477)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 13](#__RefHeading___Toc1333_94810477)

# 1 ПОСТАНОВКА ЗАДАЧИ

Создать объект, который сообщает об отработке конструктора и деструктора. У объекта нет свойств и функциональности.

Написать программу, которая:

* Создает объект с использованием оператора функции new.
* Уничтожить объект оператором функции delete.

## 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 используются:

    Оператор стандартного потока вывода(cout);

Объект obj класса MyClass:

* Поля/свойства - отсутсвуют;
* Методы;

Конструктор MyClass:

    Функционал -конструктор без параметров, создание объекта класса;

Деструктор MyClass:

    Функционал - удаление объекта класса;

Оператор функции new;

Оператор функции delete;

# 3 ОПИСАНИЕ АЛГОРИТМОВ

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

## Алгоритм конструктора класса MyClass

Функционал: Создание объекта класса.

Параметры: Отсутствуют.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса MyClass

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вывод на экран "Constructor", переход на новую строку | ∅ |

## Алгоритм деструктора класса MyClass

Функционал: Удаление объекта класса.

Параметры: Отсутствуют.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса MyClass

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вывод на экран "Destructor", переход на новую строку | ∅ |

## Алгоритм функции main

Функционал: Основной алгоритм программы.

Параметры: Отсутствуют.

Возвращаемое значение: Целочисленный - индикатор корректности выполнения программы..

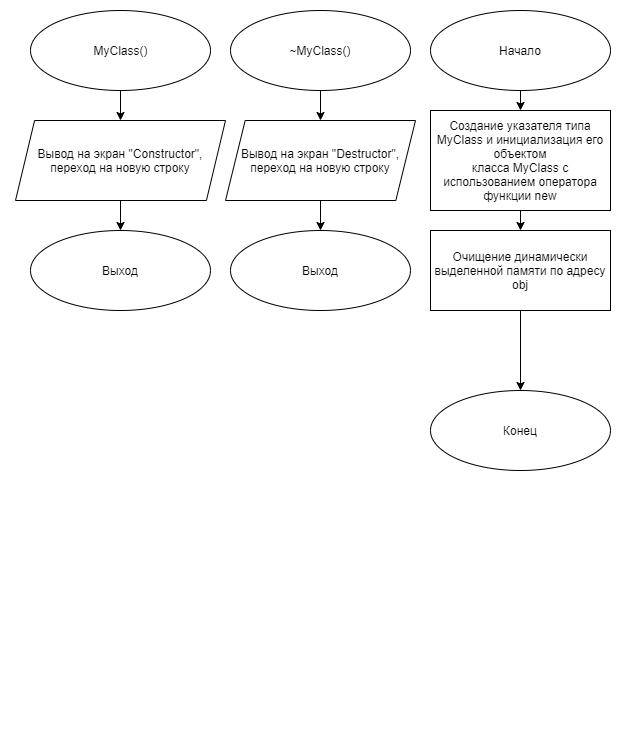
Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции main

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Создание указателя типа MyClass и инициализация его объектом  класса MyClass с использованием оператора функции new | 2 |
| 2 |  | Очищение динамически выделенной памяти по адресу obj | ∅ |

# 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.

## **Файл** **main.cpp**

Листинг 1 – main.cpp

|  |
| --- |
| #include <stdlib.h>  #include <stdio.h>  #include <iostream>  #include "MyClass.h"  using namespace std;  int main()  {  MyClass \*obj = new MyClass();  delete obj;  return(0);  } |

## **Файл** **MyClass.cpp**

Листинг 2 – MyClass.cpp

|  |
| --- |
| #include <stdlib.h>  #include <stdio.h>  #include <iostream>  #include "MyClass.h"  using namespace std;  MyClass::MyClass(){  cout << "Constructor\n";  }  MyClass::~MyClass(){  cout << "Destructor";  } |

## **Файл** **MyClass.h**

Листинг 3 – MyClass.h

|  |
| --- |
| #ifndef \_\_MyClass\_H\_\_  #define \_\_MyClass\_H\_\_  class MyClass{  public:  MyClass();  ~MyClass();  };  #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. Васильев А.Н. Объектно-ориентированное программирование на С++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.

2. Шилдт Г. С++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.

3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodichescoe\_posobie\_dlya\_laboratorny h\_rabot\_3.pdf (дата обращения 05.05.2021).

4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrora.ru/student/files/Prilozheniye\_k\_methodichke.pdf (дата обращения 05.05.2021).

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).