

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	6
1.2 Описание выходных данных.....	6
2 МЕТОД РЕШЕНИЯ.....	7
3 ОПИСАНИЕ АЛГОРИТМОВ.....	8
3.0 Алгоритм конструктора класса MyClass.....	8
3.1 Алгоритм деструктора класса MyClass.....	8
3.2 Алгоритм метода Print класса MyClass.....	9
3.3 Алгоритм метода Save класса MyClass.....	9
3.4 Алгоритм метода Print_Saved класса MyClass.....	10
3.5 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	18
5.0 Файл main.cpp.....	18
5.1 Файл MyClass.cpp.....	18
5.2 Файл MyClass.h.....	20
6 ТЕСТИРОВАНИЕ.....	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	22

1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В открытом доступе имеется указатель на массив целого типа.

Конструктору объекта передается целочисленный параметр. Параметр должен иметь значение больше 4. По значению параметра определяется размерность целочисленного массива и каждому элементу присваивается это же значение.

Объект имеет функциональность, по которой выводит содержимое целочисленного массива. Вывод производит последовательно, разделяя значения двумя пробелами.

Спроектировать систему, которая содержит два объекта. Для построения системы последовательно, с новых строк вводятся целочисленные значения. Если очередное введенное значение меньше или равно 4, то создание системы прекращается, выводится сообщение и программа завершает работу. Если очередное введенное значение больше 4, то посредством параметризованного конструктора создается первый объект, а затем, второй.

Далее система функционирует по алгоритму:

1.
2. Первому объекту присвоить второй объект.
3.
4. С первой строки вывести содержимое массива первого объекта.
5.
6. Со второй строки вывести содержимое массива второго объекта.

1.1 Описание входных данных

Первая строка:

«Целое число»

Вторая строка:

«Целое число»

Пример.

5

8

1.2 Описание выходных данных

Если система была построена, то в первой строке:

«Целое число» «Целое число» . . .

Во второй строке:

«Целое число» «Целое число» . . .

Если система не была построена, то в первой строке выводится некорректное значение и вопросительный знак:

«Целое число»?

Пример вывода.

5 5 5 5 5

8 8 8 8 8 8 8 8

2 МЕТОД РЕШЕНИЯ

Для решения используются

Объект стандартного ввода cin;

Объекты obj_a(l_mas), obj_b(l_mas) класса MyClass;

Объект стандартного вывода cout;

Объект new;

Объект delete;

Класс MyClass:

Методы:

Наименование - p_data

Тип - int*

Конструктор MyClass(int l_mas)

Функционал - объявление указателя на массив и его заполнение

Метод Print(int l_mas)

Функционал - выводит значение p_data

Модификатор доступа - public

Метод Save(int s_mas[], int l_mas)

Функционал - сохраняет значения p_data в массив поданный в параметрах метода

Модификатор доступа - public

Print_Saved(int s_mas[], int l_mas)

Функционал - выводит сохранённые значения

Модификатор доступа - public

Деструктор ~MyClass()

Функционал - удаление выделенной памяти под указатель p_data

3 ОПИСАНИЕ АЛГОРИТМОВ

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

3.0 Алгоритм конструктора класса MyClass

Функционал: Создание и заполнение массива.

Параметры: int, l_data, указатель размера массива.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса MyClass

№	Предикат	Действия	№ перехода
1	$l_data < 4$	создание указателя p_data	2
		Вывод параметра с знаком "?"	Ø
2	переменная цикла меньше параметра	Элемент указателя = параметру	2
			Ø

3.1 Алгоритм деструктора класса MyClass

Функционал: Удаление выделенной памяти.

Параметры: отсутствует.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса MyClass

№	Предикат	Действия	№ перехода
1	Указатель не пуст и это первый вызов деструктора	Удаление выделенной памяти под указатель	Ø

№	Предикат	Действия	№ перехода
			∅

3.2 Алгоритм метода Print класса MyClass

Функционал: Вывод p_data.

Параметры: int, l_data, указатель размера массива.

Возвращаемое значение: целый тип.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода Print класса MyClass

№	Предикат	Действия	№ перехода
1	Переменная цикла меньше параметра	Вывод элемента p_mas	2
			2
2	Это не последний элемент	Вывод " "	1
			∅

3.3 Алгоритм метода Save класса MyClass

Функционал: сохраняет в массив поданный в параметре значения p_data.

Параметры: int, s_mas[], массив для сохранения; int, l_data, указатель размера массива.

Возвращаемое значение: целый тип.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *Save* класса *MyClass*

№	Предикат	Действия	№ перехода
1	Переменная цикла меньше параметра(l_data)	s_mas = p_mas	1
			Ø

3.4 Алгоритм метода *Print_Saved* класса *MyClass*

Функционал: Выводит значения массива поданого в параметре.

Параметры: int, s_mas[], массив для вывода; int, l_data, указатель размера массива.

Возвращаемое значение: целый тип.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *Print_Saved* класса *MyClass*

№	Предикат	Действия	№ перехода
1	Переменная цикла меньше параметра(l_data)	Вывод элемента s_mas	2
			2
2	Это не последний элемент	Вывод " "	1
			Ø

3.5 Алгоритм функции *main*

Функционал: основная функция.

Параметры: отсутствуют.

Возвращаемое значение: целый тип, признак успешности выполнения.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление l_mas_a, l_mas_b, *s_mas	2
2		Ввод l_mas_a	3
3		Создание объекта obj_a класса MyClass с параметром l_mas_a	4
4	Указатель в объекте obj_a был создан		5
			Ø
5		Ввод l_mas_b	6
6		Создание объекта obj_b класса MyClass с параметром l_mas_b	7
7	Указатель в объекте obj_b был создан		8
			Ø
8		Создание указателя s_mas	9
9		Вызов метода Save(s_mas, l_mas_a)	10
10		obj_a = obj_b	11
11		Вызов метода Print_Saved(s_mas, l_mas_a) Переход на следующую строчку	12
12		Вызов метода Print(l_mas_b)	13
13		Удаление указателя s_data	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-6.

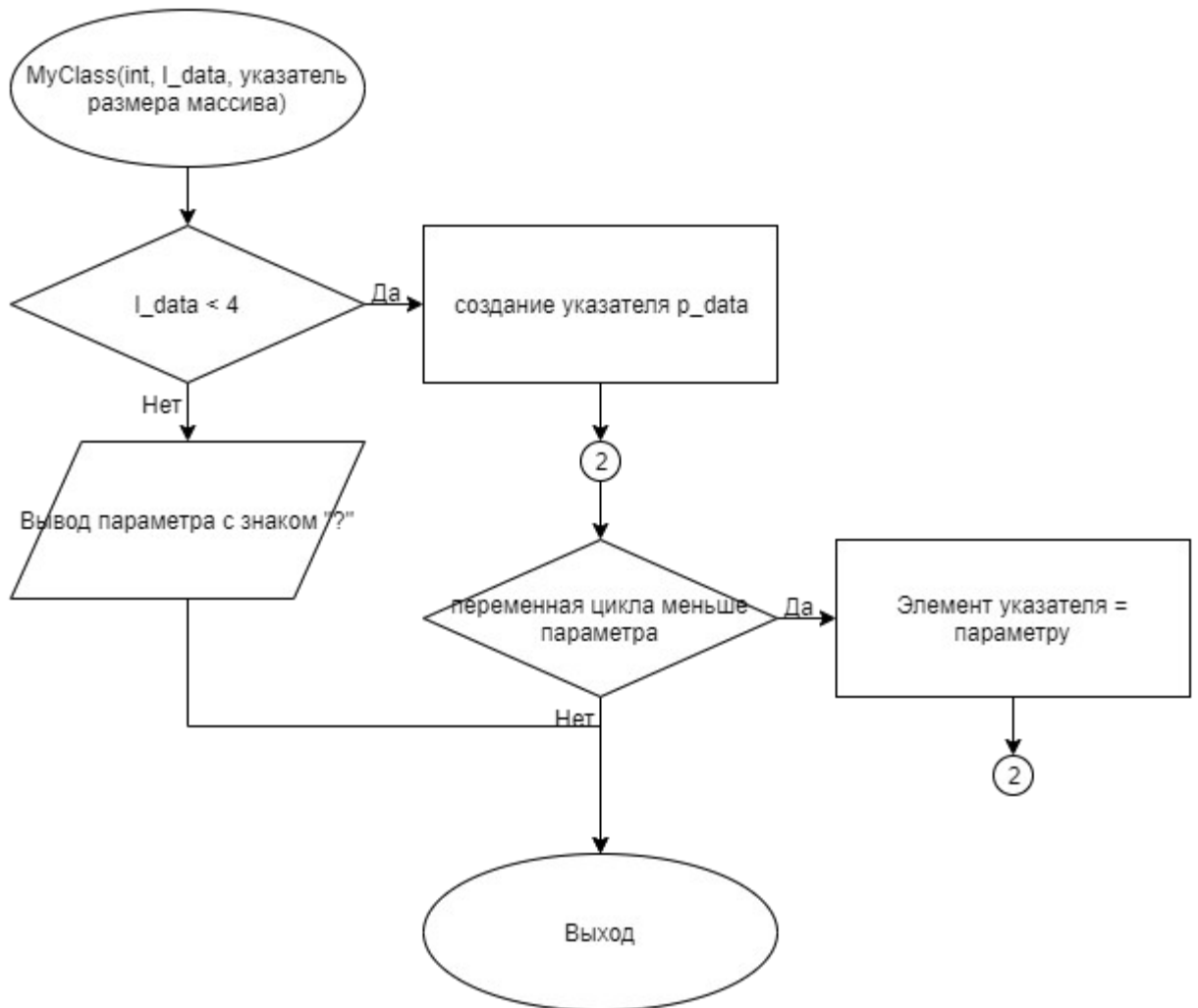


Рисунок 1 – Блок-схема алгоритма

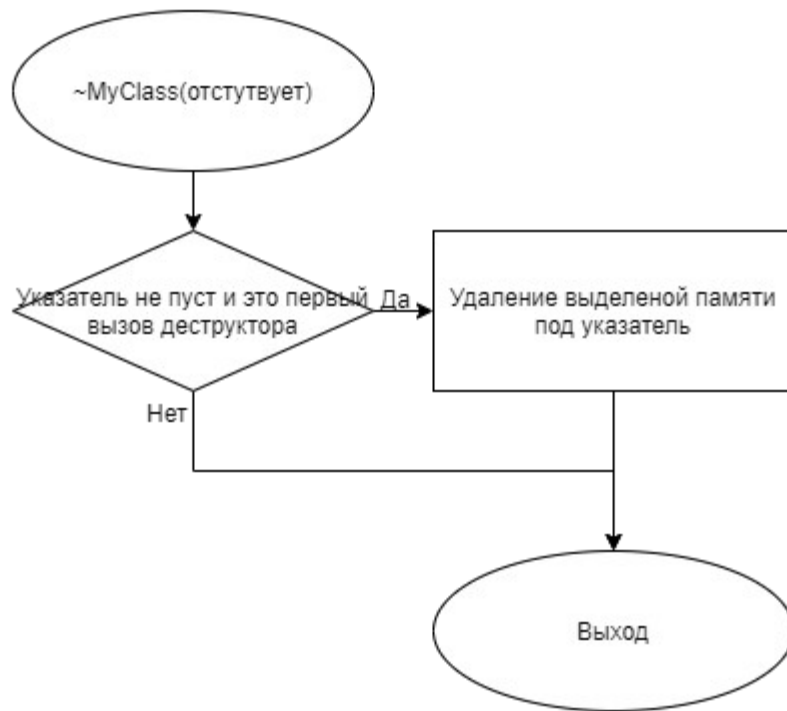


Рисунок 2 – Блок-схема алгоритма

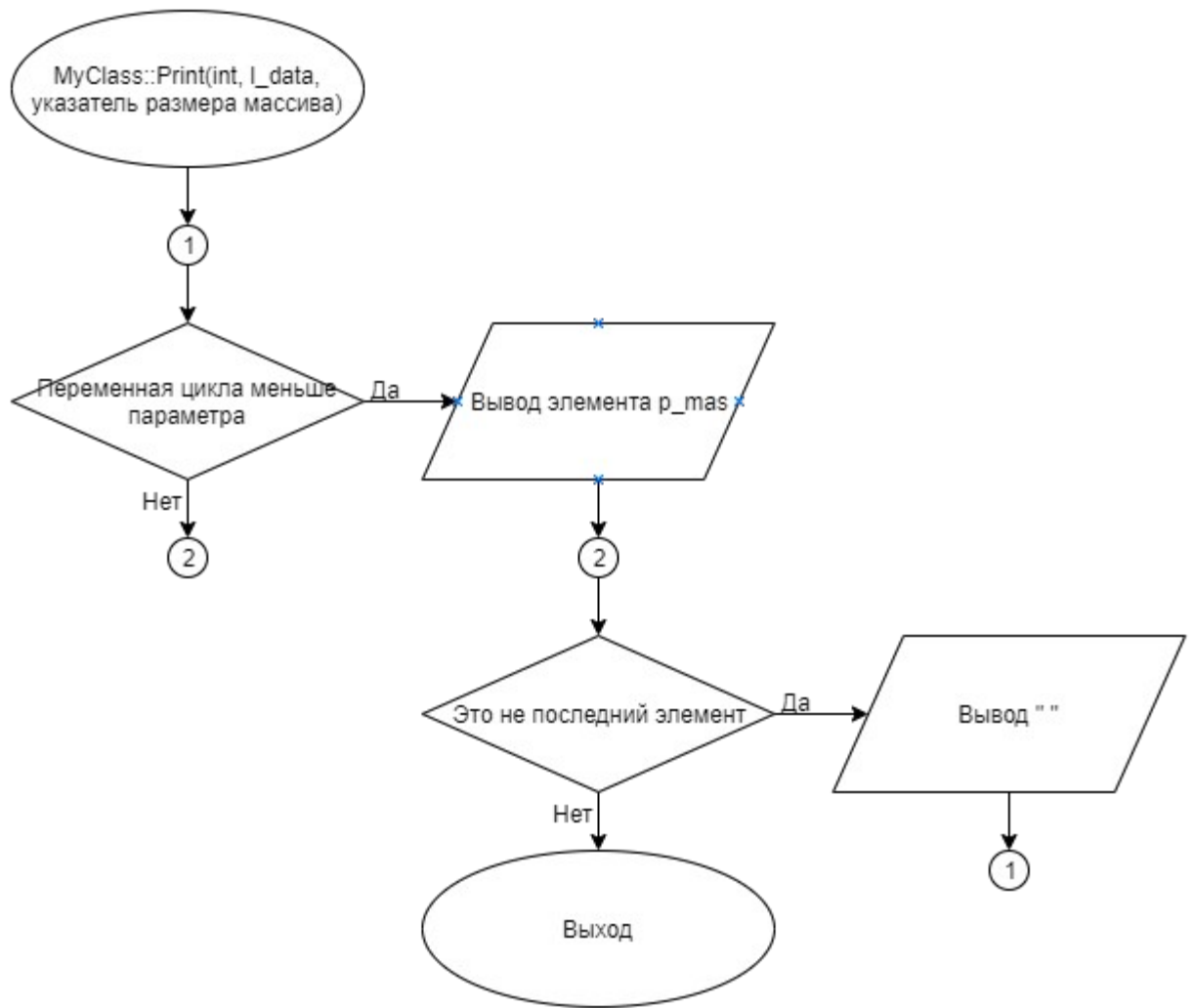


Рисунок 3 – Блок-схема алгоритма

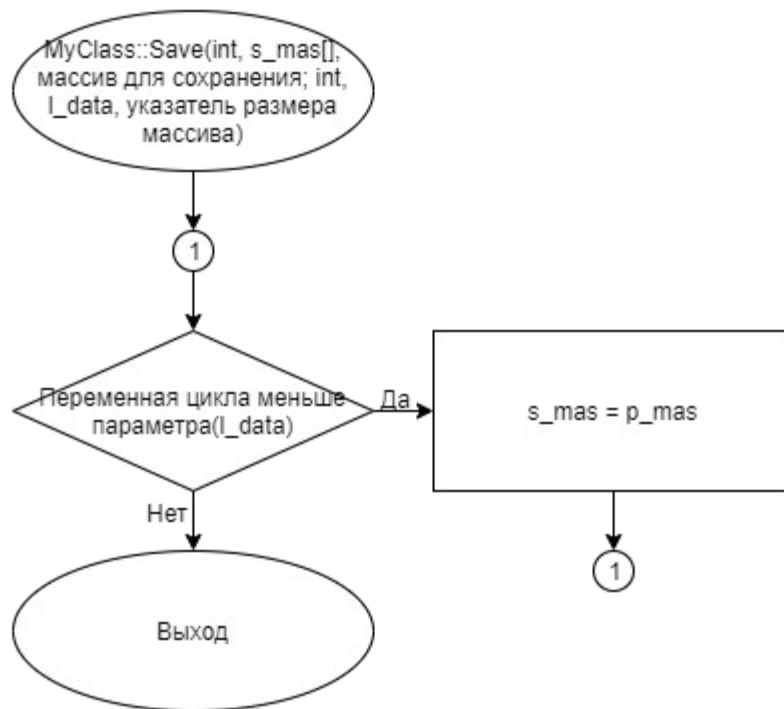


Рисунок 4 – Блок-схема алгоритма

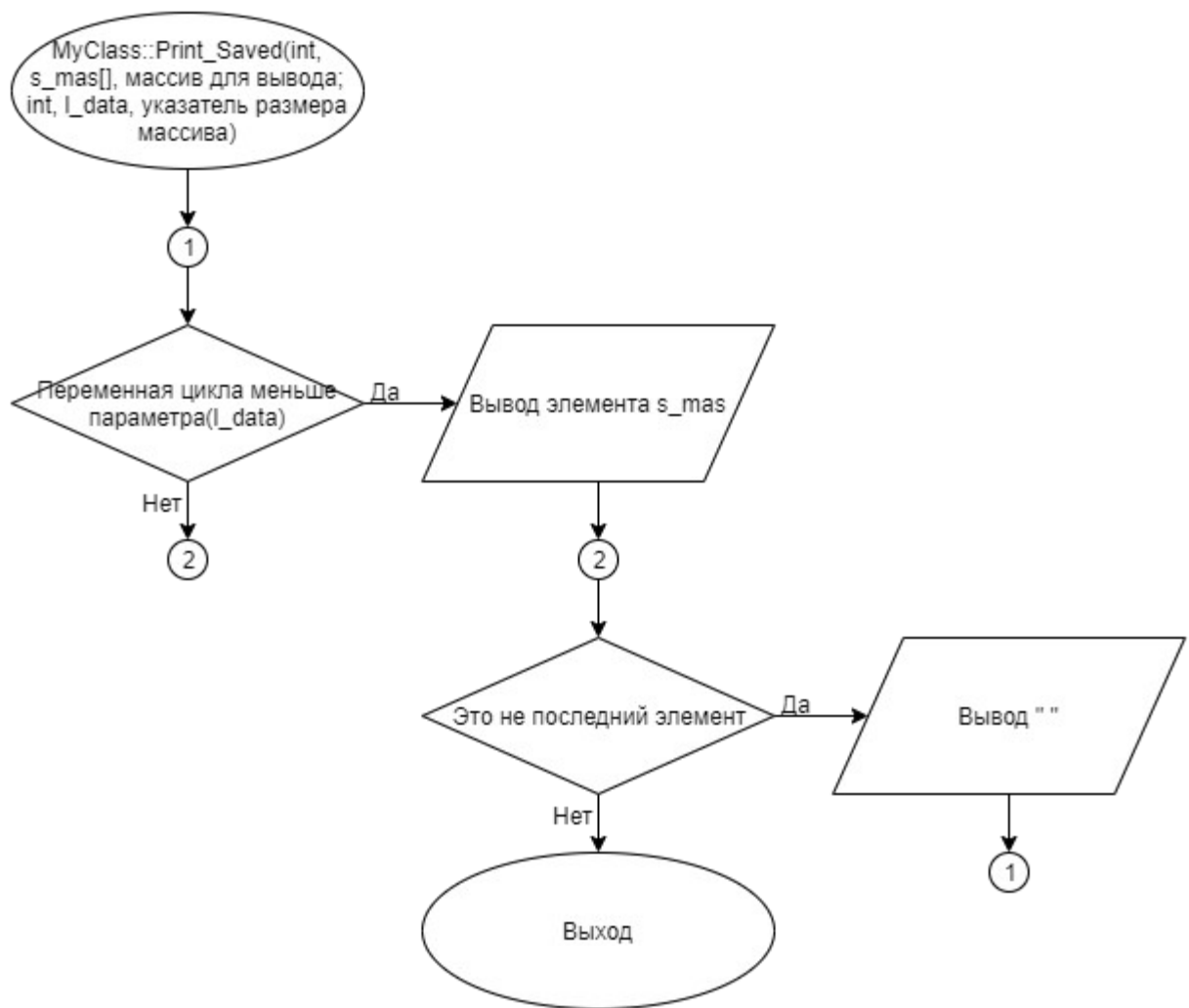


Рисунок 5 – Блок-схема алгоритма

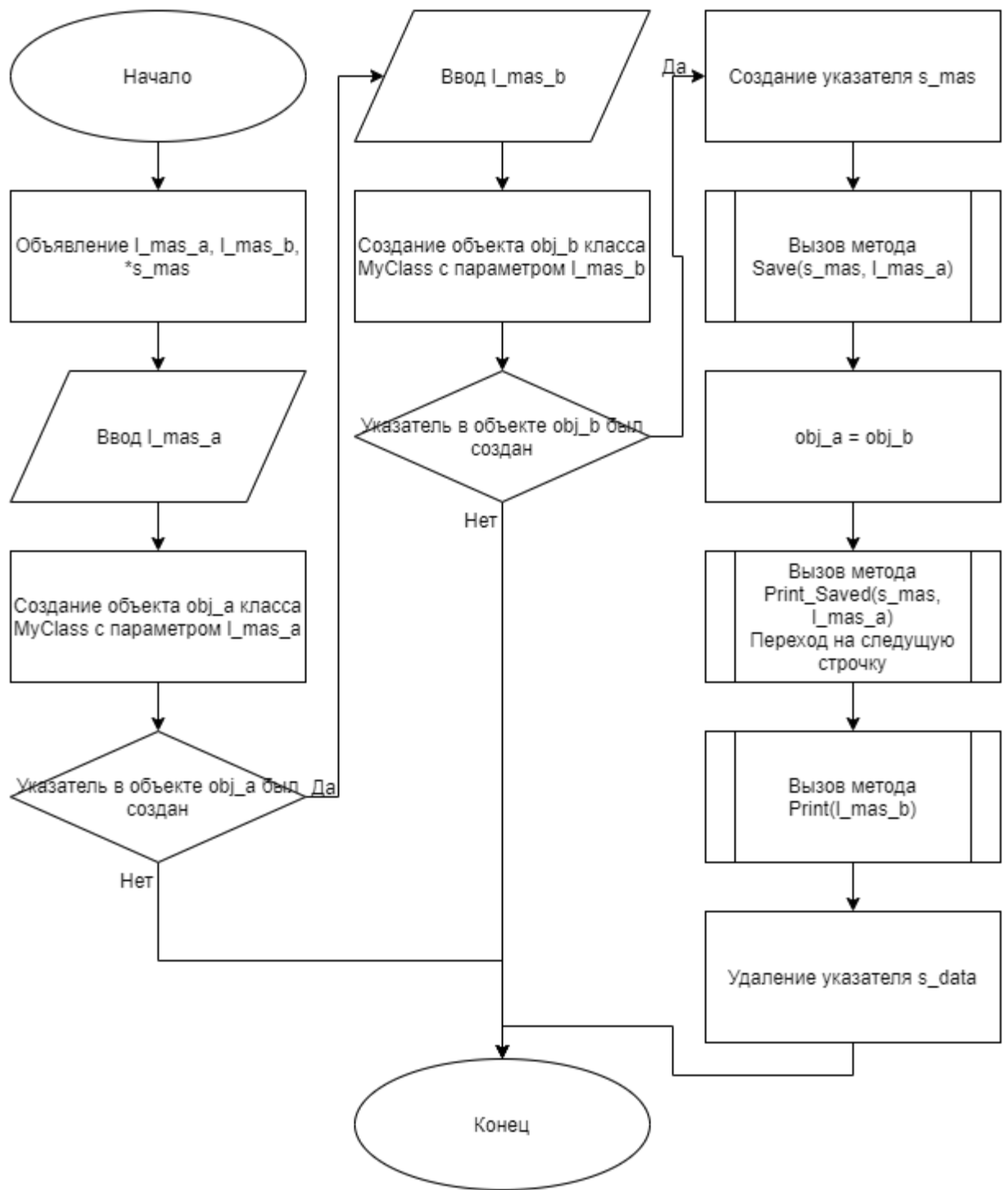


Рисунок 6 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.0 Файл main.cpp

Листинг 1 – main.cpp

```
#include "MyClass.h"

int main()
{
    int l_mas_a, l_mas_b, *s_mas;
    cin >> l_mas_a;
    MyClass obj_a(l_mas_a);
    if (obj_a.p_mas != nullptr)
    {
        cin >> l_mas_b;
        MyClass obj_b(l_mas_b);
        if (obj_b.p_mas != nullptr)
        {
            s_mas = new int[l_mas_a];
            obj_a.Save(s_mas, l_mas_a);
            obj_a = obj_b;
            obj_a.Print_Saved(s_mas, l_mas_a);
            cout << endl;
            obj_b.Print(l_mas_b);
            delete[] s_mas;
        }
    }
}
```

5.1 Файл MyClass.cpp

Листинг 2 – MyClass.cpp

```
#include "MyClass.h"

int dest = 1;

MyClass::MyClass(int l_mas)
{
    if (l_mas > 4)
    {
        p_mas = new int[l_mas];
    }
}
```



```

        for (int i = 0; i < l_mas; i++)
        {
            p_mas[i] = l_mas;
        }
    }
    else
    {
        p_mas = nullptr;
        cout << l_mas << "?";
    }
}

MyClass::~MyClass()
{
    if ((p_mas != nullptr) && (dest == 1))
    {
        delete[] p_mas;
    }
    dest++;
}

void MyClass::Print(int l_mas)
{
    for (int i = 0; i < l_mas; i++)
    {
        cout << p_mas[i];
        if (i != l_mas-1)
        {
            cout << " ";
        }
    }
}

void MyClass::Save(int s_mas[], int l_mas)
{
    for (int i = 0; i < l_mas; i++)
    {
        s_mas[i] = p_mas[i];
    }
}

void MyClass::Print_Saved(int s_mas[], int l_mas)
{
    for (int i = 0; i < l_mas; i++)
    {
        cout << s_mas[i];
        if (i != l_mas-1)
        {
            cout << " ";
        }
    }
}

```

5.2 Файл MyClass.h

Листинг 3 – MyClass.h

```
#ifndef __MYCLASS_H__
#define __MYCLASS_H__
#include <iostream>
using namespace std;
class MyClass
{
public:
    int* p_mas;
    MyClass(int l_mas);
    void Print(int l_mas);
    void Save(int s_mas[], int l_mas);
    void Print_Saved(int s_mas[], int l_mas);
    ~MyClass();
};
#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

Входные данные		Ожидаемые выходные данные								Фактические выходные данные							
5		5	5	5	5	5				5	5	5	5	5			
8		8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
4		4?								4?							
8																	
5		4?								4?							
4																	

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. C++: 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-avroora.ru/student/files/methodicheskoe_posobie_dlya_laboratornyh_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-avroora.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).