

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 out класса myClass.....	9
3.3 Алгоритм конструктора класса myClass.....	9
3.4 Алгоритм деструктора класса myClass.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	14
5.1 Файл main.cpp.....	14
5.2 Файл myClass.cpp.....	15
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 `obj1` класса `myClass` предназначен для демонстрирования присвоения объекта класса другому объекту того же класса;
- объект `obj2` класса `myClass` предназначен для демонстрирования присвоения объекта класса другому объекту того же класса;
- объект стандартного потока `cout`.

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: Демонстрация возможности присваивания объектов одного класса друг друга.

Параметры: нет.

Возвращаемое значение: `int` - индикатор успешности выполнения программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции `main`

№	Предикат	Действия	№ перехода
1		объявления целочисленных переменных <code>a</code> и <code>b</code>	2
2		ввод значения <code>a</code>	3
3	<code>a > 4</code>		4
		вывод "a?"	13
4		ввод значения <code>b</code>	5
5	<code>b > 4</code>	создание объекта <code>obj1</code> класса <code>myClass</code> с параметром <code>a</code>	6
		вывод "b?"	13
6		создание объекта класса <code>myClass</code> с параметром <code>b</code>	7
7		создание указателя на поле <code>Array</code> объекта <code>obj1</code>	8
8		присвоение объекту <code>obj1</code> объект <code>obj2</code>	9

№	Предикат	Действия	№ перехода
9		присвоение полю Array объекта obj1 значения по указателю	10
10		вызов метода out объекта obj1	11
11		вывод "\n"	12
12		вызов метода out объекта obj2	13
13		возврат 0	∅

3.2 Алгоритм метода out класса myClass

Функционал: Вывод элементов массива.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода out класса myClass

№	Предикат	Действия	№ перехода
1		инициализация переменной i = 0	2
2	i < Array[0] - 1	вывод элемента массива с индексом i	3
			4
3		увеличения значения i на 1	4
4		вывод первого элемента Array[]	∅

3.3 Алгоритм конструктора класса myClass

Функционал: создание указателя Array на целочисленный массив, а затем заполнение его.

Параметры: нет.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса *myClass*

№	Предикат	Действия	№ перехода
1		Создание указателя Array на целочисленный массив размера x	2
2		Инициализация целочисленной переменной i со значением 0	3
3	$i < \text{Array}[0]$	Присваиванию элементу массива $\text{Array}[i]$ значения x	4
			Ø
4		увеличения значения i на 1	3

3.4 Алгоритм деструктора класса *myClass*

Функционал: удаление массива по указателю.

Параметры: нет.

Алгоритм деструктора представлен в таблице 4.

Таблица 4 – Алгоритм деструктора класса *myClass*

№	Предикат	Действия	№ перехода
1		удаление массива по указателю Array	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

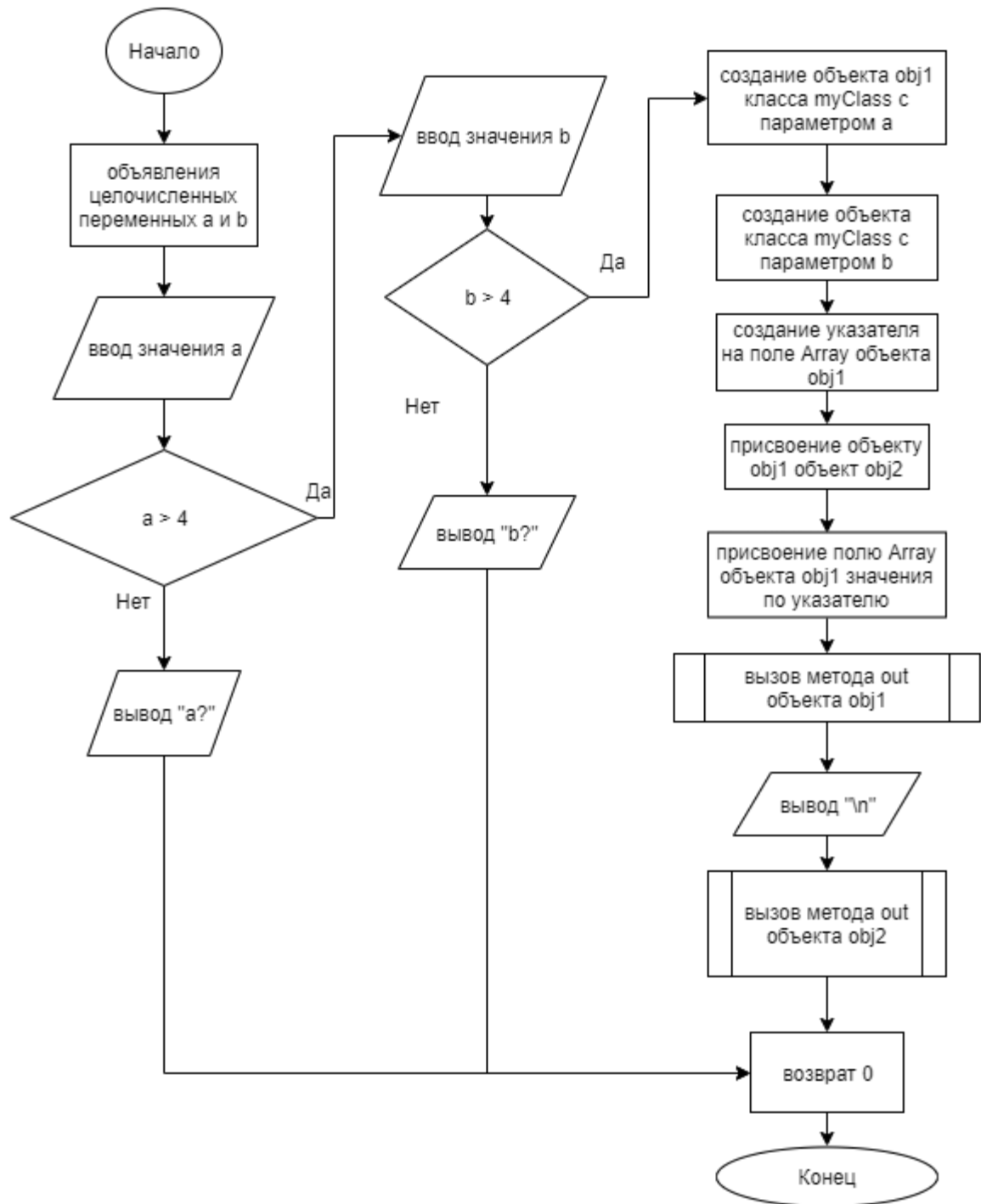


Рисунок 1 – Блок-схема алгоритма

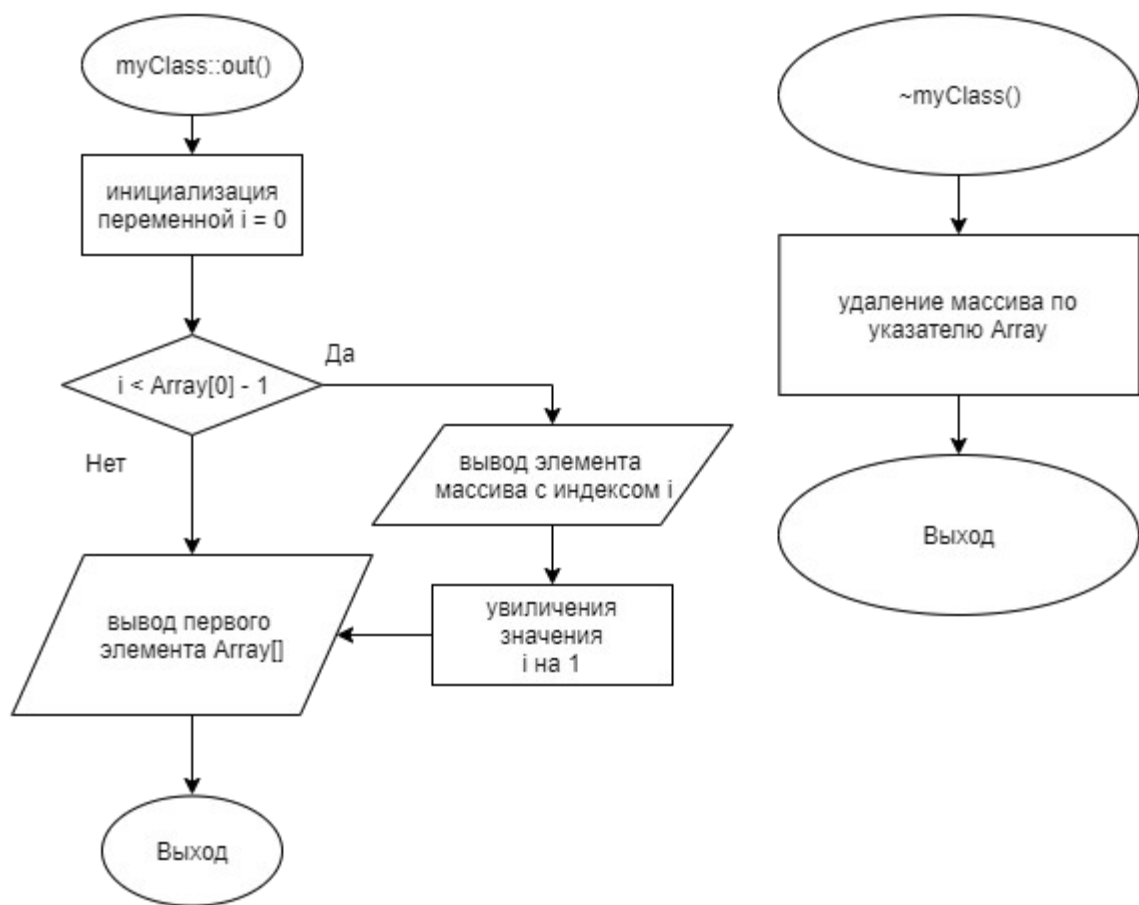


Рисунок 2 – Блок-схема алгоритма

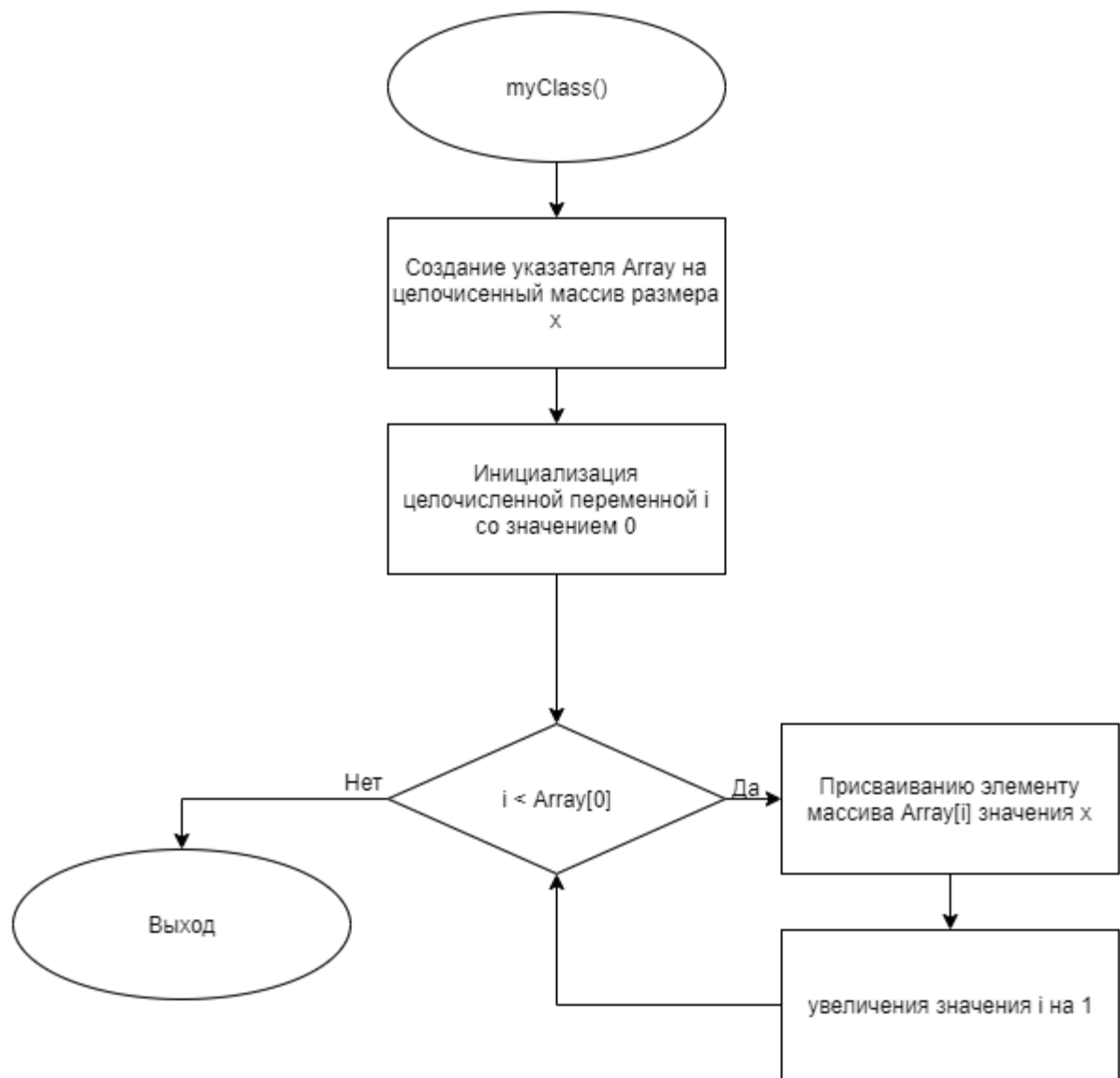


Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "myClass.h"

int main()
{
    int a, b;
    std::cin >> a;
    if (a > 4){
        std::cin >> b;
        if (b > 4){
            myClass obj1(a);
            myClass obj2(b);
            int* a = obj1.Array;
            obj1 = obj2;
            obj1.Array = a;
            obj1.out();
            std::cout << "\n";
            obj2.out();
        }
        else{
            std::cout << b << "?\n";
        }
    }
    else{
        std::cout << a << "?\n";
    }
    return 0;
}
```

5.2 Файл myClass.cpp

Листинг 2 – myClass.cpp

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

myClass::myClass(int x){
    Array = new int[x];
    for (int i = 0; i < x; i++){
        Array[i] = x;
    }
};

void myClass::out(){
    for (int i = 0; i < Array[0]-1; i++){
        std::cout << Array[i] << " ";
    }
    std::cout << Array[0];
}

myClass::~myClass(){
    delete[] Array;
}
```

5.3 Файл myClass.h

Листинг 3 – myClass.h

```
#ifndef __MYCLASS__H
#define __MYCLASS__H

class myClass{
public:
    myClass(int x);
    int* Array = nullptr;
    void out();
    ~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 1	1?	1?
5 6	5 5 5 5 5 6 6 6 6 6 6	5 5 5 5 5 6 6 6 6 6 6

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).