

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.....	8
3 ОПИСАНИЕ АЛГОРИТМОВ.....	9
3.1 Алгоритм конструктора класса Array.....	9
3.2 Алгоритм деструктора класса Array.....	9
3.3 Алгоритм метода print класса Array.....	10
3.4 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	18
5.1 Файл Array.cpp.....	18
5.2 Файл Array.h.....	18
5.3 Файл main.cpp.....	19
6 ТЕСТИРОВАНИЕ.....	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	22

1 ПОСТАНОВКА ЗАДАЧИ

Сконструировать систему, которая демонстрирует особенности выполнения оператора присвоения объектов. Показывает, какого характера ошибки могут возникнуть при присвоении объектов и как их обойти.

Спроектировать объект (разработать описание класса), который имеет указатель на массив целого типа в открытом доступе.

Функционал объекта:

- параметризованный конструктор с параметром целого типа. Конструкторе создается массив целого типа количеством элементов равным значению параметра. Каждому элементу массива присваивается это же значение. Если значение параметра меньше 4, то массив не создается и указателю присваивается значение нулевого указателя;
- выводит значения элементов целочисленного массива. Вывод производит последовательно, разделяя значения двумя пробелами.

Требование к системе: каждый объект разработанного класса должен иметь собственный массив элементов. Алгоритм сборки и отработки системы:

1. Вводит целочисленное значение.
2. Создает первый объект с помощью оператора new.
3. Вводит целочисленное значение.
4. Объявляет второй объект.
5. Если массив в одном или в обоих объектах не создан, то система выводит соответствующее сообщение и переход к пункту 12.
6. Выводит содержимое массивов первого и второго объекта.
7. Выполняет действия, для подготовки исправления ошибки, возникающей при присвоении объектов.
8. Присваивает первому объекту второй объект.

9. Выводит содержимое массивов первого и второго объекта.
10. Выполняет действия, для исправления возникшей ошибки.
11. Выводит содержимое массивов первого и второго объекта.
12. Удаляет созданный объект.
13. Завершает работу.

1.1 Описание входных данных

Первая строка:

«Целое число»

Вторая строка:

«Целое число»

Пример.

5
8

1.2 Описание выходных данных

Если система была собрана (построена), то в первой, третьей и пятой строке:

Array of the first object: «Целое число» «Целое число» . . .

Во второй, четвертой и шестой строке:

Array of the second object: «Целое число» «Целое число» . . .

Если система не была построена, то в первой строке выводится сообщение и некорректное значение количества элементов массива:

The number of array elements is incorrect: «Целое число»?

Пример вывода.

Array of the first object: 5 5 5 5 5

Array of the second object: 8 8 8 8 8 8 8 8
Array of the first object: 8 8 8 8 8 8 8 8
Array of the second object: 8 8 8 8 8 8 8 8
Array of the first object: 5 5 5 5 5
Array of the second object: 8 8 8 8 8 8 8 8

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект `arr1` класса `Array` предназначен для копирования в себя значение другого объекта;
- объект `arr2` класса `Array` предназначен для копирования своих значений в другой объект;
- функция `main` для работы основного алгоритма;
- оператор стандартного потока ввода данных - `cin`;
- оператор стандартного потока вывода данных - `cout`;
- оператор перехода на новую строку - `endl`;
- операторы логического ветвления - `if/else`;
- оператор цикла с параметром - `for`;
- оператор выделения динамической памяти - `new`;
- операторы освобождения динамической памяти - `delete, delete[]`.

Класс `Array`:

- свойства/поля:
 - поле указатель на массив:
 - наименование — `p_arr`;
 - тип — `int*`;
 - модификатор доступа — `public`;
- функционал:
 - метод `Array` — заполнение массива в зависимости от аргумента;
 - метод `~Array` — удаление создаваемых указателей;
 - метод `print` — вывод значений массива.

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 Array

Функционал: заполнение массива в зависимости от аргумента.

Параметры: int n - значение, который будет заполняться массив.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Array

№	Предикат	Действия	№ перехода
1	n >= 4?	создание динамического массива длиной n по указателю p_arg	2
			Ø
2		инициализация итератора i значением 1	3
3	i < n?	присвоение массиву по указателю p_arg по индексу i значением переменной n	4
			Ø
4		инкремент итератора i	3

3.2 Алгоритм деструктора класса Array

Функционал: удаление создаваемых указателей.

Параметры: нет.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса Array

№	Предикат	Действия	№ перехода
1		освобождение памяти по указателю p_arr	Ø

3.3 Алгоритм метода print класса Array

Функционал: вывод значений массива.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода print класса Array

№	Предикат	Действия	№ перехода
1		инициализация итератора i со значением 1	2
2	i < p_arr[n]?	вывод значения массива по указателю p_arr по индексу i	3
			Ø
3	не вывелся последний элемент?	вывод двух пробелов в консоль	4
			Ø
4		инкремент итератора i	2

3.4 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое, индикатор работоспособности программы.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		объявление переменных digit1, digit2 типа int	2
2		ввод значения переменной digit1	3
3		создание объекта класса Array с аргументом digit1 по указателю arr1	4
4		ввод значения переменной digit2	5
5		объявление объекта arr2 класса Array с аргументом digit2	6
6	оба объекта объявились корректно?		7
			28
7		вывод строки "Array of the first object: " и значения arr1	8
8		вызов метода print у объекта по указателю arr1	9
9		переход на следующую строку	10
10		вывод строки "Array of the second object: " и значения arr2	11
11		вызов метода print у объекта по указателю arr2	12
12		переход на следующую строку	13
13		создание указателя p_temp_arr указывающего на значение поля p_arr объекта по указателю arr1	14
14		присвоение объекту по указателю arr1 объекту arr2	15
15		вывод строки "Array of the first object: " и значения arr1	16
16		вызов метода print у объекта по указателю arr1	17
17		переход на новую строку	18
18		вывод строки "Array of the second object: " и значения arr2	19

№	Предикат	Действия	№ перехода
19		вызов у объекта arr2 метода print	20
20		переход на следующую строку	21
21		присвоение полю p_arr объекта по указателю arr1 значения по указателю p_temp	22
22		вывод строки "Array of the first object: " и значения arr1	23
23		вызов метода print у объекта по указателю arr1	24
24		переход на новую строку	25
25		вывод строки "Array of the second object: " и значения arr2	26
26		вызов у объекта arr2 метода print	27
27		переход на следующую строку	28
28	корректно инициализирован массив у объекта по указателю arr1?	вывод строки "The number of array elements is incorrect: ", переменной digit1 и символа "?"	29
			29
29	корректно инициализирован массив у объекта по указателю arr1?	вывод строки "The number of array elements is incorrect: ", переменной digit2 и символа "?"	30
			30
30		освобождение памяти по указателю arr1	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-5.

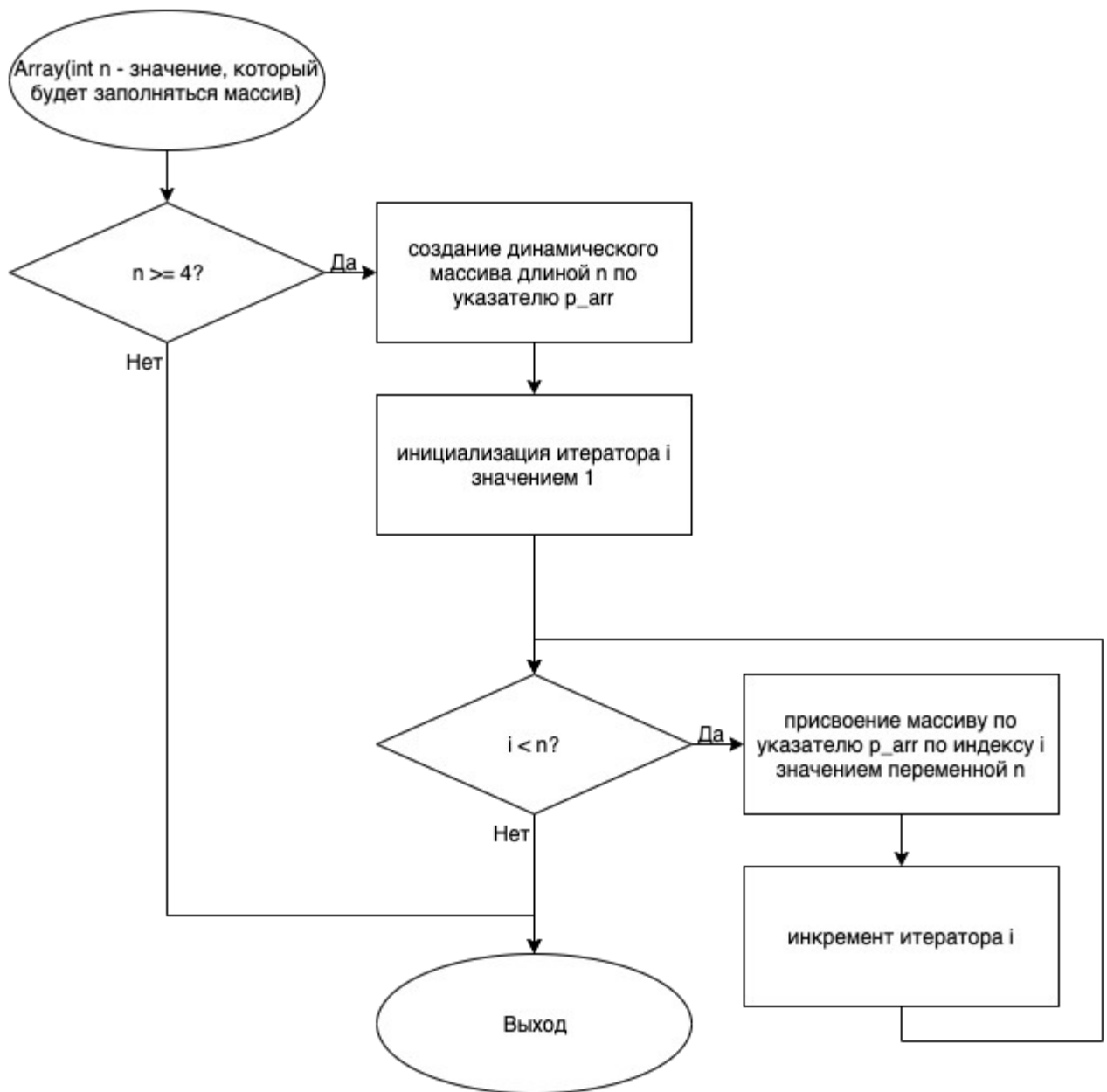


Рисунок 1 – Блок-схема алгоритма

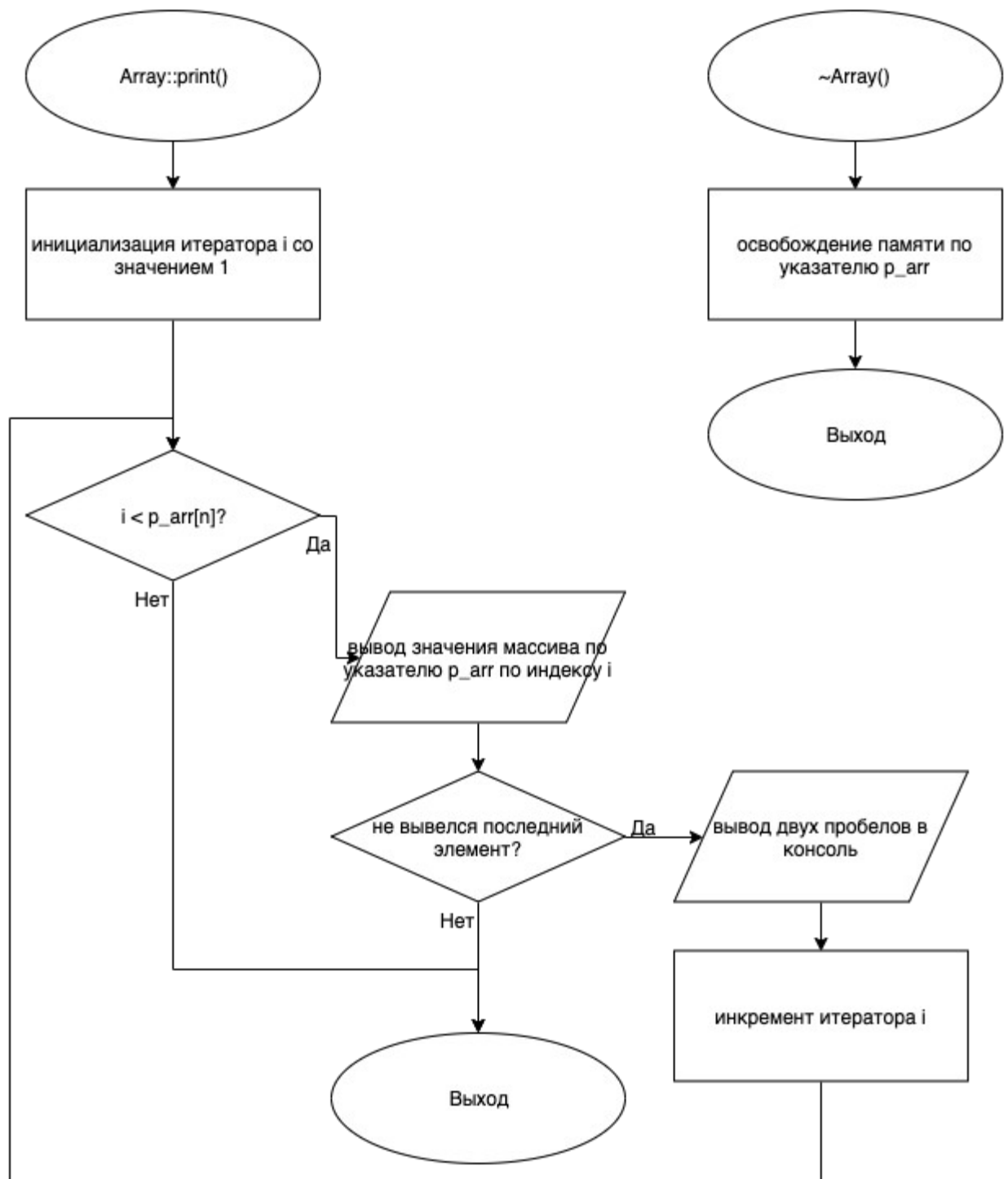


Рисунок 2 – Блок-схема алгоритма



Рисунок 3 – Блок-схема алгоритма

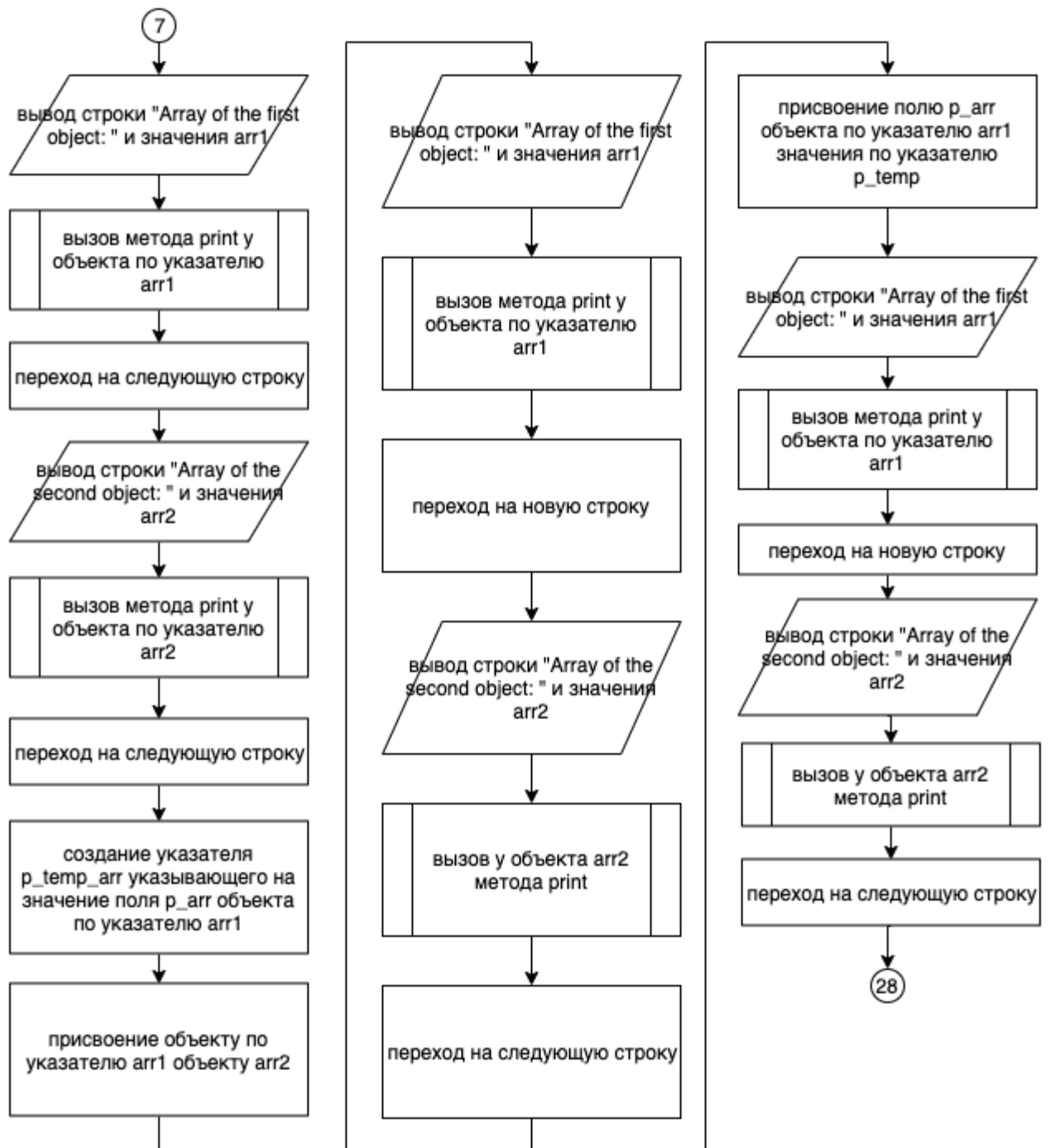


Рисунок 4 – Блок-схема алгоритма

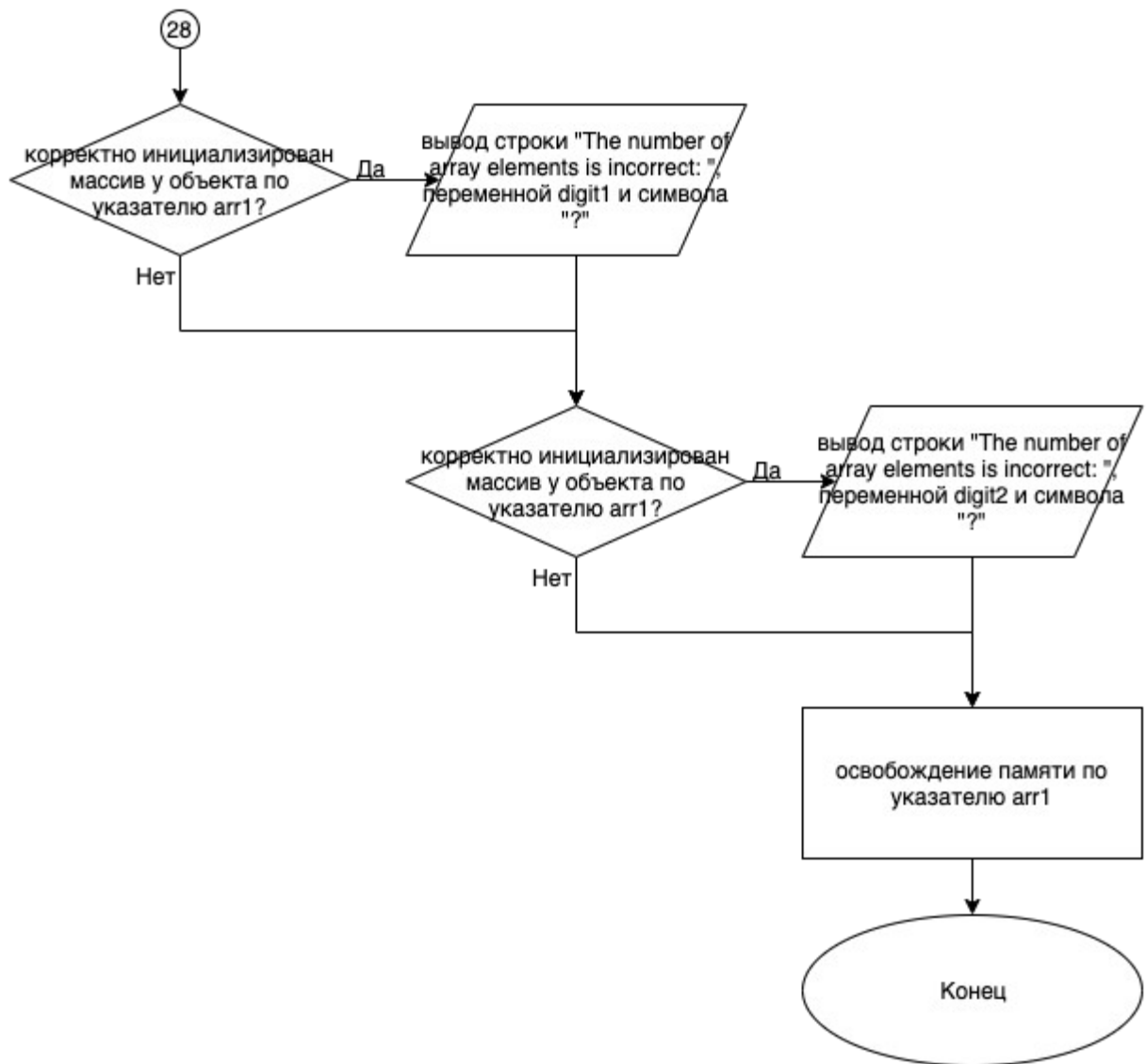


Рисунок 5 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Array.cpp

Листинг 1 – Array.cpp

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

Array::Array(int n){
    if (n >= 4){
        p_arr = new int[n];
        for (unsigned i = 0; i < n; i++){
            p_arr[i] = n;
        }
    }
}

void Array::print(){
    for (unsigned i = 0; i < p_arr[0]; i++){
        cout << p_arr[i];
        if (i != p_arr[0] - 1)
            cout << " ";
    }
}

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

5.2 Файл Array.h

Листинг 2 – Array.h

```
#ifndef __ARRAY__H
#define __ARRAY__H

class Array{
public:
```



```

    int* p_arr = nullptr;

    Array(int);
    ~Array();
    void print();
};

#endif

```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```

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

int main()
{
    int digit1, digit2;
    cin >> digit1;
    Array* arr1 = new Array(digit1);

    cin >> digit2;
    Array arr2(digit2);

    if (arr1->p_arr != nullptr && arr2.p_arr != nullptr){
        cout << "Array of the first object: "; arr1->print(); cout << endl;
        cout << "Array of the second object: "; arr2.print(); cout << endl;

        int* p_temp_arr = arr1->p_arr;
        (*arr1) = arr2;

        cout << "Array of the first object: "; arr1->print(); cout << endl;
        cout << "Array of the second object: "; arr2.print(); cout << endl;

        arr1->p_arr = p_temp_arr;

        cout << "Array of the first object: "; arr1->print(); cout << endl;
        cout << "Array of the second object: "; arr2.print(); cout << endl;

    } else {
        if (!arr1->p_arr)
            cout << "The number of array elements is incorrect: " << digit1 <<
            "?" << endl;
        if (!arr2.p_arr && digit1 != digit2)
            cout << "The number of array elements is incorrect: " << digit2 <<
            "?" << endl;
    }
    delete arr1;
}

```

```
    return(0);  
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 8	Array of the first object: 5 5 5 5 5 Array of the second object: 8 8 8 8 8 8 8 8 Array of the first object: 8 8 8 8 8 8 8 8 Array of the second object: 8 8 8 8 8 8 8 8 Array of the first object: 5 5 5 5 5 Array of the second object: 8 8 8 8 8 8 8 8	Array of the first object: 5 5 5 5 5 Array of the second object: 8 8 8 8 8 8 8 8 Array of the first object: 8 8 8 8 8 8 8 8 Array of the second object: 8 8 8 8 8 8 8 8 Array of the first object: 5 5 5 5 5 Array of the second object: 8 8 8 8 8 8 8 8
3 2	The number of array elements is incorrect: 3? The number of array elements is incorrect: 2?	The number of array elements is incorrect: 3? The number of array elements is incorrect: 2?
9 1	The number of array elements is incorrect: 1?	The number of array elements is incorrect: 1?

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).