



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**« МИРЭА Российский технологический университет»**

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Вычислительной техники

**УЧЕБНОЕ ЗАДАНИЕ**

по дисциплине

« Объектно-ориентированное программирование»

Наименование задачи:

**« Задание 2\_1\_3 »**

С тудент группы

ИКБО-13-21

Черномуров С.А.

Руководитель практики

Ассистент

Асадова Ю.С.

Работа представлена

«\_\_»\_\_\_\_\_ 2022 г.

\_\_\_\_\_

(подпись студента)

Оценка

\_\_\_\_\_

(подпись руководителя)

Москва 2022

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
Постановка задачи.....	5
Метод решения.....	7
Описание алгоритма.....	9
Блок-схема алгоритма.....	13
Код программы.....	15
Тестирование.....	17
ЗАКЛЮЧЕНИЕ.....	18
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	19

## **ВВЕДЕНИЕ**

## Постановка задачи

Создать объект, который обрабатывает массив целых чисел не более 10 элементов.

Количество элементов определяются в момент конструирования объекта.

Объект обладает следующей функциональностью:

- в конструкторе считывает значение количества элементов массива, выводит значение количества элементов;
- считывает значения элементов массива;
- выводит значения элементов массива;
- разворачивает последовательность значений элементов массива.

Написать программу, которая:

1. Создает объект и в конструкторе считывает количество элементов массива;
2. Считывает элементы массива;
3. Выводит значения элементов массива согласно исходной последовательности;
4. Разворачивает элементы массива;
5. Выводит значения элементов массива согласно новому их порядку следования.

## Описание входных данных

**Первая**

**строка:**

целое число в десятичном формате.

**Вторая**

**строка:**

последовательность целых чисел в десятичном формате разделенных пробелом.

## Описание выходных данных

**Первая**

**строка:**

N = «количество элементов»

**Вторая строка** (исходный порядок следования элементов):

Значения элементов массива, значение каждого элемента занимает 5 позиции,

выравнивание по правому краю.

**Третья строка** (порядок следования элементов после разворота):

Значения элементов массива, значение каждого элемента занимает 5 позиции,

выравнивание по правому краю.

## Метод решения

Для решения задачи используются:

- Объекты стандартных потоков ввода и вывода `cin` и `cout` соответственно. Используются для ввода с клавиатуры и вывода на экран.
- Объект `arr` класса `Array`. Используется для создания объекта.
- Оператор цикла со счетчиком `for`. Используется для последовательного ввода с клавиатуры и вывода на экран значений элементов массива.
- Функция `reverse`, принадлежит библиотеке `algorithm`. Используется для разворота массива.
- Манипулятор потока ввода/вывода `setw`, принадлежит библиотеке `iomanip`. Используется для форматирования вывода.
- **Класс `Array`:**
  - Свойства поля:
    - Поле:
      - Наименование - `n`;
      - Тип - целочисленный;
      - Модификатор доступа - закрытый.
    - Поле:
      - Наименование - `*mas`;
      - Тип - указатель на целочисленный динамический массив;
      - Модификатор доступа - закрытый.
  - Методы:
    - Метод `Array`:
      - Функционал - параметризованный конструктор, содержащий значение количества элементов массива,

выводит количество элементов массива на экран.

- Метод SetArr:
  - Функционал - параметризованный метод, считывающий с клавиатуры значения элементов массива.
- Метод CoutArr:
  - Функционал - параметризованный метод, выводящий на экран значения элементов массива.
- Метод ReverseArr:
  - Функционал - параметризованный метод, разворачивающий массив.

## Описание алгоритма

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

Функция: main

Функционал: Основной алгоритм программы

Параметры: Без параметров

Возвращаемое значение: Целочисленный тип данных - код возврата

Алгоритм функции представлен в таблице 1.

Таблица 1. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление переменной n целочисленного типа данных	2	
2		Считывание с клавиатуры значения переменной n	3	
3		Создание объекта arr класса Array путем вызова конструктора с параметром n	4	
4		Вызов метода SetArr объекта arr	5	
5		Вызов метода CoutArr объекта arr	6	
6		Вызов метода ReverseArr объекта arr	7	
7		Вызов метода CoutArr объектов arr	Ø	

Конструктор класса: Array

Модификатор доступа: public



Функционал: Параметризированный конструктор, содержащий значение количества элементов массива, выводит количество элементов массива на экран

Параметры: Переменная N целочисленного типа данных

Алгоритм конструктора представлен в таблице 2.

Таблица 2. Алгоритм конструктора класса Array

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация переменной n значением переменной N	2	
2		Динамическое выделение памяти под n элементов динамического массива mas	3	
3		Вывод на экран "N = ", n с последующим переносом на новую строку	Ø	

Класс объекта: Array

Модификатор доступа: public

Метод: SetArr

Функционал: Параметризированный метод, считывающий с клавиатуры значения элементов массива

Параметры: Без параметров

Возвращаемое значение: Отсутствует

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода SetArr класса Array

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной переменной с инициализацией $i=0$	2	Использование переменной $i$ для счетчика
2	$i$ меньше $n$	Считывание с клавиатуры значения $i$ -го элемента динамического массива $mas$	3	Считывание с клавиатуры значений элементов массива
			Ø	Выход из цикла
3		инкрементирование $i$	2	

Класс объекта: Array

Модификатор доступа: public

Метод: CoutArr

Функционал: Параметризованный метод, выводящий на экран значения элементов массива

Параметры: Без параметров

Возвращаемое значение: Отсутствует

Алгоритм метода представлен в таблице 4.

Таблица 4. Алгоритм метода CoutArr класса Array

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной переменной с инициализацией $i=0$	2	Использование переменной $i$ для счетчика
2	$i$ меньше $n$	Вывод на экран значения $i$ -го элемента динамического массива $mas$	3	Вывод массива на экран
			Ø	Выход из цикла
3		Инкрементирование $i$	2	

Класс объекта: Array

Модификатор доступа: public

Метод: ReverseArr

Функционал: Параметризированный метод, разворачивающий массив

Параметры: Без параметров

Возвращаемое значение: Отсутствует

Алгоритм метода представлен в таблице 5.

Таблица 5. Алгоритм метода ReverseArr класса Array

№	Предикат	Действия	№ перехода	Комментарий
1		Разворот динамического массива mas	Ø	

## Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

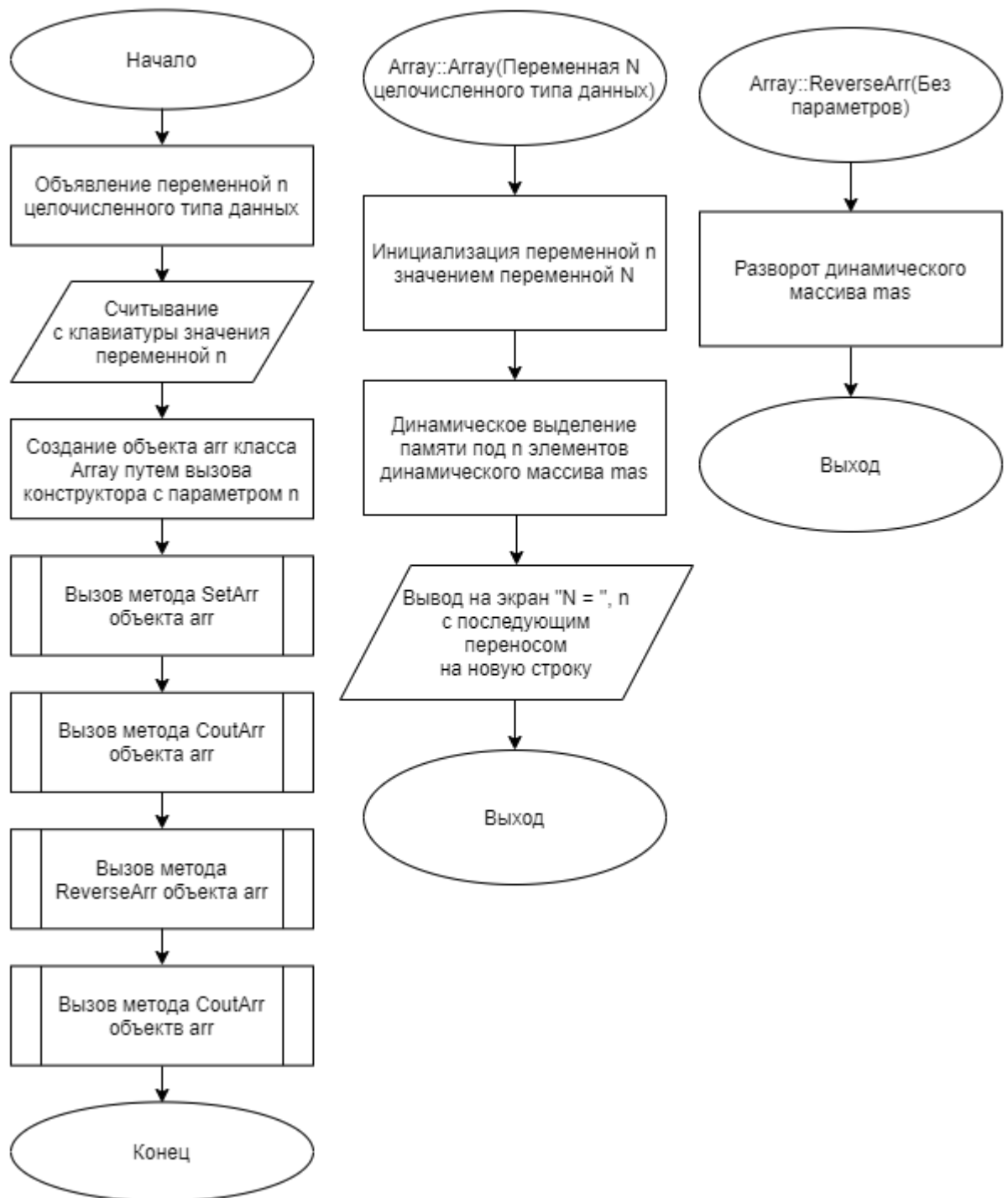


Рис. 1. Блок-схема алгоритма.

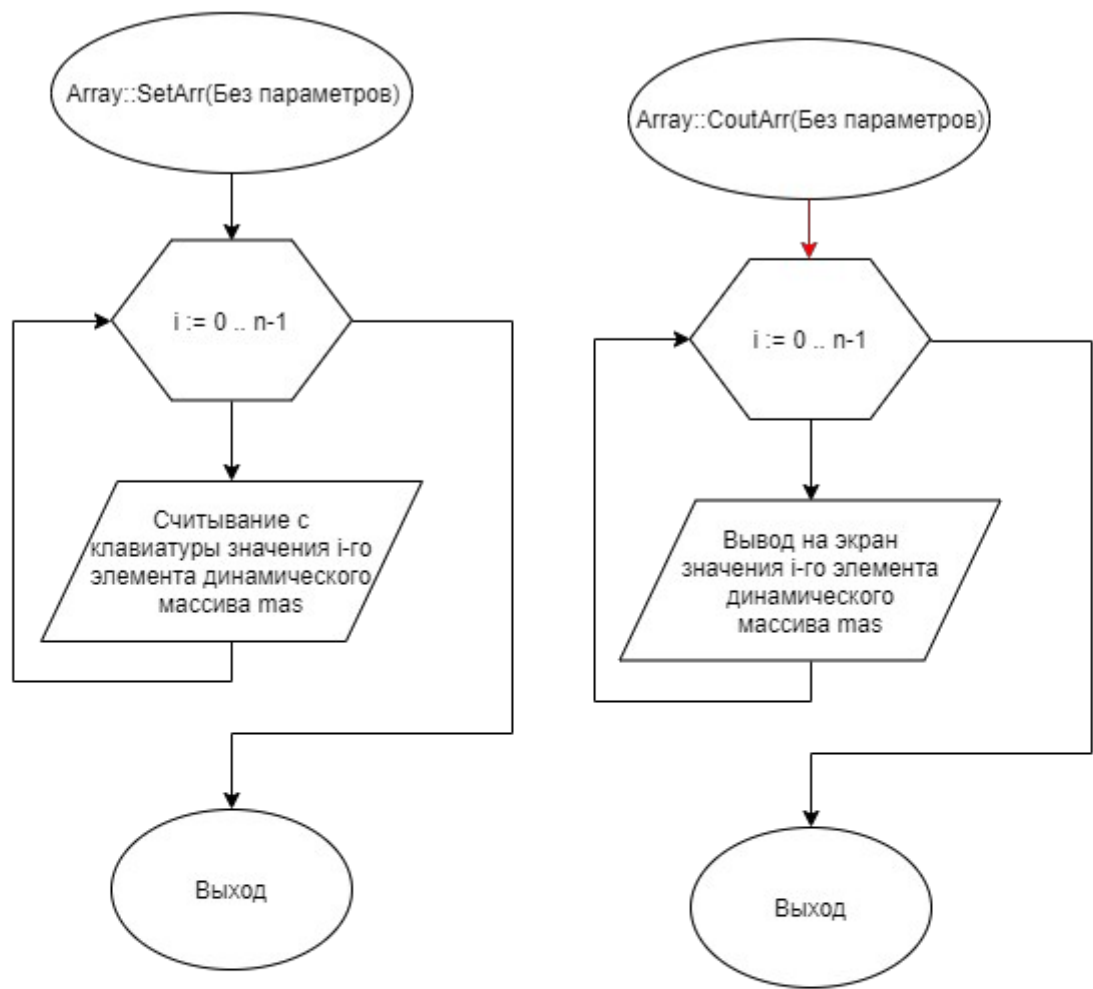


Рис. 2. Блок-схема алгоритма.

## Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

### Файл Array.cpp

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

Array :: Array(int N){
    n = N; // определение длины массива
    mas = new int[n]; // динамическое выделение памяти под массива
    cout << "N = " << n << "\n"; // вывод количества элементов
массива
}

void Array :: SetArr(){
    for (int i = 0; i < n; i++) // ввод элементов массива через
цикл
        cin >> mas[i];
}

void Array :: CoutArr(){
    for (int i = 0; i < n; i++) // форматированный вывод элементов
массива через цикл
        cout << right << setw(5) << mas[i];
}

void Array :: ReverseArr(){
    reverse(mas,mas+n); // разворот массива
}
```

### Файл Array.h

```
#ifndef _ARRA_H
#define _ARRA_H

class Array
{
    private:
        int n; // переменная, отвечающая за длину массива
        int *mas; // указатель на динамический массив

    public:
        Array(int N); // конструктор
```

```

        void SetArr(); // ввод элементов массива
        void CoutArr(); // вывод элементов массива
        void ReverseArr(); // разворот массива
};

#endif

```

## Файл main.cpp

```

#include "Array.h"
#include <iostream>

int main()
{
    int n; // объявление переменной, отвечающей за длину массива

    std :: cin >> n; // инициализация переменной, отвечающей за длину
массива

    Array arr(n); // создание объекта arr класса array, с передачей в него
параметра длины массива
    arr.SetArr(); // ввод элементов массива
    arr.CoutArr(); // вывод элементов массива

    std :: cout<<"\n";

    arr.ReverseArr(); // разворот массива
    arr.CoutArr(); // вывод развернутого массива
}

```

## Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 1 2 3 4 5	N = 5 1 2 3 4 5 5 4 3 2 1	N = 5 1 2 3 4 5 5 4 3 2 1
3 5 8 3	N = 3 5 8 3 3 8 5	N = 3 5 8 3 3 8 5
1 5	N = 1 5 5	N = 1 5 5



## **ЗАКЛЮЧЕНИЕ**

## **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (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-avrrora.ru/student/files/methodichescoe\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avrrora.ru/student/files/methodichescoe_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-avrrora.ru/student/files/Prilozheniye\\_k\\_methodichke.pdf](https://mirea.aco-avrrora.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).