

Здесь будет титульник, листай ниже

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Постановка задачи.....	5
2 Метод решения.....	8
3 Описание алгоритма.....	11
4 Блок-схема алгоритма.....	12
5 Код программы.....	14
6 Тестирование.....	18
ЗАКЛЮЧЕНИЕ.....	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	21

# 1 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.

## 1.1 Описание входных данных

**Первая строка:**

целое число в десятичном формате.

**Вторая строка:**

последовательность целых чисел в десятичном формате разделенных пробелом.

## **1.2 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.

## 2 МЕТОД РЕШЕНИЯ

Для решения поставленной задачи используется:

объект класса cl1

объект потокового ввода cin

объект потокового вывода cout

функция обмена swap

оператор цикла со счетчиков for

манипулятор потока вывода setw

манипулятор потока вывода right

манипулятор потока вывода setfill

Класс cl1:

Поля:

Доступные:

int length - длина массива

int nums - массив

Методы:

Открытые:

конструктор cl1(int num) - Считывание количества эл-ов массива.

void input() - Считывание эл-ов массива

void reverse() - Разворот последовательности значений эл-ов массива

void output - Вывод значений эл-ов 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 функции main

Функционал: главный метод программы.

Параметры: .

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление целочисленной переменной	2
2		ввод целочисленного значения переменной	3
3		создание объекта класса cl1	4
4		вызов метода считывания значения элементов массива	5
5		вызов метода считывания значений элементов массива	6
6		вызов метода разворота последовательности значений элементов массива	7
7		Вызов метода вывода элементов массива	Ø

### 3.2 Алгоритм конструктора класса cl1

Функционал: считывание количества элементов массива.

Параметры: int n.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса *cl1*

№	Предикат	Действия	№ перехода
1		присвоение параметра к длине массива объекта	2
2		изменение длины массива объекта соответственно параметру	∅

### 3.3 Алгоритм метода *input* класса *cl1*

Функционал: считывание значений элементов массива.

Параметры: .

Возвращаемое значение: *void*.

Алгоритм метода представлен в таблице 3.

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

№	Предикат	Действия	№ перехода
1	считчик меньше длины массива объекта	ввод значения элемента массива	2
			∅
2		прибавление единицы к счетчику	1

### 3.4 Алгоритм метода *reverse* класса *cl1*

Функционал: разворот элементов массива.

Параметры: .

Возвращаемое значение: *void*.

Алгоритм метода представлен в таблице 4.

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

№	Предикат	Действия	№ перехода
1		объявление счетчика <i>en</i>	2
2	счетчик <i>st</i> меньше счетчика	замена местами элементов массива с индексами	3

№	Предикат	Действия	№ перехода
	en	счетчиков	
			∅
3		прибавление единицы к счетчику st	4
4		прибавление единицы к счетчику en	2

### 3.5 Алгоритм метода output класса cl1

Функционал: вывод значений элементов массива.

Параметры: .

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода output класса cl1

№	Предикат	Действия	№ перехода
1	счетчик меньше длины массива объекта	вывод значения элемента счетчика	2
			∅
2		прибавление единицы к счетчику	1



## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

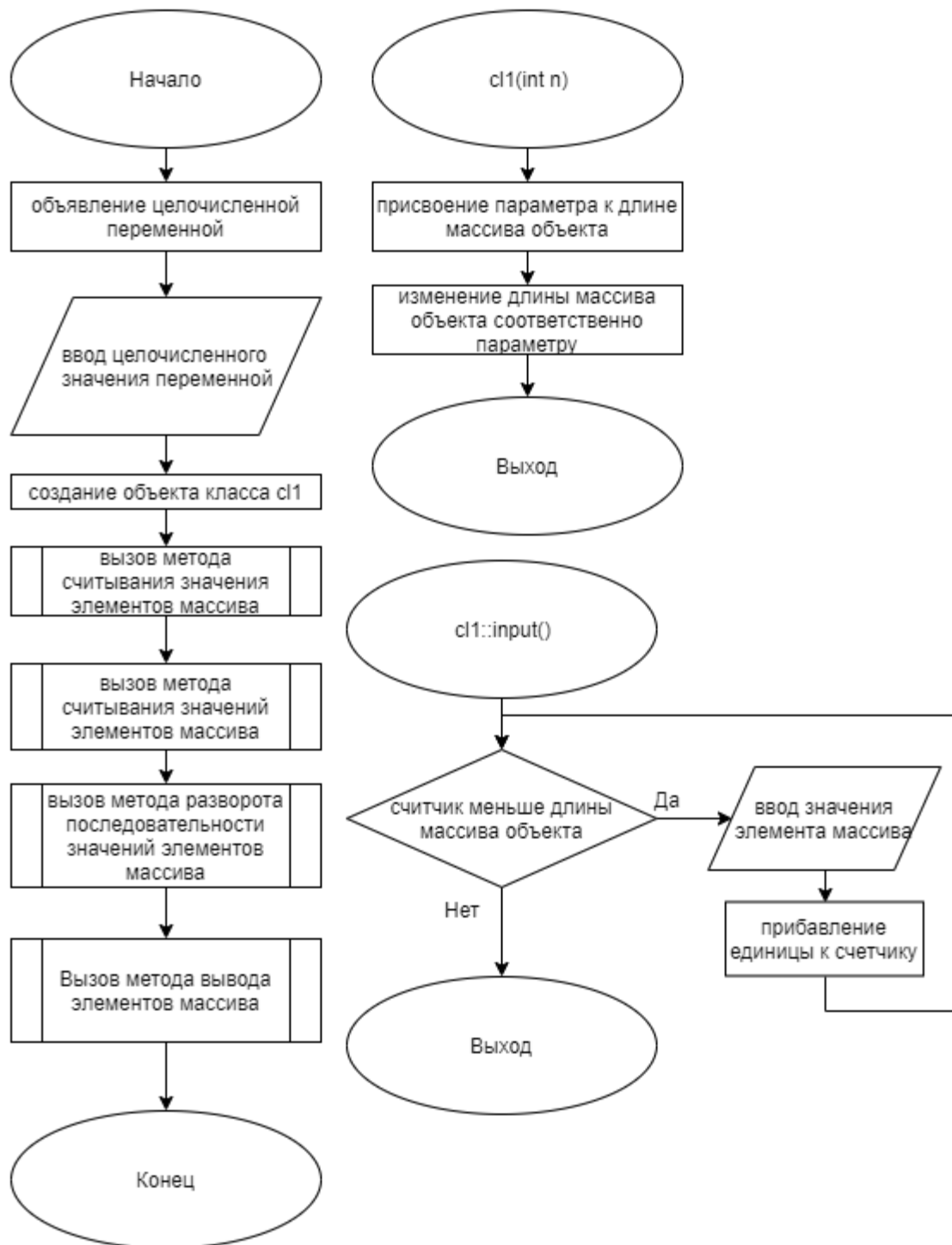


Рисунок 1 – Блок-схема алгоритма

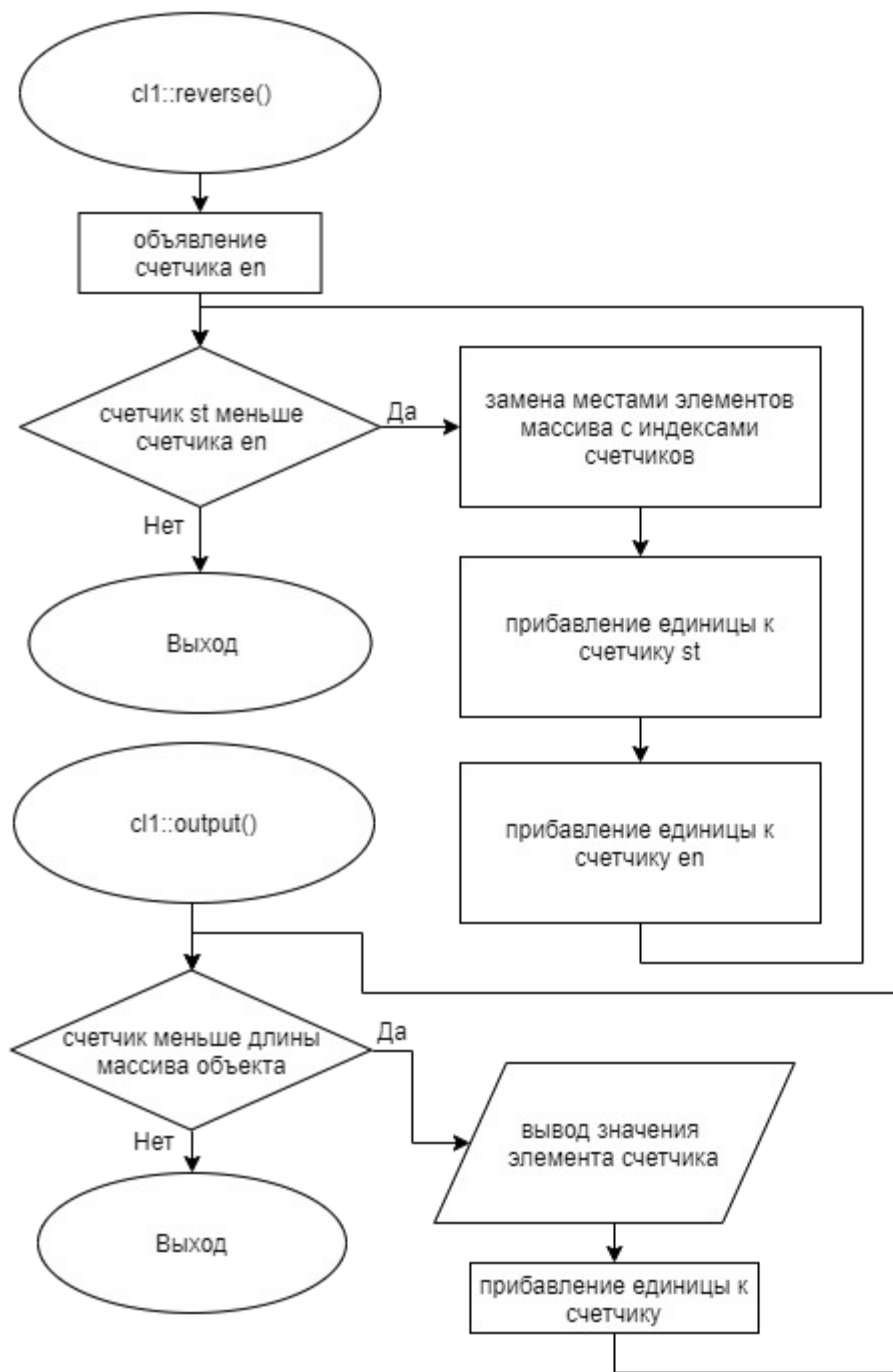


Рисунок 2 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл cl1

*Листинг 1 – cl1*

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <iomanip>
using namespace std;
class cl1{
public:
    int length;
    int nums[10];
    cl1(int n){
        length = n;
        int nums[n];
    }
    void input(){
        for (int i = 0; i < length; i++){
            cin >> nums[i];
        }
    }
    void reverse(){
        for (int st=0, en=length-1; st < en; st++, en--){
            swap(nums[st], nums[en]);
        }
    }
    void output(){
        for (int i = 0; i < length; i++){
            cout << right << setw(5) << setfill(' ') << nums[i];
        }
    }
};
```

### 5.2 Файл main.cpp

*Листинг 2 – main.cpp*

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
```

```
#include "cl1"
using namespace std;
int main()
{
    int n;
    cin >> n;
    cout << "N = " << n << endl;
    cl1 obj(n);
    obj.input();
    obj.output();
    cout << endl;
    obj.reverse();
    obj.output();
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

Таблица 6 – Результат тестирования программы

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

## 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).