

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	5
1.2 Описание выходных данных.....	6
2 МЕТОД РЕШЕНИЯ.....	7
3 ОПИСАНИЕ АЛГОРИТМОВ.....	8
3.1 Алгоритм конструктора класса sort.....	8
3.2 Алгоритм метода set класса sort.....	8
3.3 Алгоритм метода get класса sort.....	9
3.4 Алгоритм метода reverse класса sort.....	9
3.5 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	14
5.1 Файл main.cpp.....	14
5.2 Файл sort.cpp.....	14
5.3 Файл sort.h.....	15
6 ТЕСТИРОВАНИЕ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17

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 задачи используется:

- объект sort класса sort;
- функция main для Основная функция;
- Объект стандартного потока ввода/ вывода cin/cout;
- Оператор цикла со счётчиком;
- объект obj класса sort;
- Функция для форматированного вывода.

Класс sort:

- свойства/поля:
 - поле Количество элементов:
 - наименование — n;
 - тип — int;
 - модификатор доступа — private;
 - поле Массив:
 - наименование — a[i];
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод sort — Конструктор;
 - метод set — Счёт элементов массива;
 - метод get — Вывод элементов массива;
 - метод reverse — Разворачивание элементов 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 sort

Функционал: Конструктор.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса sort

№	Предикат	Действия	№ перехода
1		Ввод целочисленной переменной n	2
2		Вывод "N = " , переход на новую строку	∅

3.2 Алгоритм метода set класса sort

Функционал: Ввод массива.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода set класса sort

№	Предикат	Действия	№ перехода
1		Инициализация целой переменной i = 0	2
2	i < n	Ввод i-го элемента массива	3
			∅
3		i++	2

3.3 Алгоритм метода get класса sort

Функционал: Вывод массива.

Параметры: нет.

Возвращаемое значение: целое.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода get класса sort

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной $i = 0$	2
2	$i < n$	Вывод элементов массива	3
			Ø
3		$i++$	2

3.4 Алгоритм метода reverse класса sort

Функционал: Разворот элементов массива.

Параметры: нет.

Возвращаемое значение: Нет.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода reverse класса sort

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной $b = 0$ и $i = 0$	2
2	$i < n/2$	присваивание а эмента массива b на позиции $b[i]$, находящийся на той же позиции,если начинать отсчёт с конца массива, присваиваем элементу в позиции с конца значение a	3
			Ø

№	Предикат	Действия	№ перехода
3		i++	2

3.5 Алгоритм функции main

Функционал: Основная функция.

Параметры: нет.

Возвращаемое значение: Целое.

Алгоритм функции представлен в таблице 5.

Таблица 5 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Создание объекта obj	2
2		Вызов методов set(), get()	3
3		Вызов перехода на новую строку	4
4		Вызов методов reverse(), get()	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.

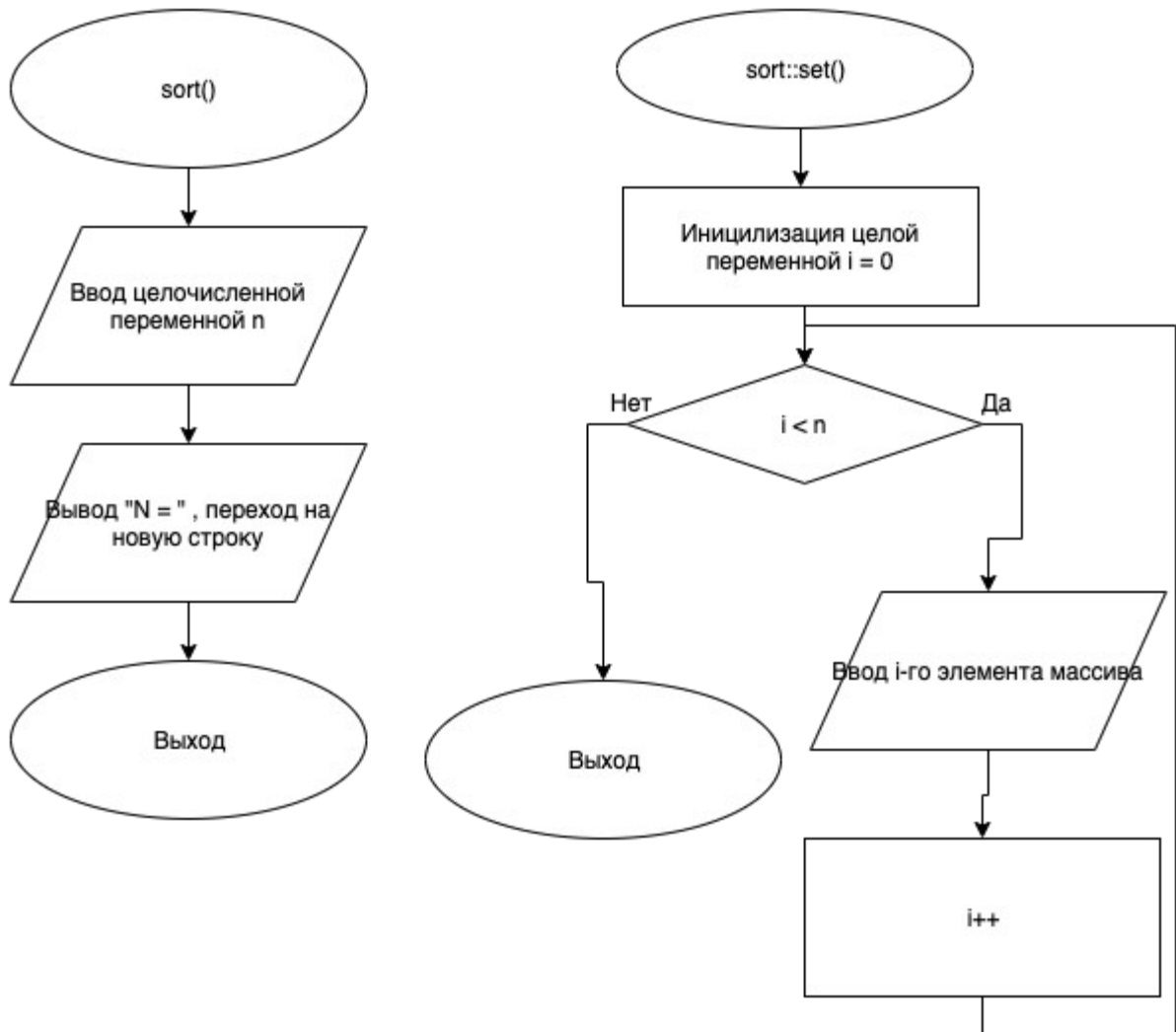


Рисунок 1 – Блок-схема алгоритма

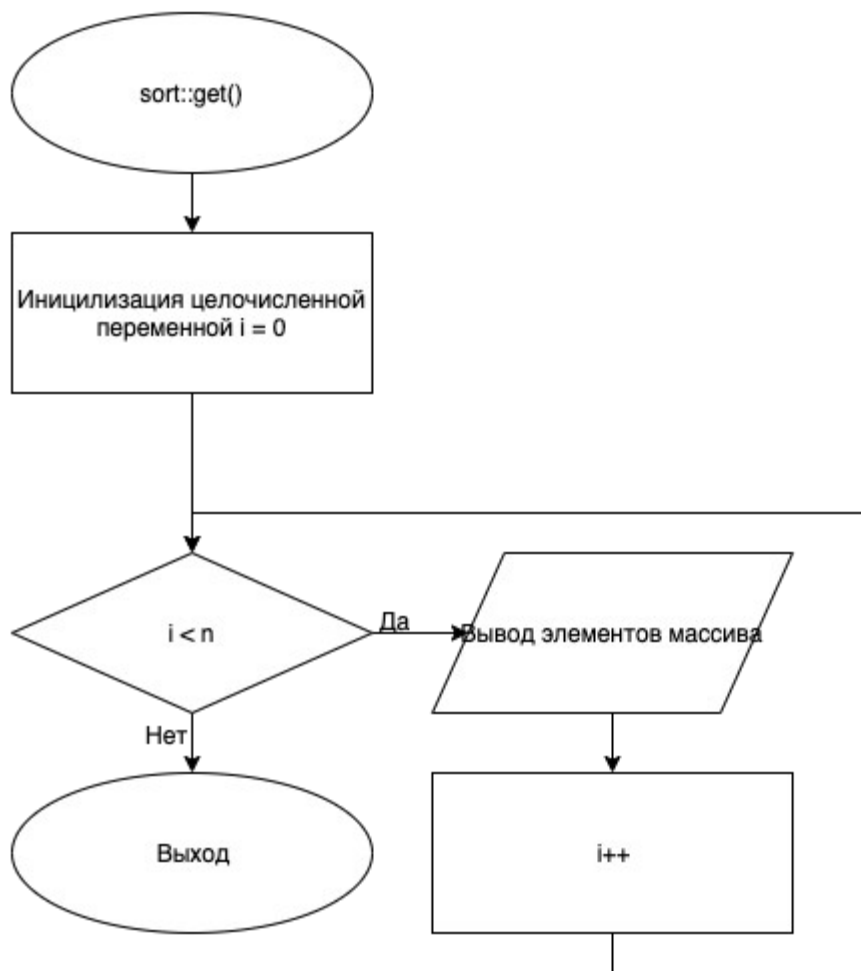


Рисунок 2 – Блок-схема алгоритма

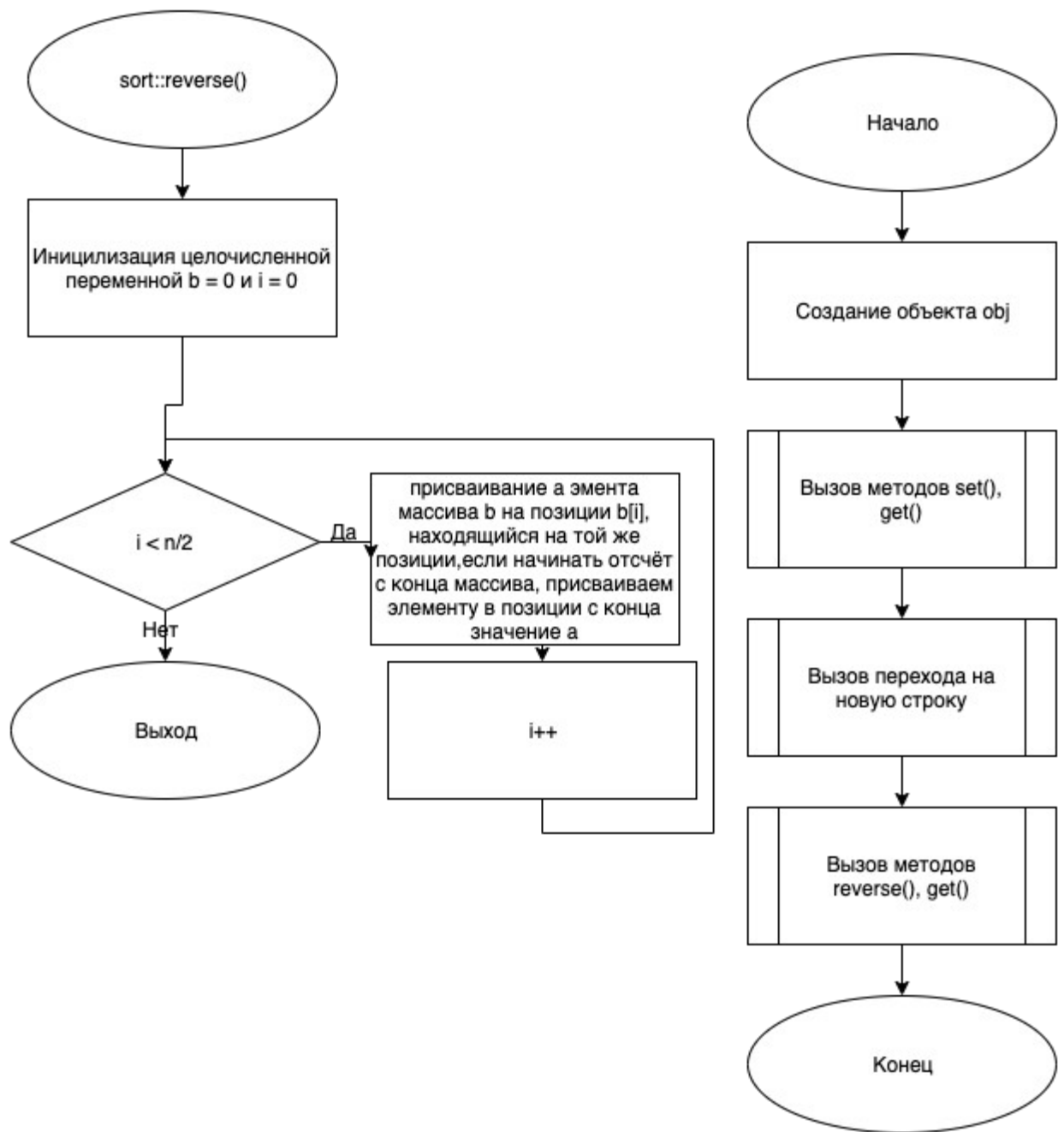


Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "sort.h"
int main()
{
    sort sort;
    sort.set();
    sort.get();

    cout << "\n";

    sort.reverse();
    sort.get();
    return(0);
}
```

5.2 Файл sort.cpp

Листинг 2 – sort.cpp

```
#include "sort.h"
sort::sort()
{
    cin >> n;
    cout << "N = " << n << "\n";
}
void sort::set()
{
    for (int i = 0; i < n; i++)
    {
        cin >> a[i];
    }
}
void sort::get()
```

```

{
    for(int i = 0; i < n; i++)
    {
        printf("%5d", a[i]);
    }
}
void sort::reverse()
{
    int b;
    for(int i = 0; i < n/2; i++)
    {
        b = a[i];
        a[i] = a[n-1-i];
        a[n-1-i] = b;
    }
}

```

5.3 Файл sort.h

Листинг 3 – sort.h

```

#ifndef __SORT__H
#define __SORT__H
#include <iostream>
using namespace std;

class sort
{
private:
    int n, a[10];
public:
    sort();
    void set();
    void get();
    void reverse();
};

#endif

```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

Таблица 6 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1 1	N = 1 1 1	N = 1 1 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).