

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 конструктора класса MyClass.....	8
3.2 Алгоритм метода get класса MyClass.....	8
3.3 Алгоритм функции main.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	16
5.1 Файл main.cpp.....	16
5.2 Файл MyClass.cpp.....	17
5.3 Файл MyClass.hpp.....	17
6 ТЕСТИРОВАНИЕ.....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19

# 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 Obj1 класса MyClass предназначен для выполнения поставленной задачи;
- объект Obj2 класса MyClass предназначен для выполнения поставленной задачи;
- Объект стандартного потока cout вывода библиотеки iostream;
- Объект стандартного потока cin ввода библиотеки iostream;
- Оператор создания динамического объекта new;
- Условный оператор if;
- Оператор цикла for.

Класс MyClass:

- свойства/поля:
  - поле Указатель на целочисленный динамический массив:
    - наименование — array;
    - тип — int \*;
    - модификатор доступа — public;
- функционал:
  - метод MyClass — Параметризованный конструктор;
  - метод get — Вывод состояния объекта.

## 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 MyClass

Функционал: Параметризированный конструктор.

Параметры: Целочисленный параметр size.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса MyClass

№	Предикат	Действия	№ перехода
1		Присвоить указателю array значение адреса целочисленного массива размера size созданного с помощью оператора new	2
2		Объявление целочисленной переменной i, значение которой равно нулю	3
3	Значение целочисленной переменной i меньше параметра size	Элементу i в целочисленном массиве array присвоить значение целочисленного параметра size	4
			Ø
4		Увеличить значение переменной i на 1	3

### 3.2 Алгоритм метода get класса MyClass

Функционал: Вывод состояния объекта.

Параметры: нет.

Возвращаемое значение: ничего.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода *get* класса *MyClass*

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной <i>i</i> , значение которой равно нулю	2
2	Значение целочисленной переменной <i>i</i> меньше значение первого элемента целочисленного массива <i>array</i>		3
			∅
3	Значение целочисленной переменной <i>i</i> увеличенное на 1 равно значению первого элемента целочисленного массива <i>array</i>	Вывод на экран значения элемента целочисленного массива <i>array</i> и "\n"	4
		Вывод на экран значения элемента целочисленного массива <i>array</i> и " "	4
4		Увеличить значение целочисленной переменной <i>i</i> на 1	2

### 3.3 Алгоритм функции *main*

Функционал: Главная функция.

Параметры: нет.

Возвращаемое значение: Целочисленное значение уведомления о завершении программы.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной <i>size</i>	2
2		Считать значение целочисленной переменной <i>size</i>	3
3	Значение целочисленной переменной <i>size</i> меньше или равно четырём	Вывод на экран значения <i>size</i> и знака вопроса	∅
			4
4		Объявление целочисленной переменной <i>size2</i>	5
5		Считать значение целочисленной переменной <i>size2</i>	6
6	Значение целочисленной переменной <i>size2</i> меньше или равно четырём	Вывод на экран значения <i>size2</i> и знака вопроса	∅
			7
7		Создать объект <i>Obj1</i> класса <i>MyClass</i> используя параметризованный конструктор, в который в качестве параметра передаём значение <i>size</i>	8
8		Создать объект <i>Obj2</i> класса <i>MyClass</i> используя параметризованный конструктор, в который в качестве параметра передаём значение <i>size2</i>	9
9		Объявить целочисленный указатель <i>temp_array</i>	10
10		Присвоить указателю <i>temp_array</i> адрес, на который ссылается <i>array</i> объекта <i>Obj1</i> класса <i>MyClass</i>	11
11		Присвоить объекту <i>Obj1</i> класса <i>MyClass</i> значение объекта <i>Obj2</i> класса <i>MyClass</i>	12
12		Присвоить массиву <i>array</i> объекта <i>Obj1</i> значение <i>temp_array</i> , который хранит в себе адрес на целочисленный массив	13



<b>№</b>	<b>Предикат</b>	<b>Действия</b>	<b>№ перехода</b>
13		Вызов метода get объекта Obj1, которое выводит состояние объекта	14
14		Вызов метода get объекта Obj2, которое выводит состояние объекта	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-4.

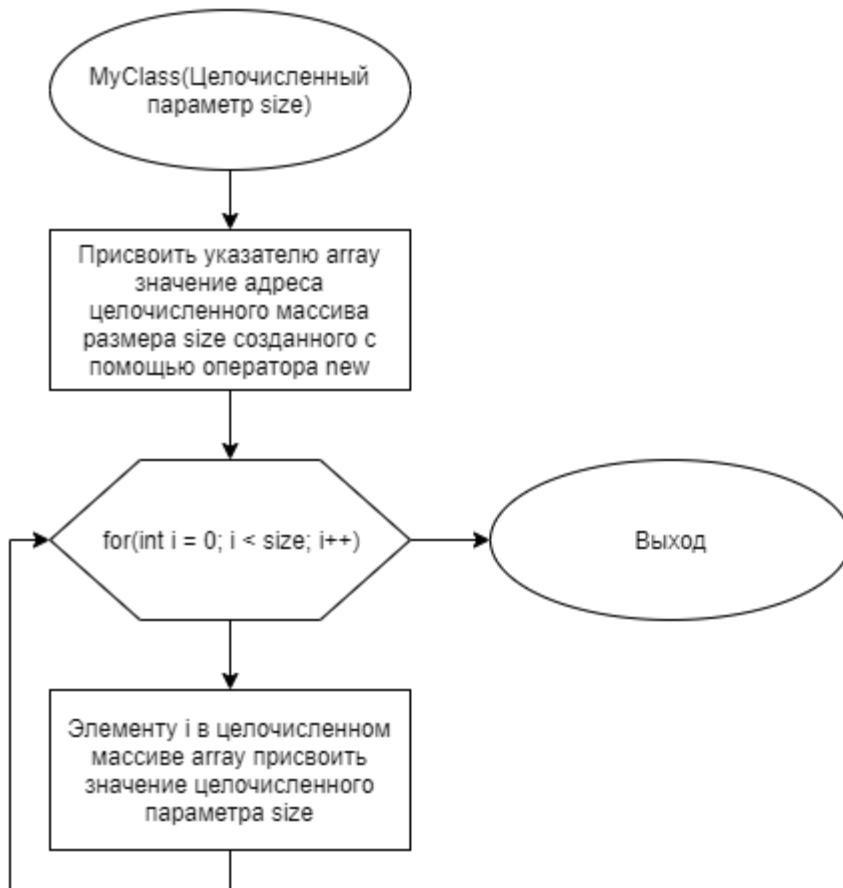
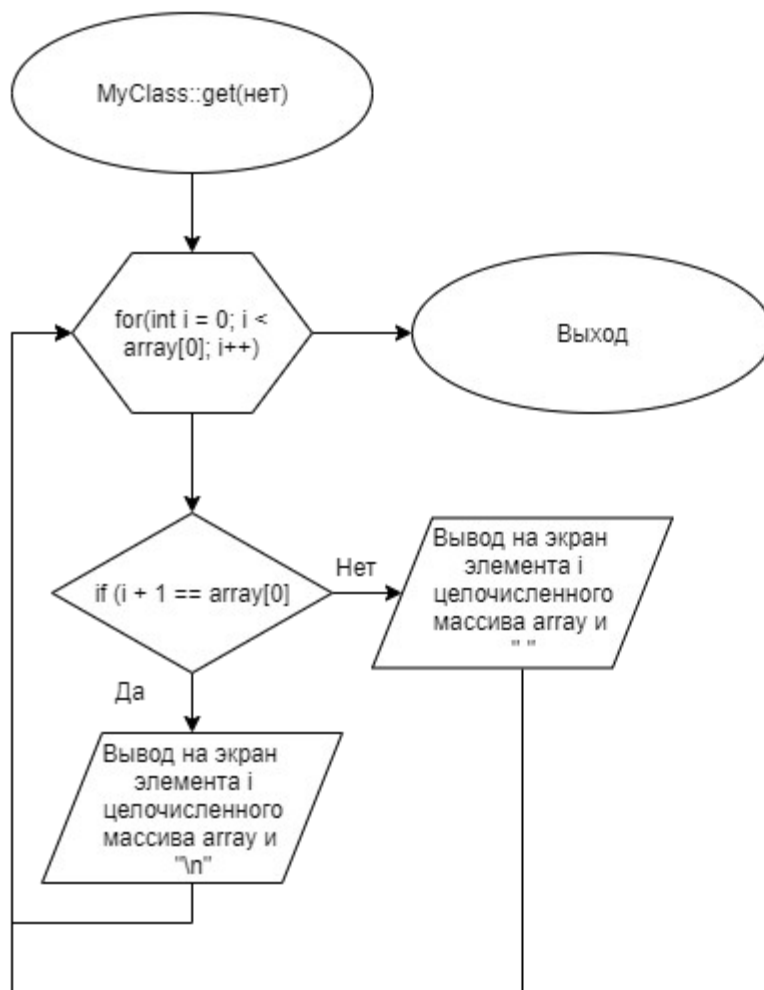


Рисунок 1 – Блок-схема алгоритма



**Рисунок 2 – Блок-схема алгоритма**

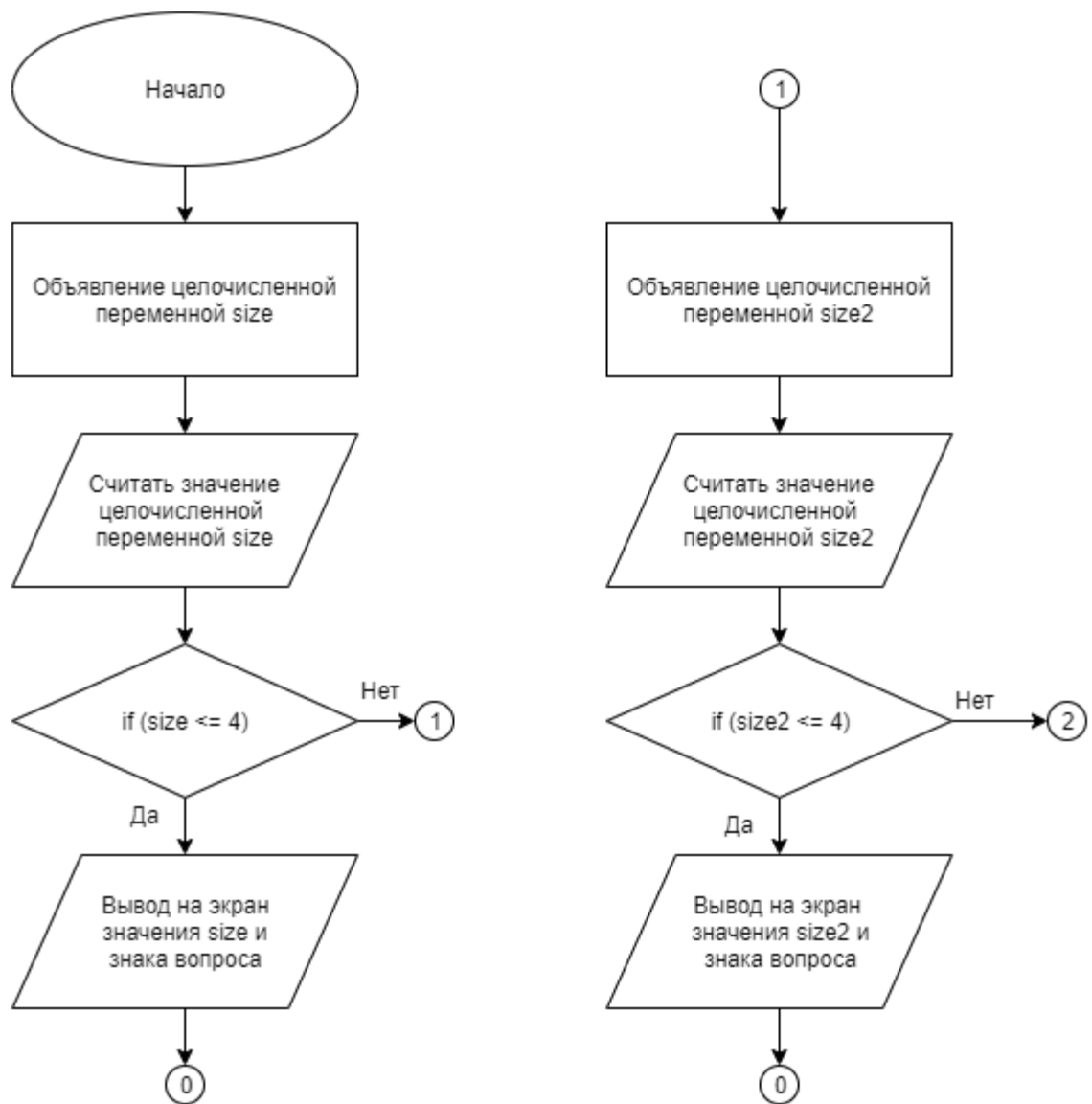


Рисунок 3 – Блок-схема алгоритма

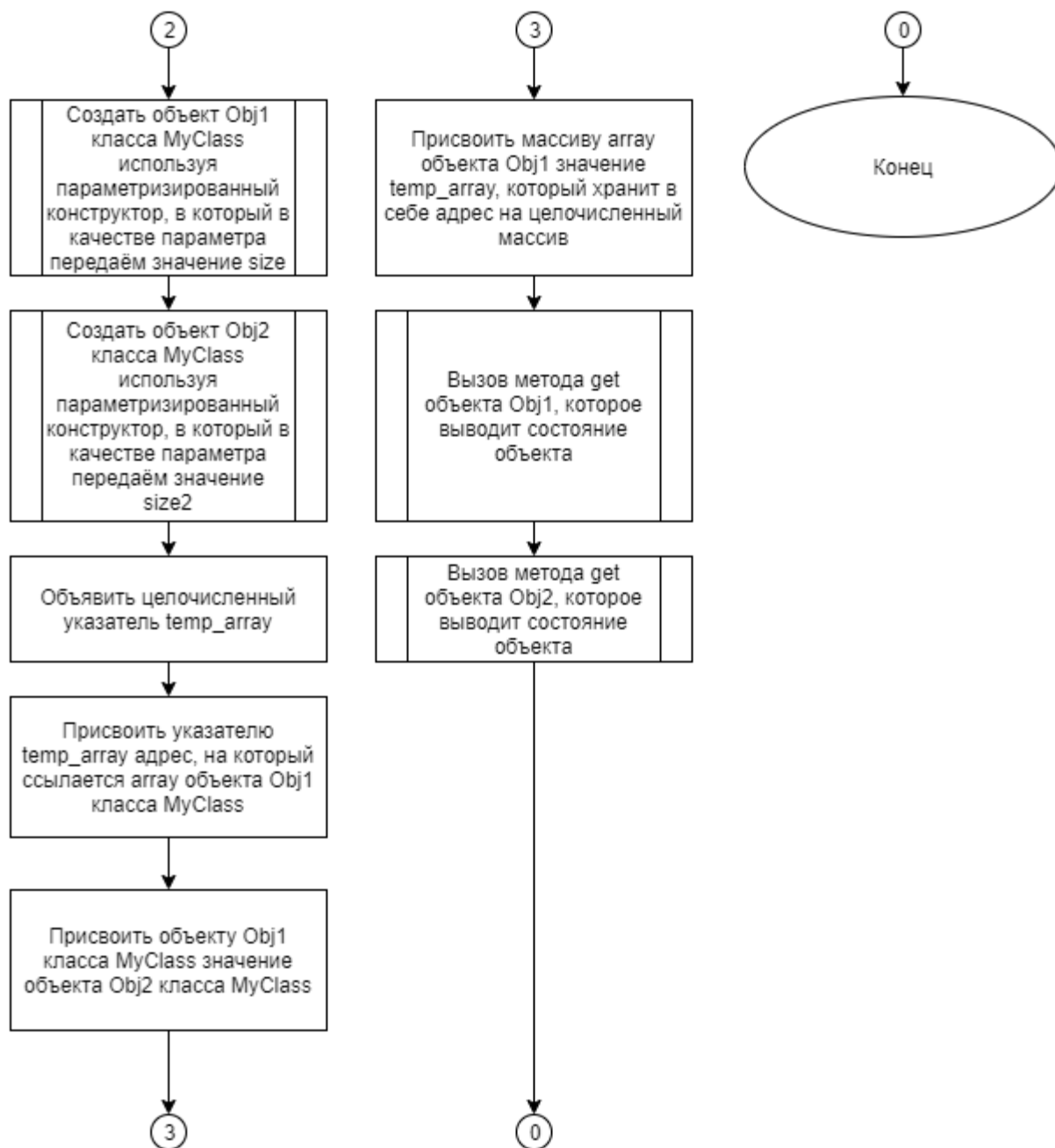


Рисунок 4 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл main.cpp

*Листинг 1 – main.cpp*

```
#include <stdlib.h>
#include <stdio.h>
#include "MyClass.hpp"

int main()
{
    int size;
    int size2;
    std::cin >> size;
    if (size <= 4)
    {
        std::cout << size << "?";
        return(0);
    }
    std::cin >> size2;
    if (size2 <= 4)
    {
        std::cout << size2 << "?";
        return(0);
    }

    MyClass Obj1(size);
    MyClass Obj2(size2);

    int * temp_array = Obj1.array;
    Obj1 = Obj2;
    Obj1.array = temp_array;

    Obj1.get();
    Obj2.get();
}
```

## 5.2 Файл MyClass.cpp

*Листинг 2 – MyClass.cpp*

```
#include "MyClass.hpp"

MyClass::MyClass(int size)
{
    array = new int[size];
    for (int i = 0; i < size; i++) array[i] = size;
}
void MyClass::get()
{
    for (int i = 0; i < array[0]; i++)
    {
        if (i + 1 == array[0])
        {
            std::cout << array[i] << "\n";
        }
        else
        {
            std::cout << array[i] << " ";
        }
    }
}
```

## 5.3 Файл MyClass.hpp

*Листинг 3 – MyClass.hpp*

```
#ifndef __MYCLASS__H
#define __MYCLASS__H
#include <iostream>

class MyClass
{
public:
    MyClass(int size);
    void get();
    int * array;
};

#endif
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 4.

Таблица 4 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1 1	1?	1?
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/methodichescoc\\_posobie\\_dlya\\_laboratornyh\\_robot\\_3.pdf](https://mirea.aco-avvora.ru/student/files/methodichescoc_posobie_dlya_laboratornyh_robot_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](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).