

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 get_array класса Object.....	10
3.3 Алгоритм метода set_array класса Object.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	17
5.1 Файл main.cpp.....	17
5.2 Файл Object.cpp.....	18
5.3 Файл Object.h.....	20
6 ТЕСТИРОВАНИЕ.....	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	22

# 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:

- ...
- Первому объекту присвоить второй объект.
- ...
- С первой строки вывести содержимое массива первого объекта.
- ...
- 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):
  - Целочисленный массив:
    - Наименование - array;
    - Тип - динамический массив целых чисел;
    - Модификатор доступа - закрытый;
- Методы:
  - Метод get\_array:
    - Функционал - возвращение указателя на скрытое поле array;
    - Возвращаемое значение - int\* (указатель на целочисленный массив);
    - Модификатор доступа - открытый;
    - Параметры - отсутствуют;
  - Метод set\_array:
    - Функционал - установка значения скрытого поля array;
    - Возвращаемое значение - void (отсутствует);
    - Модификатор доступа - открытый;
    - Параметры - указатель на целочисленный массив new\_array.

## 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: .

Возвращаемое значение: целочисленный код завершения работы программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной num1 значением 0	2
2		Инициализация целочисленной переменной num2 значением 0	3
3		Ввод num1	4
4		Ввод num2	5
5	Значение переменной num1 меньше или равно 4	Вывод num1 и "?"	∅
			6
6	Значение переменной num2 меньше или равно 4	Вывод num2 и "?"	∅
			7
7		Объявление указателя на целочисленный массив array	8

№	Предикат	Действия	№ перехода
8		Инициализация целочисленной переменной size значением 0	9
9		Создание объекта типа Object obj1 с использованием параметризованного конструктора и переменной num1 в качестве параметра	10
10		Создание объекта типа Object obj2 с использованием параметризованного конструктора и переменной num2 в качестве параметра	11
11		Присваивание указателю array поля array объекта obj1, полученного через вызов метода get_array, объекта obj1	12
12		Присваивание переменной size поля size объекта obj1	13
13		Присваивание объекта obj2 объекту obj1	14
14		Присваивание полю array объекта obj1 указателя array с помощью вызова метода set_array объекта obj1	15
15		Присваивание полю size объекта obj2 переменной size	16
16		Вызов метода print объекта obj1	17
17		Вывод "\n"	18
18		Вызов метода print объекта obj2	∅

### 3.2 Алгоритм метода `get_array` класса `Object`

Функционал: возвращение указателя на скрытое поле `array`.

Параметры: .

Возвращаемое значение: `int*` - указатель на целочисленный массив `array`.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `get_array` класса `Object`

№	Предикат	Действия	№ перехода
1		Вернуть поле (указатель) <code>array</code> объекта	Ø

### 3.3 Алгоритм метода `set_array` класса `Object`

Функционал: установка значения скрытого поля `array`.

Параметры: указатель на целочисленный массив `new_array`.

Возвращаемое значение: `void` - отсутствует.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода `set_array` класса `Object`

№	Предикат	Действия	№ перехода
1		Присвоение полю <code>array</code> объекта указателя <code>new_array</code>	Ø



## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-6.

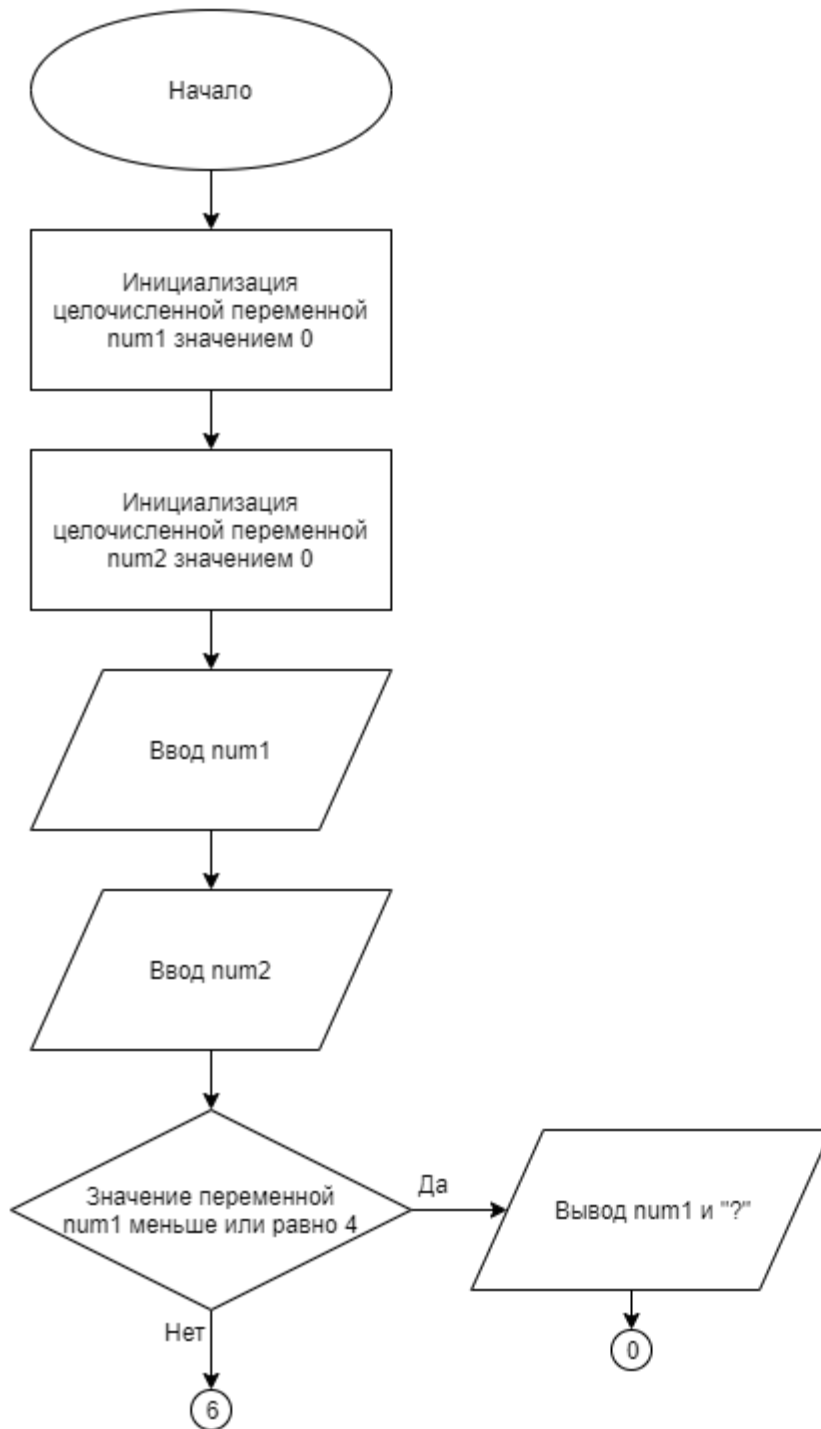


Рисунок 1 – Блок-схема алгоритма

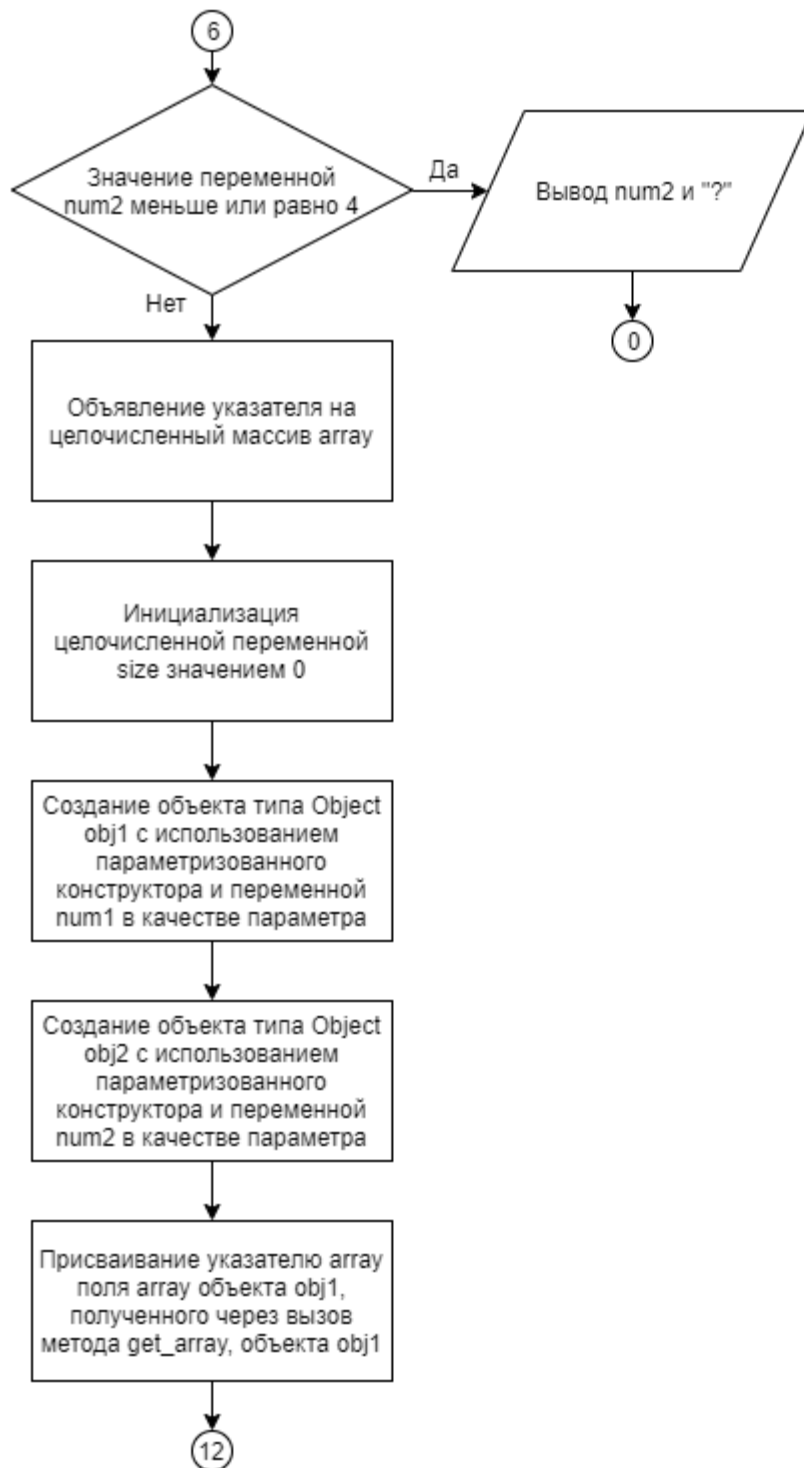
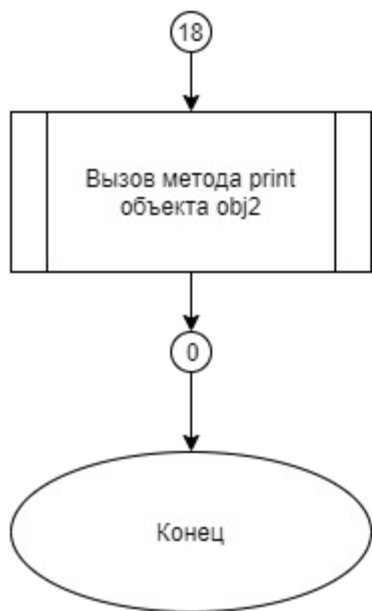


Рисунок 2 – Блок-схема алгоритма



**Рисунок 3 – Блок-схема алгоритма**



**Рисунок 4 – Блок-схема алгоритма**



**Рисунок 5 – Блок-схема алгоритма**



**Рисунок 6 – 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 "Object.h"

int main()
{
    // Инициализация целочисленной переменной
    // num1 значением 0
    int num1 = 0;
    // Инициализация целочисленной переменной
    // num2 значением 0
    int num2 = 0;
    // Ввод num1
    std::cin >> num1;
    // Ввод num2
    std::cin >> num2;

    // Если num1 не больше 4
    if (num1 <= 4)
    {
        // Вывод num1 и '?'
        std::cout << num1 << '?';
    }
    // Если num2 не больше 4
    else if (num2 <= 4)
    {
        // Вывод num2 и '?'
        std::cout << num2 << '?';
    }
    else
    {
        // Объявление указателя
        // на целочисленный массив
        int* array;
        // Инициализация целочисленной переменной
        // size значением 0
        int size = 0;

        // Создание объекта типа Object
        // obj1 с использованием
        // параметризованного конструктора
        Object obj1(num1);
    }
}
```

```

        // Создание объекта типа Object
        // obj2 с использованием
        // параметризованного конструктора
        Object obj2(num2);

        // Присваивание указателю
        // array поля array, полученного
        // через вызов метода
        // get_array, объекта obj1
        array = obj1.get_array();
        // Присваивание переменной
        // size поля size
        // объекта obj1
        size = obj1.size;

        // Присваивание объекта
        // obj2 объекту obj1
        obj1 = obj2;

        // Присваивание полю
        // array объекта obj1
        // указателя array
        // с помощью вызова метода
        // set_array указателя obj1
        obj1.set_array(array);
        // Присваивание полю
        // size объекта obj2
        // переменной size
        obj1.size = size;

        // Вызов метода print
        // объекта obj1
        obj1.print();
        // Переход на новую строку
        std::cout << '\n';
        // Вызов метода print
        // объекта obj2
        obj2.print();
    }

    return 0;
}

```

## 5.2 Файл Object.cpp

*Листинг 2 – Object.cpp*

```

#include "Object.h"

Object::Object(int size)
{
    // Присваивание полю
    // size объекта параметра size

```



```

        this->size = size;
        // Присваивание полю array указателя
        // на новый целочисленный массив размера
        // size, созданный с помощью оператора new
        array = new int[size];

        // Цикл for с переменной счётчиком
        // i принимающей значения в диапазоне
        // [0; size)
        for (int i = 0; i < size; i++)
        {
            // Присваивание i-тому элементу
            // массива array значения поля size
            array[i] = size;
        }
    }

Object::~Object()
{
    // Освобождение памяти
    // по указателю array
    // с помощью оператора delete
    delete[] array;
}

int* Object::get_array()
{
    return array;
}

void Object::set_array(int* new_array)
{
    array = new_array;
}

void Object::print()
{
    // Цикл for с переменной счётчиком
    // i принимающей значения в диапазоне
    // [0; size)
    for (int i = 0; i < size; i++)
    {
        // Вывод элемента с индексом i массива
        std::cout << array[i];
        // Если i не равно size - 1
        if (i != size - 1)
        {
            // Вывод " " (2 пробела)
            std::cout << " ";
        }
    }
}
}

```

## 5.3 Файл Object.h

*Листинг 3 – Object.h*

```
#ifndef OBJECT_H
#define OBJECT_H

#include <iostream>

class Object
{
private:
    int* array;
public:
    int size;
    Object(int size);
    ~Object();
    int* get_array();
    void set_array(int* new_array);
    void print();
};

#endif
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 4.

Таблица 4 – Результат тестирования программы

Входные данные		Ожидаемые выходные данные								Фактические выходные данные							
5		5	5	5	5	5				5	5	5	5	5			
8		8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
4		4?								4?							
6																	
6		4?								4?							
4																	

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.
3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] — URL: [https://mirea.aco-avroora.ru/student/files/methodicheskoe\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avroora.ru/student/files/methodicheskoe_posobie_dlya_laboratornyh_rabot_3.pdf) (дата обращения 05.05.2021).
4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: [https://mirea.aco-avroora.ru/student/files/Prilozheniye\\_k\\_methodichke.pdf](https://mirea.aco-avroora.ru/student/files/Prilozheniye_k_methodichke.pdf) (дата обращения 05.05.2021).
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).