

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.....	9
3.1 Алгоритм конструктора класса cl1.....	9
3.2 Алгоритм метода PrintArray класса cl1.....	10
3.3 Алгоритм метода getter класса cl1.....	10
3.4 Алгоритм метода setter класса cl1.....	11
3.5 Алгоритм деструктора класса cl1.....	11
3.6 Алгоритм функции main.....	12
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	15
5.1 Файл cl1.cpp.....	15
5.2 Файл cl1.h.....	16
5.3 Файл main.cpp.....	16
6 ТЕСТИРОВАНИЕ.....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19

# 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

Для решения задачи используется:

- объект `Object1` класса `cl1` предназначен для Демонстрация работы операции присвоения посредством оператора присвоения;
- объект `Object2` класса `cl1` предназначен для Демонстрация работы операции присвоения посредством оператора присвоения;
- функция `main` для главная функция программы;
- объект стандартного потока ввода `cin` с клавиатуры;
- объект стандартного потока вывода `cout` на экран;
- библиотека `iostream`;
- пространство имён `std`;
- оператор уикла со счётчиком `for`;
- условная конструкция.

Класс `cl1`:

- свойства/поля:
  - поле `array`:
    - наименование — Объявление указателя на динамический массив;
    - тип — `int*`;
    - модификатор доступа — `private`;
- функционал:
  - метод `cl1` — параметризированный конструктор для создания объекта класса `cl1`;
  - метод `PrintArray` — вывод всех элементов массива на экран;
  - метод `getter` — возврат значения скрытого поля `array` класса `cl1`;
  - метод `setter` — установка нового значения скрытого поля `array`

класса `cl1`;

- о метод `~cl1` — деструктор для удаления объекта класса `cl1`.

## 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 cl1

Функционал: параметризованный конструктор для создания объекта класса cl1.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса cl1

№	Предикат	Действия	№ перехода
1	Значение переменной sizeofarray меньше или равно 4	Присвоение полю array значения нулевого указателя	2
	Значение переменной sizeofarray не меньше или не равно 4	Выделение памяти под динамический массив array размером, равному значению переменной sizeofarray, посредством оператора new	4
2		Вывод: <<значение sizeofarray>> "?"	3
3		Вызов деструктора класса cl1	∅
4		Инициализация целочисленной переменной i значением 0	5
5	Значение переменной i меньше значения переменной sizeofarray	Присваивание i-ой ячейке памяти массива array значения переменной sizeofarray	6
	Значение переменной i не меньше значения		∅

№	Предикат	Действия	№ перехода
	переменной sizeofarray		
6		Инкремент перменной i	5

### 3.2 Алгоритм метода PrintArray класса cl1

Функционал: вывод всех элементов массива на экран.

Параметры: нет.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода PrintArray класса cl1

№	Предикат	Действия	№ перехода
1		Вывод: <<значение первого элемента массива array>>	2
2		Инициализация целочисленной переменной i значением 1	3
3	Значение переменной i меньше значения первого элемента массива array	Вывод: " <<значение i-го элемента массива array>>	4
	Значение переменной i не меньше значения первого элемента массива array		Ø
4		Инкремент перменной i	3

### 3.3 Алгоритм метода getter класса cl1

Функционал: возврат значения скрытого поля array класса cl1.

Параметры: нет.

Возвращаемое значение: int\*, указатель на первый элемент массива array.



Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода *getter* класса *cl1*

№	Предикат	Действия	№ перехода
1		Возврат значения скрытого поля <i>array</i>	Ø

### 3.4 Алгоритм метода *setter* класса *cl1*

Функционал: установка нового значения скрытого поля *array* класса *cl1*.

Параметры: *int\** *temp*, динамический массив для изменения значения поля *array* класса *cl1*.

Возвращаемое значение: *int\**, указатель на первый элемент массива *array*.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *setter* класса *cl1*

№	Предикат	Действия	№ перехода
1		Возвращение значения скрытого поля <i>array</i>	Ø

### 3.5 Алгоритм деструктора класса *cl1*

Функционал: деструктор для удаления объекта класса *cl1*.

Параметры: нет.

Алгоритм деструктора представлен в таблице 5.

Таблица 5 – Алгоритм деструктора класса *cl1*

№	Предикат	Действия	№ перехода
1	Значение поля <i>array</i> класса <i>cl1</i> не равно значению нулевого указателя	Удаление динамического массива <i>array</i> и освобождение выделенной под него памяти посредством оператора <i>delete</i>	Ø
	Значение поля <i>array</i> класса <i>cl1</i> равно значению нулевого		Ø

№	Предикат	Действия	№ перехода
	указателя		

### 3.6 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое, индикация корректности работы программы.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной data	2
2		Ввод значения перменной data	3
3		Создание объекта Object1 класса cl1 путём вызова конструктора с параметром data	4
4		Ввод значения переменной data	5
5		Создание объекта Object2 класса cl1 путём вызова конструктора с параметром data	6
6		Инициализация динамического массива temp значением поля array путём вызова метода getter объекта Object1 класса cl1	7
7		Присваивание объекту Object1 значения объекта Object2	8
8		Вызов метода setter объекта Object1 с передачей в качестве аргумента динамического массива temp	9
9		Вызов метода PrintArray объекта Object1	10
10		Переход на новую строку	11
11		Вызов метода PrintArray объекта Object2	Ø

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

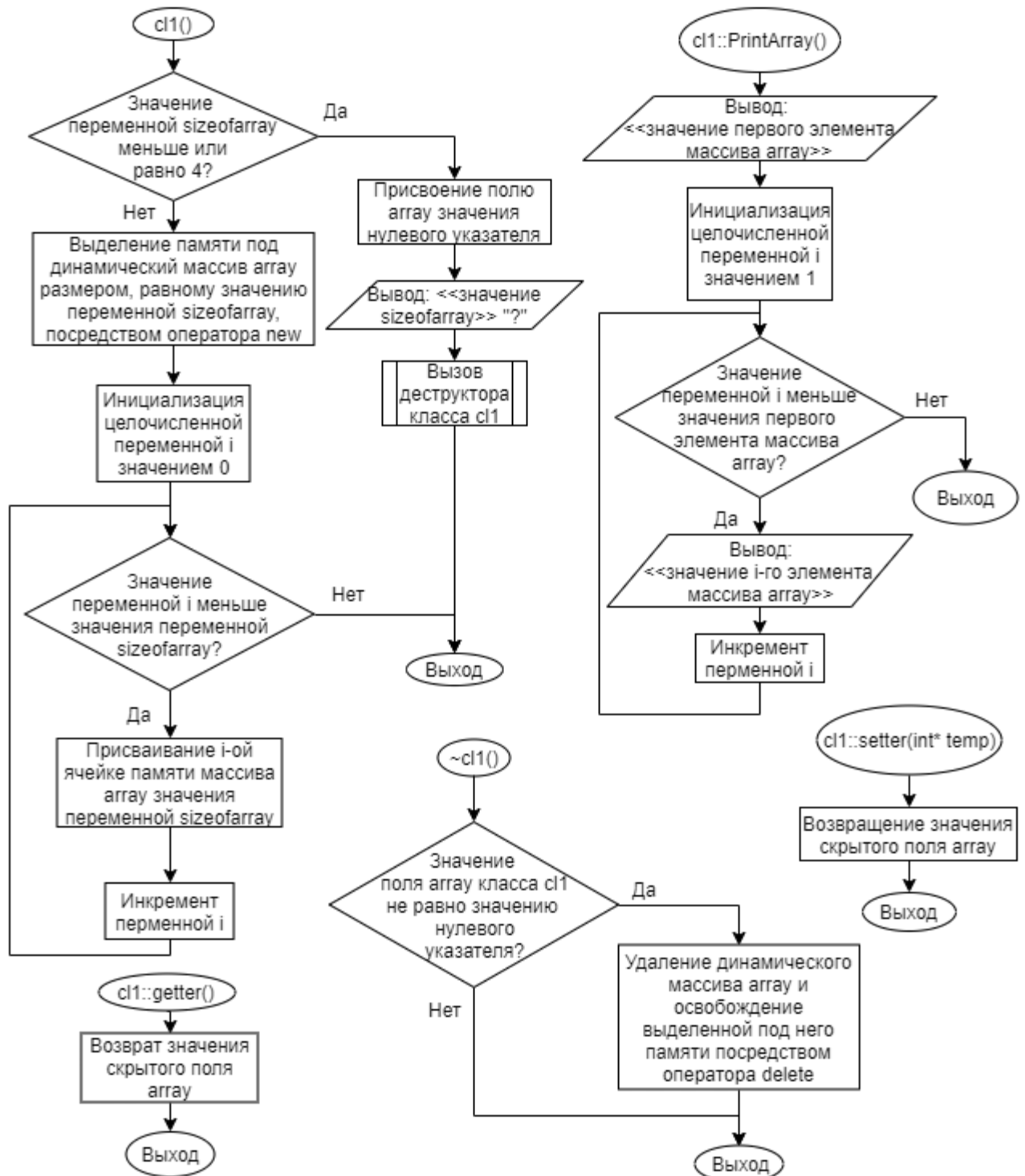


Рисунок 1 – Блок-схема алгоритма

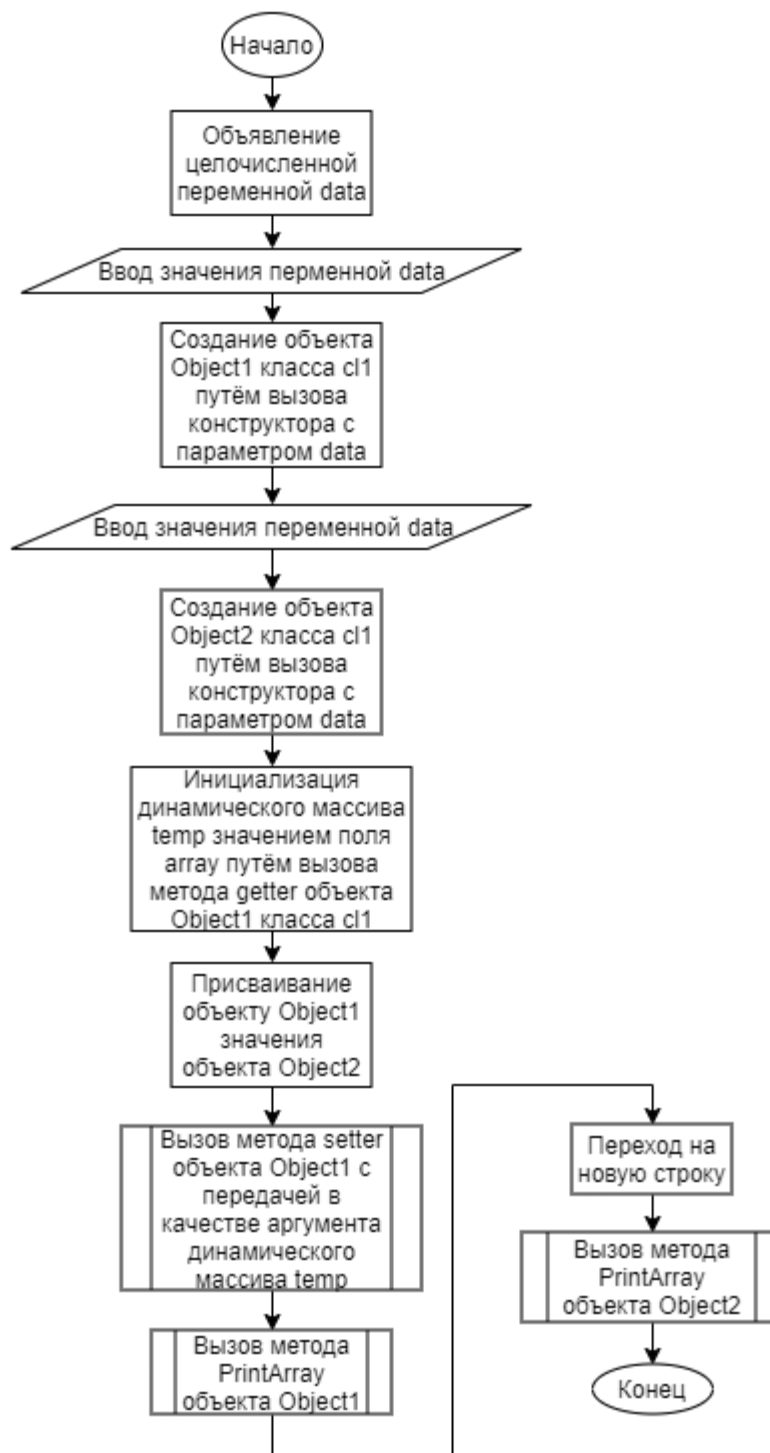


Рисунок 2 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл cl1.cpp

*Листинг 1 – cl1.cpp*

```
#include <iostream>
#include "cl1.h"
using namespace std;

cl1::cl1(int sizeofarray)
{
    if (sizeofarray <= 4)
    {
        array = nullptr;
        cout << sizeofarray << "?";
        exit(0);
    }
    array = new int[sizeofarray];
    for (int i = 0; i < sizeofarray; i++)
    {
        array[i] = sizeofarray;
    }
}
void cl1::PrintArray()
{
    cout << array[0];
    for (int i = 1; i < array[0]; i++)
    {
        cout << " " << array[i];
    }
}
int* cl1::getter()
{
    return array;
}
void cl1::setter(int* temp)
{
    array = temp;
}
cl1::~cl1()
{
    if(array != nullptr)
    {
        delete[] array;
    }
}
```

```
}  
}
```

## 5.2 Файл cl1.h

*Листинг 2 – cl1.h*

```
#ifndef CLASS1_H  
#define CLASS1_H  
class cl1  
{  
    public:  
        cl1(int size_of_array);  
        void PrintArray();  
        ~cl1();  
        int* getter();  
        void setter(int* temp);  
    private:  
        int* array;  
};  
#endif
```

## 5.3 Файл main.cpp

*Листинг 3 – main.cpp*

```
#include <iostream>  
#include "cl1.h"  
using namespace std;  
int main()  
{  
    int data;  
    cin >> data;  
    cl1 Object1(data);  
    cin >> data;  
    cl1 Object2(data);  
    int* temp = Object1.getter();  
    Object1 = Object2;  
    Object1.setter(temp);  
    Object1.PrintArray();  
    cout << endl;  
    Object2.PrintArray();  
    cout << endl;  
    return(0);  
}
```



## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
3 4	3?	3?
4 6	4?	4?
5 6	5 5 5 5 5 6 6 6 6 6 6	5 5 5 5 5 6 6 6 6 6 6
7 9	7 7 7 7 7 7 7 9 9 9 9 9 9 9 9 9	7 7 7 7 7 7 7 9 9 9 9 9 9 9 9 9



## 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).