

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 get_Array класса object.....	8
3.2 Алгоритм метода set_Array класса object.....	8
3.3 Алгоритм функции main.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	10
5 КОД ПРОГРАММЫ.....	12
5.1 Файл main.cpp.....	12
5.2 Файл object.cpp.....	12
5.3 Файл object.h.....	13
6 ТЕСТИРОВАНИЕ.....	14
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	15

# 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 object:

- свойства/поля:
  - поле указатель на массив объекта:
    - наименование — massiv;
    - тип — int\*;
    - модификатор доступа — private;
- функционал:
  - метод get\_Array — получение массива от объекта;
  - метод set\_Array — получение массива в объект.

## 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 `get_Array` класса `object`

Функционал: получение массива от объекта.

Параметры: нет.

Возвращаемое значение: `int*` - указатель на массив объекта.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода `get_Array` класса `object`

№	Предикат	Действия	№ перехода
1		возвращение указателя на массив объекта	Ø

### 3.2 Алгоритм метода `set_Array` класса `object`

Функционал: получение массива в объект.

Параметры: `int* array`.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `set_Array` класса `object`

№	Предикат	Действия	№ перехода
1		присваивание полю <code>massiv</code> значения указателя на массив <code>array</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 переменной data_1 типа int	2
2		ввод значения data_1 с клавиатуры	3
3	data_1 > 4	объявление объекта ob_1 класса object с аргументом data_1	4
		вывод на экран data_1 '?'	12
4		объявление переменной data_2 типа int	5
5		ввод значения data_2 с клавиатуры	6
6	data_2 > 4	объявление объекта ob_2 класса object с аргументом data_2	7
		вывод на экран data_2 '?'	12
7		инициализация переменной massiv типа int* значением указателя на массив первого объекта с помощью метода get_Array() объекта ob_1	8
8		присваивание объекту ob_1 значения объекта ob_2	9
9		вызов метода set_Array() объекта ob_1 с аргументом massiv	10
10		вызов метода print() объекта ob_1	11
11		вызов метода print() объекта ob_2	12
12		возврат значения 0	Ø

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

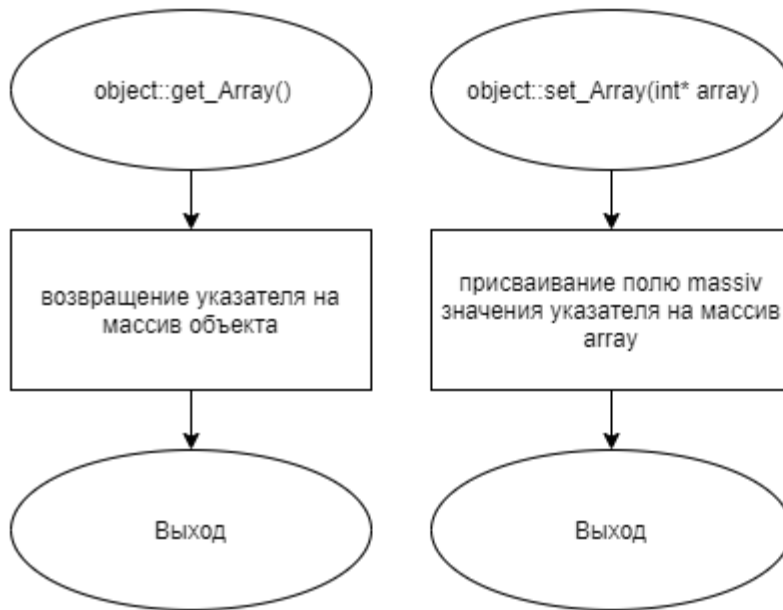


Рисунок 1 – Блок-схема алгоритма



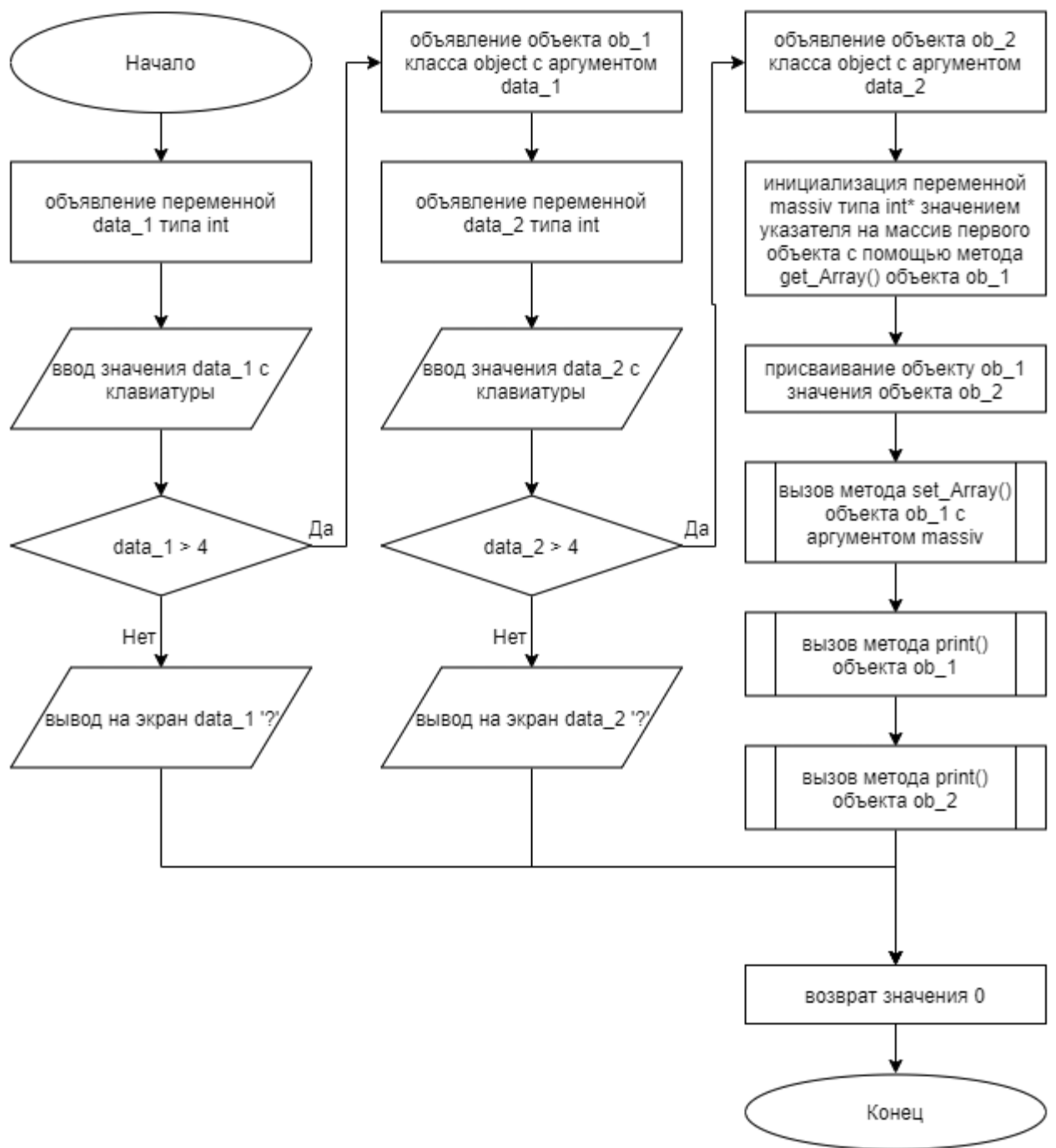


Рисунок 2 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл main.cpp

*Листинг 1 – main.cpp*

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

int main() {
    int data_1;
    std::cin >> data_1;
    if (data_1 > 4) {
        object ob_1(data_1);
        int data_2;
        std::cin >> data_2;
        if (data_2 > 4) {
            object ob_2(data_2);
            int* massiv = ob_1.get_Array();
            ob_1 = ob_2;
            ob_1.set_Array(massiv);
            ob_1.print();
            ob_2.print();
        }
        else
            std::cout << data_2 << '?';
    }
    else
        std::cout << data_1 << '?';
    return 0;
}
```

### 5.2 Файл object.cpp

*Листинг 2 – object.cpp*

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

```

object::object(int value) {
    massiv = new int[value];
    for (int i = 0; i < value; i++)
        *(massiv+ i) = value;
}
object::~~object() {
    delete[] massiv;
}
void object::print() {
    for (int i = 0; i < massiv[0]; i++) {
        std::cout << massiv[i];
        if (i != massiv[0] - 1)
            std::cout << " ";
    }
    std::cout << std::endl;
}
void object::set_Array(int* array) {
    massiv = array;
}
int* object::get_Array() {
    return massiv;
}

```

### 5.3 Файл object.h

*Листинг 3 – object.h*

```

#ifndef __OBJECT__H
#define __OBJECT__H

class object {
private:
    int* massiv;
public:
    void print();
    int* get_Array();
    void set_Array(int* array);
    object(int);
    ~object();
};

#endif

```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 4.

Таблица 4 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 2	2?	2?
1 7	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/methodichescoc\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avvora.ru/student/files/methodichescoc_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).