

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
3.1 Алгоритм функции main.....	8
3.2 Алгоритм метода Print класса MyClass.....	8
3.3 Алгоритм деструктора класса MyClass.....	9
3.4 Алгоритм конструктора класса MyClass.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	14
5.1 Файл main.cpp.....	14
5.2 Файл MyClass.cpp.....	14
5.3 Файл MyClass.h.....	15
6 ТЕСТИРОВАНИЕ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17

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 задачи используется:

- объект `cout` класса `iostream` предназначен для поток вывода на экран;
- объект `cin` класса `iostream` предназначен для поток ввода с клавиатуры;
- функция `main` для основной функции программы.

Класс `MyClass`:

- свойства/поля:
 - поле объявления динамического массива:
 - наименование — `array`;
 - тип — `int`;
 - модификатор доступа — `public`;
- функционал:
 - метод `MyClass` — параметризованный конструктор для создания объекта класса `MyClass`;
 - метод `Print` — вывод всех элементов массива;
 - метод `~MyClass` — 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 функции main

Функционал: основной алгоритм программы.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление int переменной temp	2
2		ввод значения переменной temp	3
3		Создание объекта Obj1 класса MyClass с параметром temp	4
4		ввод значения переменной temp	5
5		Создание объекта Obj2 класса MyClass с параметром temp	6
6		вызов метода Print объекта Obj1	7
7		вызов метода Print объекта Obj2	Ø

3.2 Алгоритм метода Print класса MyClass

Функционал: вывод всех элементов массива.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода Print класса MyClass

№	Предикат	Действия	№ перехода
1		вывод первого значения массива array	2
2	$i < \text{первого элемента массива}$	вывод двух пробелов и i -ого элемента массива array	3
			Ø
3		инкремент значения i	2

3.3 Алгоритм деструктора класса MyClass

Функционал: деструктор для удаления объекта класса.

Параметры: нет.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса MyClass

№	Предикат	Действия	№ перехода
1	array не равен значению нулевого указателя	удаление динамического массива array и освобождение выделенной под него памяти через оператор delete	Ø
			Ø

3.4 Алгоритм конструктора класса MyClass

Функционал: Параметризованный конструктор для создания объекта класса MyClass.

Параметры: int size.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса MyClass

№	Предикат	Действия	№ перехода
1	size <= 4	присвоение полю array значение нулевого указателя	2
		выделение памяти под динамический массив array размера size	4
2		вывод значения переменной size и "?"	3
3		вызов деструктора класса MyClass	4
4		инициализация переменной i = 0	5
5	i < size	присваивание i-той ячейку массива array значение переменной size	6
			∅
6		инкремент значения i	5

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.

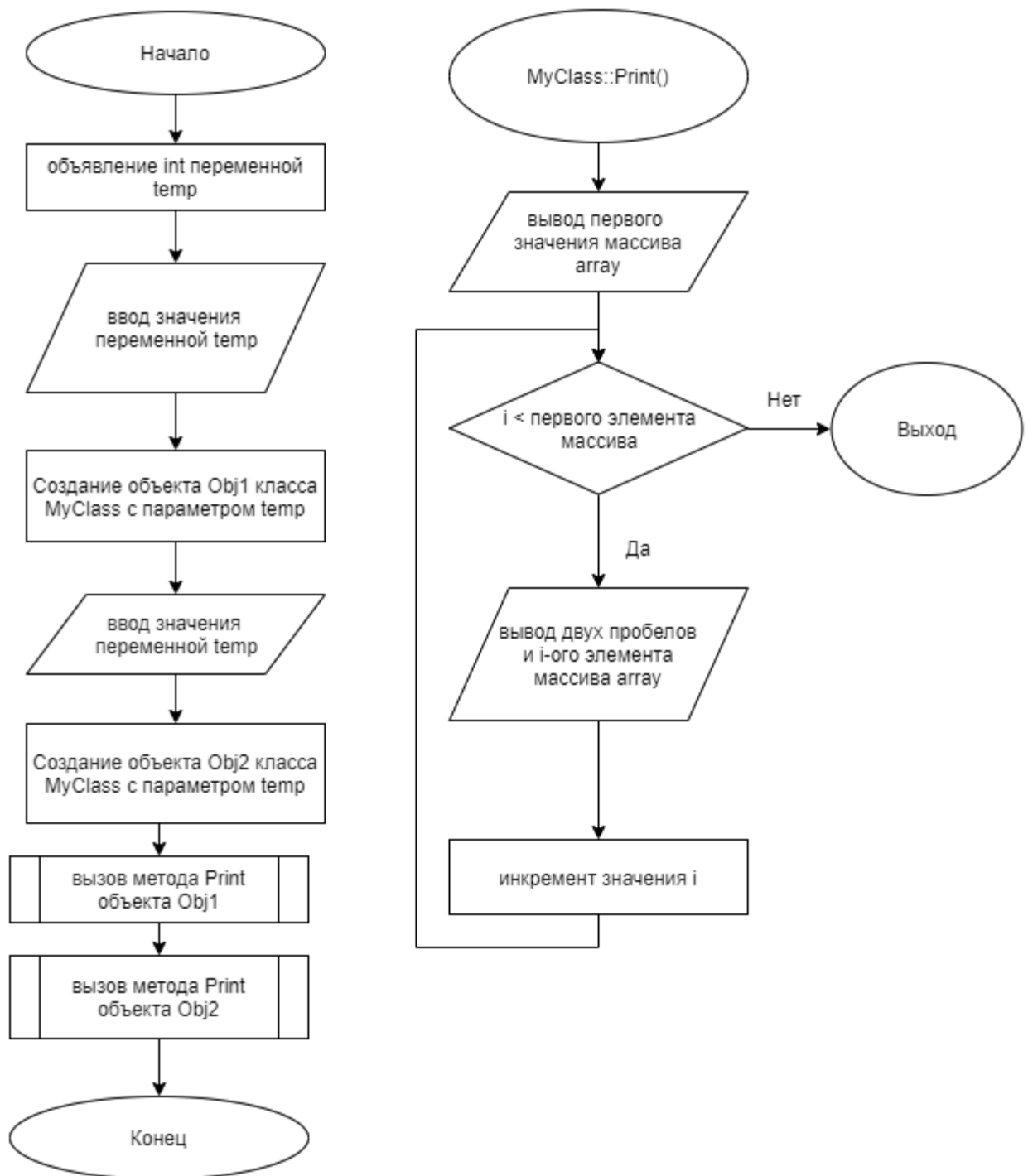


Рисунок 1 – Блок-схема алгоритма

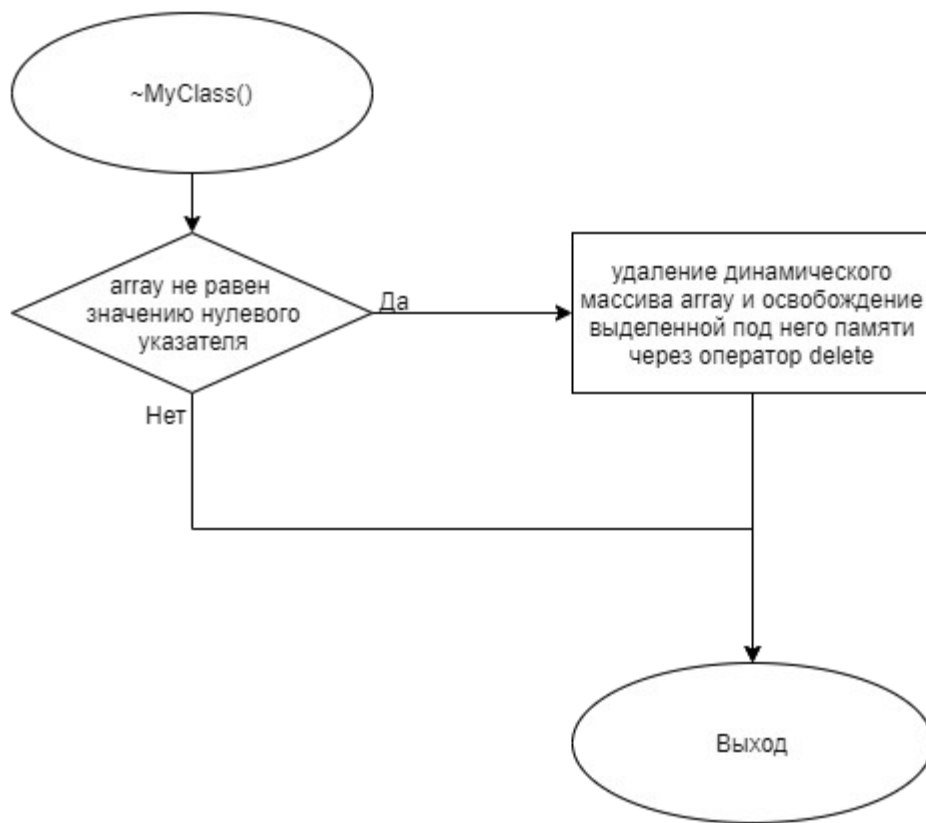


Рисунок 2 – Блок-схема алгоритма

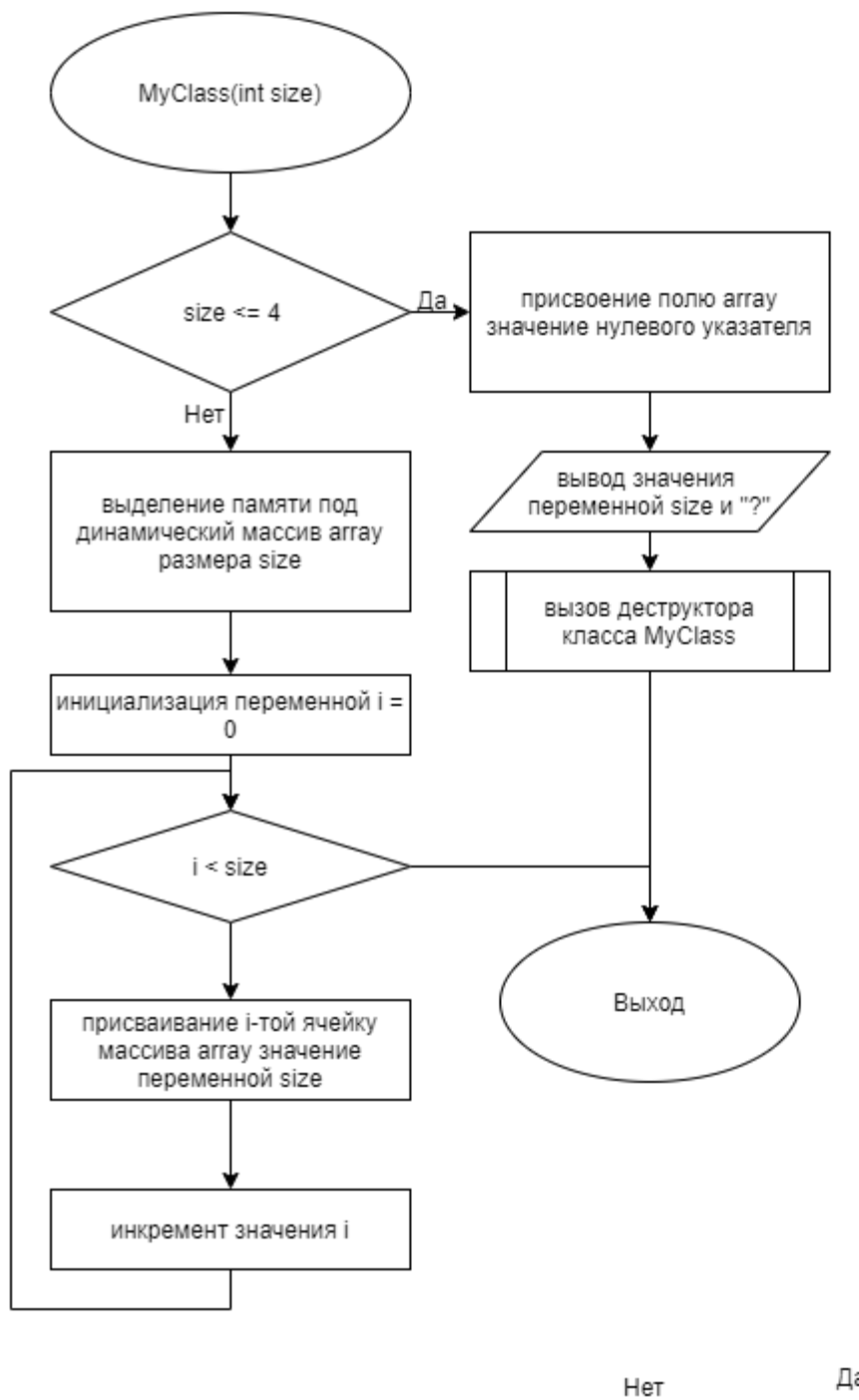


Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <iostream>
#include "MyClass.h"
using namespace std;

int main(){
    int temp;
    cin>>temp;
    MyClass Obj1(temp);

    cin>>temp;
    MyClass Obj2(temp);

    Obj1.Print();
    cout<<endl;
    Obj2.Print();

    return(0);
}
```

5.2 Файл MyClass.cpp

Листинг 2 – MyClass.cpp

```
#include <iostream>
#include "MyClass.h"
using namespace std;

MyClass::MyClass(int size){
    if (size <= 4){
        array = nullptr;
        cout <<size << "?";
        exit(0);
    }
    array = new int [size];
}
```

```

        for (int i=0;i<size;i++){
            array[i]=size;
        }
    }
    void MyClass::Print(){
        cout<< array[0];
        for (int i = 1; i<array[0]; i++)
            cout<<" "<<array[i];
    }
    MyClass::~MyClass(){
        if (array!=nullptr){
            delete[] array;
        }
    }
}

```

5.3 Файл MyClass.h

Листинг 3 – MyClass.h

```

#ifndef __MYCLASS__H
#define __MYCLASS__H

class MyClass{
public:
    int* array;
    MyClass(int size);
    void Print();
    ~MyClass();
};
#endif

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1	1?	1?
5 2	2?	2?
5 5	5 5 5 5 5 5 5 5 5 5	5 5 5 5 5 5 5 5 5 5
5 8	5 5 5 5 5 8 8 8 8 8 8 8 8	5 5 5 5 5 8 8 8 8 8 8 8 8

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).