

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 метода getPointer класса Cls.....	8
3.2 Алгоритм метода setPointer класса Cls.....	8
3.3 Алгоритм функции main.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	10
5 КОД ПРОГРАММЫ.....	12
5.1 Файл Cls.cpp.....	12
5.2 Файл Cls.h.....	12
5.3 Файл main.cpp.....	13
6 ТЕСТИРОВАНИЕ.....	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	16

# 1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В закрытом 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 из закрытой области и каждому элементу присваивается это же значение.

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

Функциональность объекта можно расширить по усмотрению разработчика не более чем на два метода.

Спроектировать систему, которая содержит два объекта. Для построения системы последовательно, с новых строк вводятся целочисленные значения. Если значение меньше или равно 4, то создание системы прекращается и выводится сообщение. Если система построена, то посредством параметризованного конструктора создаются объекты.

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

Класс Cls:

- свойства/поля:
  - поле Указатель на массив:
    - наименование — `arg`;
    - тип — `int*`;
    - модификатор доступа — `private`;
- функционал:
  - метод `getPointer` — Возврат указателя `arg`;
  - метод `setPointer` — Установка указателя `arg` по параметру.

## 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 метода `getPointer` класса `Cls`

Функционал: Возврат указателя `arg`.

Параметры: нет.

Возвращаемое значение: `int *`.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода `getPointer` класса `Cls`

№	Предикат	Действия	№ перехода
1		Возврат <code>arg</code>	Ø

### 3.2 Алгоритм метода `setPointer` класса `Cls`

Функционал: Установка указателя `arg` по параметру.

Параметры: `narg`.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `setPointer` класса `Cls`

№	Предикат	Действия	№ перехода
1		<code>arg = narg</code>	Ø

### 3.3 Алгоритм функции main

Функционал: основная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление переменных a и b типа int	2
2		Ввод значений a и b	3
3	a <= 4	Вывод a?	13
			4
4	b <= 4	Вывод b?	13
			5
5		Создание объекта Obj1 класса Cls с параметром a	6
6		Создание объекта Obj1 класса Cls с параметром b	7
7		Инициализация указателя int * s равному значению возвращённого значения вызванного метода getPointer у объекта Obj1	8
8		Obj1 = Obj2	9
9		Вызов метода setPointer у метода Obj1 с параметром s	10
10		Вызов метода show у объекта Obj1	11
11		Вывод конца строки	12
12		Вызов метода show у объекта Obj2	13
13		Возврат значения 0	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

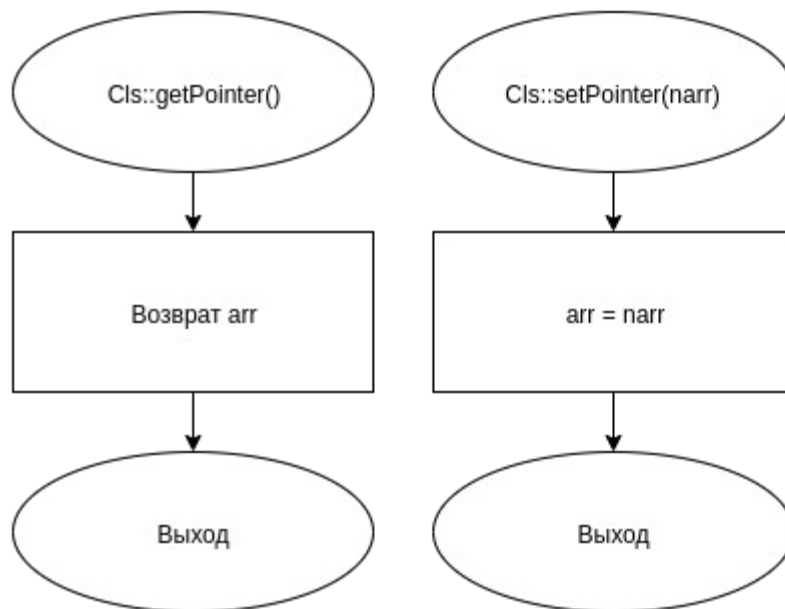


Рисунок 1 – Блок-схема алгоритма



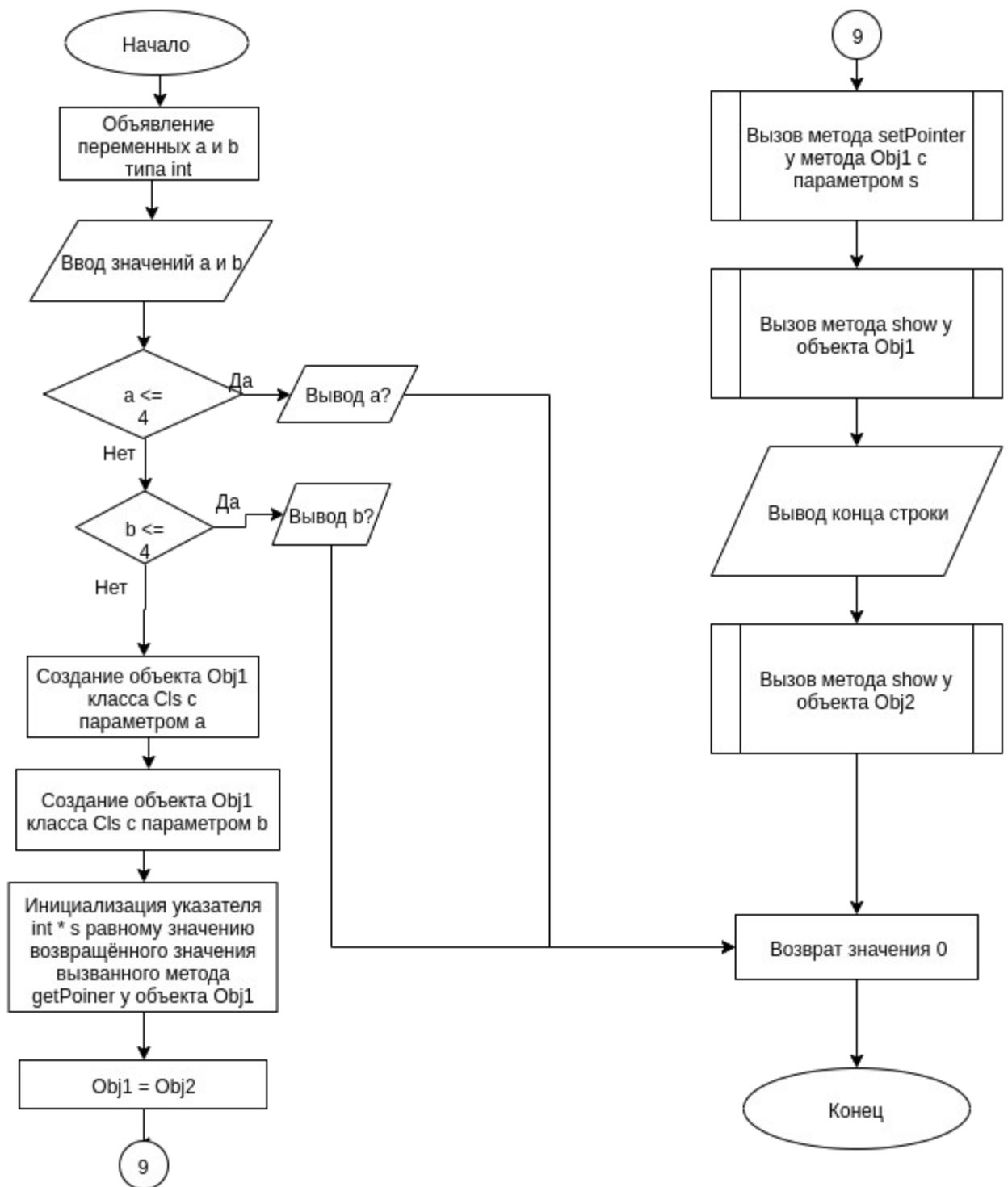


Рисунок 2 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл Cls.cpp

*Листинг 1 – Cls.cpp*

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

Cls::Cls(int a) {
    arr = new int[a];
    for(int i = 0; i < a; i++){
        arr[i] = a;
    }
}

void Cls::show() {
    for(int i = 0; i < arr[0]; i++){
        std::cout << arr[i];
        if(i != arr[0]- 1) std::cout << " ";
    }
}

int* Cls::getPointer() {
    return arr;
}

void Cls::setPointer(int* narr) {
    arr = narr;
}
```

### 5.2 Файл Cls.h

*Листинг 2 – Cls.h*

```
#ifndef __CLS__H
```

```

#define __CLS__H

class Cls{
private:
    int * arr;
public:
    int size;
    Cls(int a);
    void show();

    int* getPointer();
    void setPointer(int * narr);
};

#endif

```

### 5.3 Файл main.cpp

*Листинг 3 – main.cpp*

```

#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "Cls.h"

using namespace std;

int main()
{
    int a, b;
    cin >> a >> b;

    if(a <= 4)
        cout << a << "?";

    else {
        if(b <= 4) cout << b << "?";
        else {
            Cls Obj1(a);
            Cls Obj2(b);

            int * s = Obj1.getPointer();

            Obj1 = Obj2;

            Obj1.setPointer(s);

            Obj1.show();

```

```
        cout << endl;  
        Obj2.show();  
    }  
    }  
    return(0);  
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 4.

Таблица 4 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
3 5	3?	3?
8 2	2?	2?
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](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](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).