

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.
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 демонстрации работы программы;
- функция `main` для определения входной точки программы;
- указатель;
- класс;
- заголовочный файл.

Класс `Array`:

- свойства/поля:
 - поле :
 - наименование — `p_arr`;
 - тип — `int*`;
 - модификатор доступа — `public`;
- функционал:
 - метод `Array` — создание объекта;
 - метод `~Array` — удаление объекта;
 - метод `print` — выводит содержимое 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: Создание объекта.

Параметры: int n.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Array

№	Предикат	Действия	№ перехода
1	$n \geq 4$	Создание динамического массива длиной n указателя p_arr	2
			Ø
2		Инициализация $i = 0$	3
3	$i < n$	$p_arr[i] = n$	4
			Ø
4		$i++$	5
5			Ø

3.2 Алгоритм деструктора класса Array

Функционал: Освобождение памяти массива.

Параметры: нет.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса Array

№	Предикат	Действия	№ перехода
1		Освобождение памяти указателя p_arr	∅

3.3 Алгоритм метода print класса Array

Функционал: Выводит содержимое массива.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода print класса Array

№	Предикат	Действия	№ перехода
1		Инициализация i = 0	2
2	i < p_arr[0]	Вывод p_arr[i]	3
			∅
3	i != p_arr[0] - 1	Вывод двух пробелов	2
			∅

3.4 Алгоритм функции main

Функционал: Входная точка программы.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление переменных digit1, digit2 типа int	2
2		Ввод digit1	3

№	Предикат	Действия	№ перехода
3		Создание объекта класса Array с аргументом digit1 указателя arr1	4
4		Ввод digit2	5
5		Объявление объекта arr2 класса Array с аргументом digit2	6
6	Оба объекта != nullptr		7
			28
7		Вывод "Array of the first object: "	8
8		Вызов arr1->print()	9
9		Вывод "\n"	10
10		Вывод в консоль строки "Array of the second object: "	11
11		Вызов у объекта arr2 метода print	12
12		Вывод в консоль перехода 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 объекта arr2	15
15		Вывод в консоль строки "Array of the first object: "	16
16		Вызов метода print у объекта по указателю arr1	17
17		Вывод в консоль перехода на следующую строку	18
18		Вывод в консоль строки "Array of the second object: "	19
19		Вызов у объекта arr2 метода print	20
20		Вывод "\n"	21
21		Присвоение полю p_arr объекта по указателю arr1 значения по указателю p_temp	22

№	Предикат	Действия	№ перехода
22		Вывод "Array of the first object: "	23
23		Вызов метода print у объекта по указателю arr1	24
24		Вывод "\n"	25
25		Вывод "Array of the second object: "	26
26		Вызов у объекта arr2 метода print	27
27		Вывод "\n"	28
28	У объекта по указателю arr1 корректно инициализировался массив	Вывод в консоль строки "The number of array elements is incorrect: ", переменной digit1 и символа '?'	29
			29
29	У объекта arr2 корректно инициализировался массив	Вывод в консоль 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.

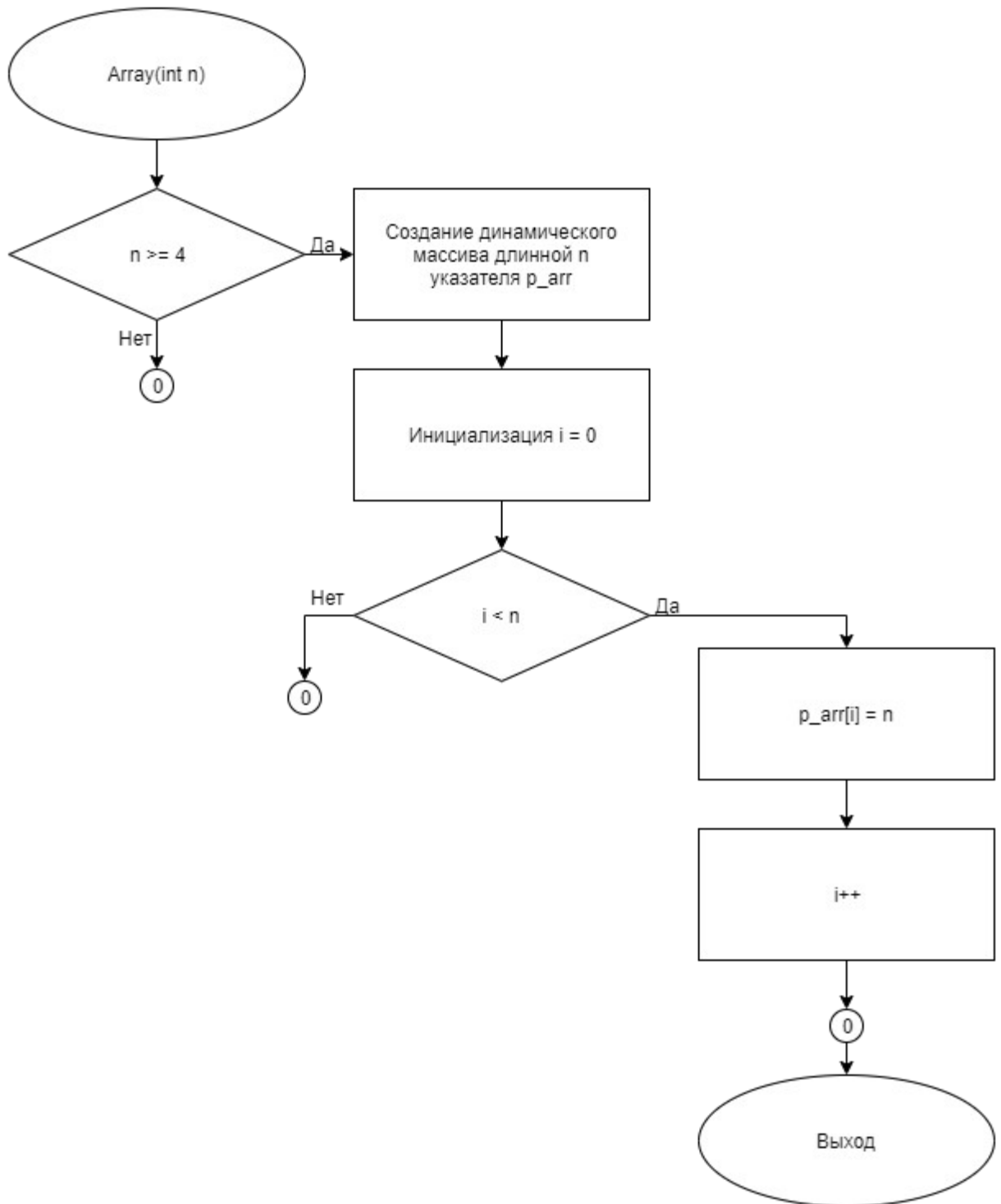


Рисунок 1 – Блок-схема алгоритма

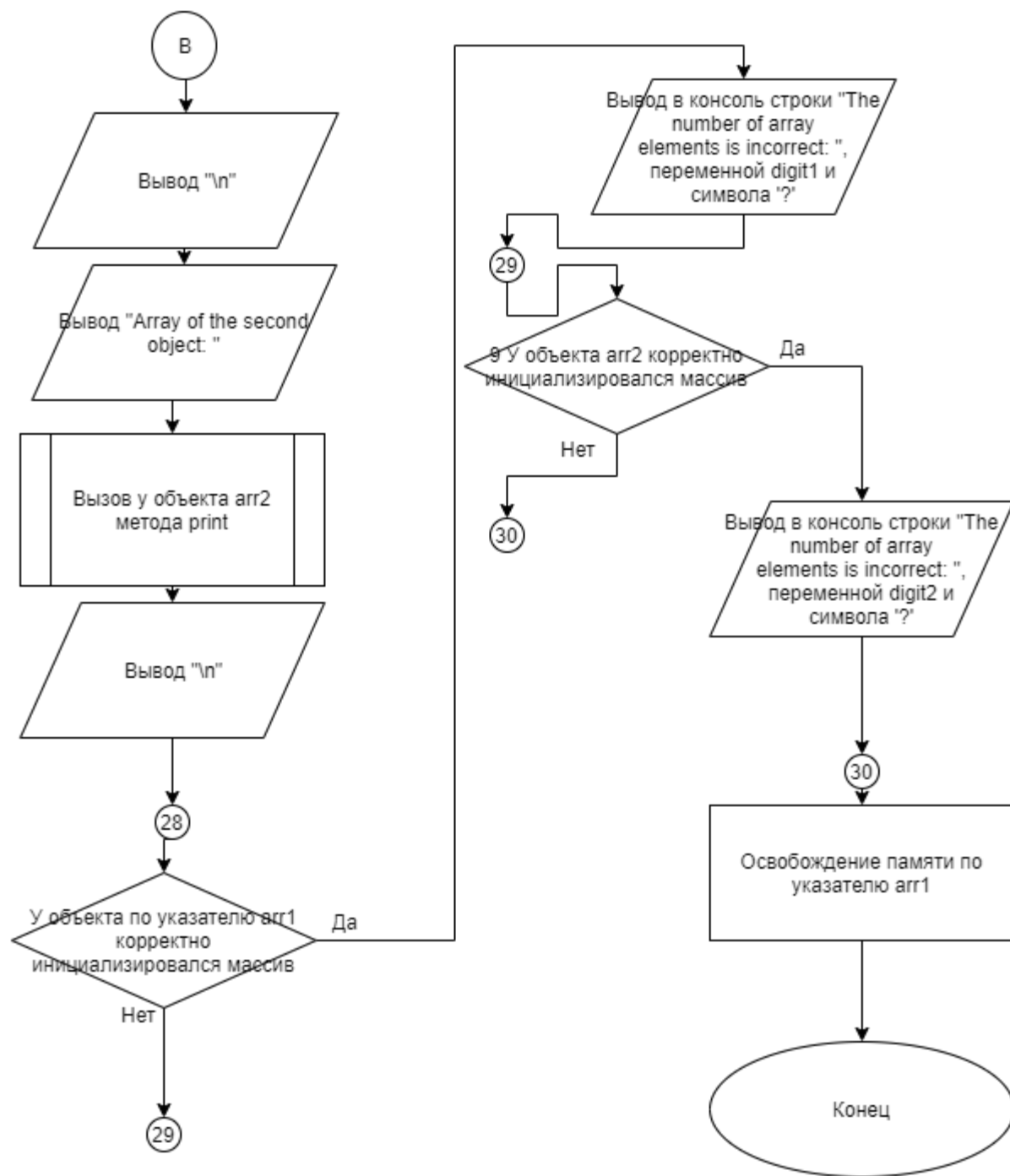


Рисунок 2 – Блок-схема алгоритма

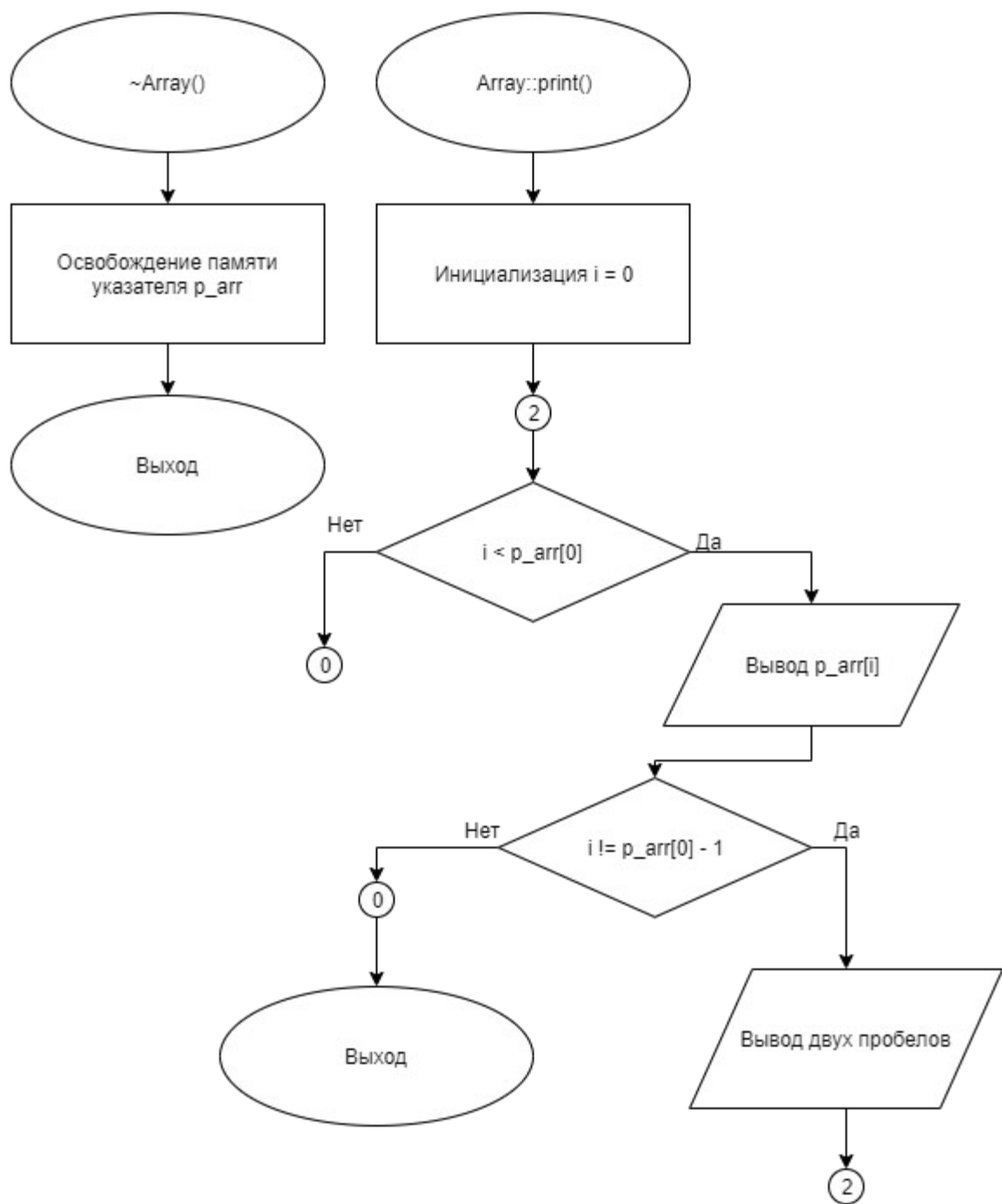


Рисунок 3 – Блок-схема алгоритма

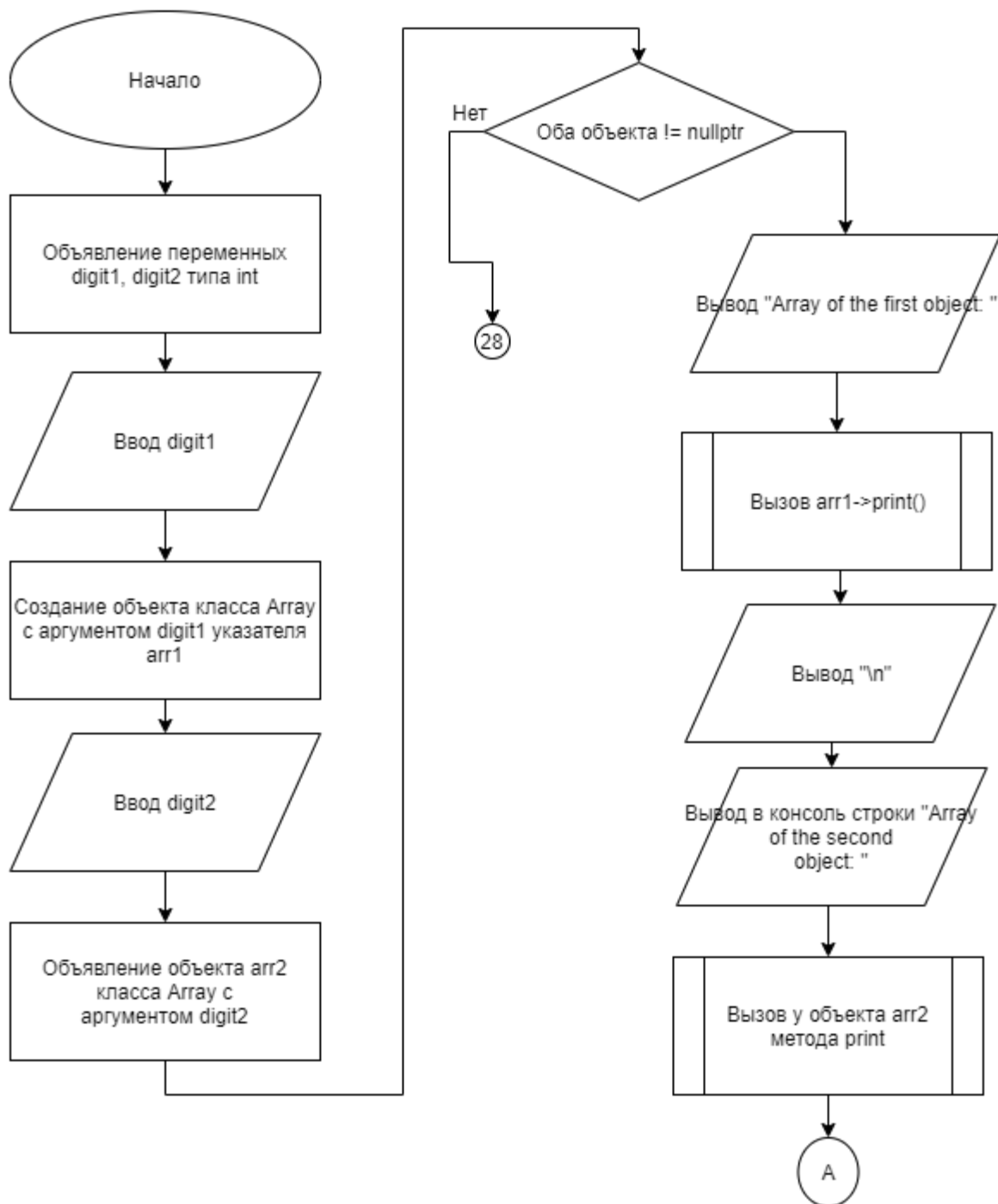


Рисунок 4 – Блок-схема алгоритма

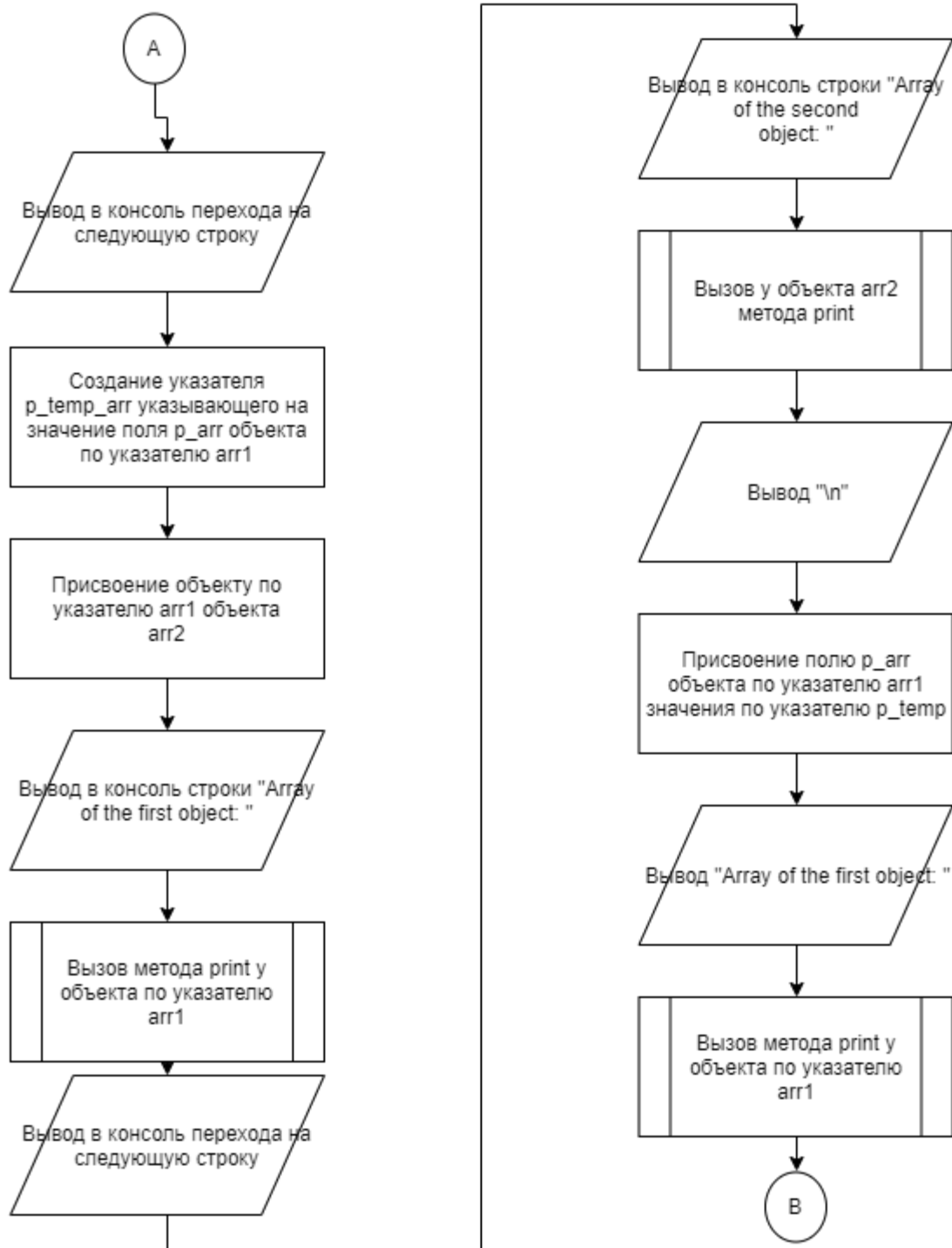


Рисунок 5 – Блок-схема алгоритма

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>

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

void Array::print()
{
    for (int i = 0; i < p_arr[0]; i++)
    {
        std::cout << p_arr[i];
        if (i != p_arr[0] - 1) std::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 <iostream>
#include "Array.h"

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

    std::cin >> digit2;
    Array arr2(digit2);

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

        std::cout << "Array of the second object: ";
        arr2.print();
        std::cout << std::endl;

        int* p_temp_arr = arr1->p_arr;
        *arr1 = arr2;
        std::cout << "Array of the first object: ";
        arr1->print();
        std::cout << std::endl;

        std::cout << "Array of the second object: ";
        arr2.print();
        std::cout << std::endl;

        arr1->p_arr = p_temp_arr;
    }
}

```

```

        std::cout << "Array of the first object: ";
        arr1->print();
        std::cout << std::endl;
        std::cout << "Array of the second object: ";
        arr2.print();
        std::cout << std::endl;
    }
    else
    {
        if (!arr1->p_arr)
        {
            std::cout << "The number of array elements is incorrect: " << digit1
<< '?' << std::endl;
        }

        if (!arr2.p_arr && digit1 != digit2)
        {
            std::cout << "The number of array elements is incorrect: " << digit2
<< '?' << std::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

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).