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 1_2_2 »

С тудент группы	ИКБО-22-20	Елизаров Д.А.
Руководитель практики	Ассистент	Люлява Д.В.
Работа представлена	«___»_____ 2021 г.	_____
		(подпись студента)
Оценка		_____
		(подпись руководителя)

Москва 2021

Постановка задачи

Создать объект, который обрабатывает массив целых чисел не более 10 элементов. Количество элементов определяются в момент конструирования объекта. Объект обладает следующей функциональностью:
- считывает элементы массива;
- выводит элементы массива;
- разворачивает последовательность 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.

Третья строка (порядок следования элементов после разворота): Значения элементов массива, значение каждого элемента занимает 5 позиции.

Метод решения

Для решения задачи будем использовать:

Объекты стандартных потоков ввода/вывода

Стандартная функция swap().

Объект arr класса mas.

Свойства класса mas:

Поле отвечающее за размер массива:

1. Наименование - N.
2. Тип - Целочисленное
3. Модификатор доступа - private

Поле отвечающее за массив:

1. Наименование - arr.
2. Тип данных - Целочисленный динамический массив
3. Модификатор доступа - private

Описание алгоритма

Функция: main

Функционал: Основной алгоритм программы

Параметры: без параметров

Возвращаемое значение: Целое - индикатор корректности завершения программы

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной	2	

		переменной n		
2		Ввод с клавиатуры значения n	3	
3		Выводим на экран "N = "+n+переход на следущую строку	4	
4		Создание объекта arr класса mass и конструирование путем вызова параметризированного конструктора mass с значением переменной n	5	
5		Вызов метода setArr() объекта arr	6	Установка значений массива
6		Вызов метода getArr() объекта arr	7	Вывод значений массива
7		Вызов метода reverseArr() объекта arr	8	Переворот последовательности значений массива
8		Вывод на экран перехода на новую строку	9	
9		Вызов метода getArr() объекта arr	∅	

Класс объекта: mass

Метод: mass

Функционал: Парметризированный конструктор

Параметры: Целое n - размер массива

Возвращаемое значение: Пустое множество значений

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация поля N, принимающего значение параметра n	2	
2		Создание целочисленного динамического массива arr размера N	∅	

Класс объекта: mass

Метод: setArr

Функционал: Инициализация элементов массива

Параметры: Без параметров

Возвращаемое значение: Пустое множество значений

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной переменной с инициализацией i=0	2	Создание переменной счетчика

2	i<N	Ввод с клавиатуры значения arr[i]	3	
			∅	
3		Инкрементируем значение переменной i	2	

Класс объекта: mass

Метод: reverseArr

Функционал: Переворот последовательности значений массива

Параметры: Без параметров

Возвращаемое значение: Пустое множество значений

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной переменной с инициализацией i=0	2	Создание переменной счетчика
2	i<N/2	Используем стандартную функцию swap и меняем значения arr[i] и arr[N-i-1] местами	3	
			∅	
3		Инкрементируем значение переменной i	2	

Класс объекта: mass

Метод: getArr

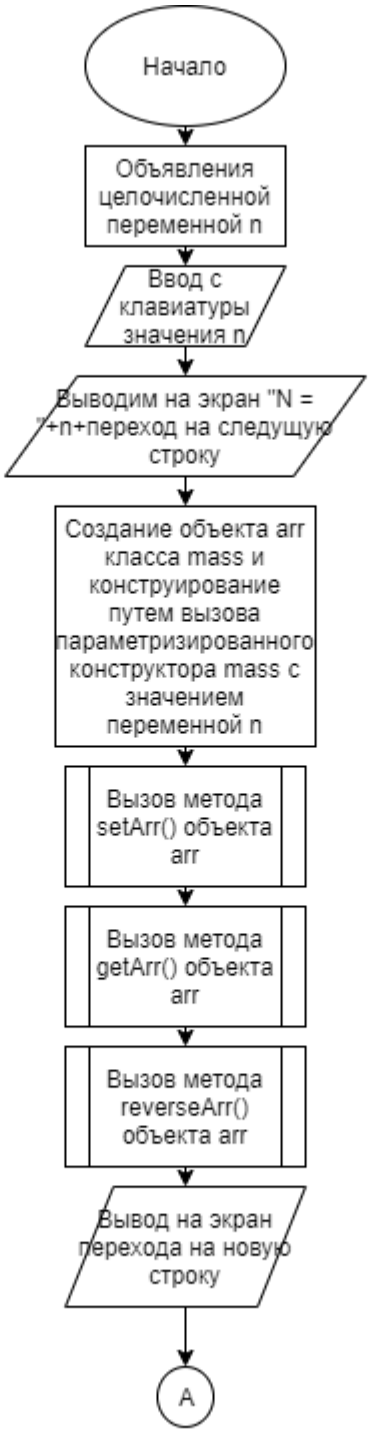
Функционал: Вывод массива на экран

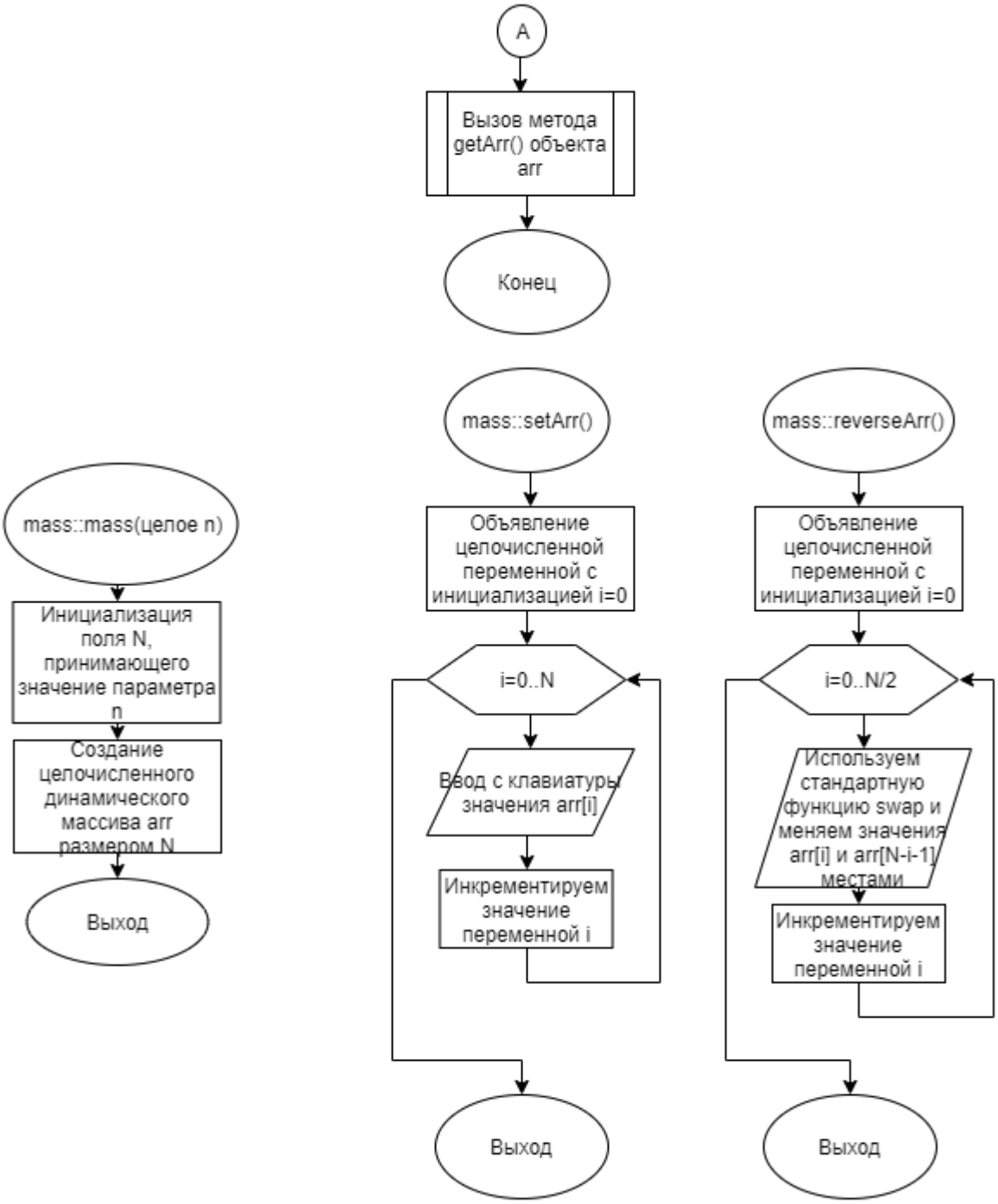
Параметры: Без параметров

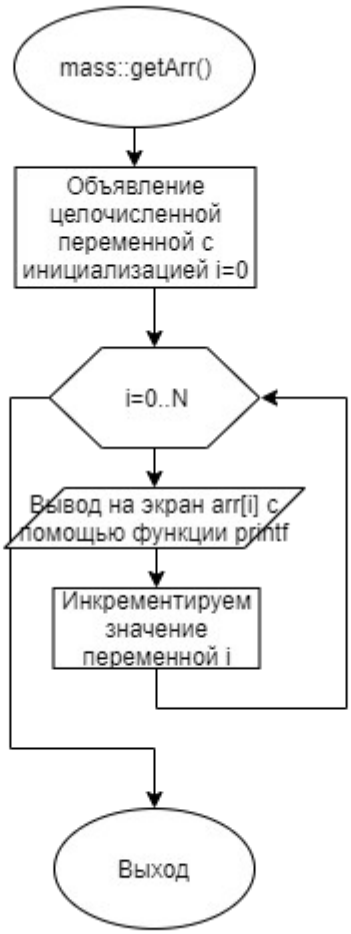
Возвращаемое значение: Без возвращаемого значения

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной переменной с инициализацией i=0	2	
2	i<N	Вывод на экран значения arr[i] с помощью функции printf()	3	Выводим все значения массива, каждое значение занимает 5 позиций
			∅	
3		Инкрементируем значение переменной i	2	

Блок-схема алгоритма




A
Вызов метода getArr() объекта arr
Конец
mass::mass(целое n)
Инициализация поля N, принимающего значение параметра n
Создание целочисленного динамического массива arr размером N
Выход
mass::setArr()
Объявление целочисленной переменной с инициализацией i=0
i=0..N
Ввод с клавиатуры значения arr[i]
Инкрементируем значение переменной i
Выход
mass::reverseArr()
Объявление целочисленной переменной с инициализацией i=0
i=0..N/2
Используем стандартную функцию swap и меняем значения arr[i] и arr[N-i-1] местами
Инкрементируем значение переменной i
Выход



Код программы

Файл main.cpp

```
#include <iostream>
#include "mass.h"

int main()
{
        int n;
        std::cin>>n;
        std::cout<<"N = "<<n<<std::endl;
        mass arr(n);
        arr.setArr();
        arr.getArr();
        arr.reverseArr();
        std::cout<<std::endl;
        arr.getArr();
        return(0);
}
```

Файл mass.cpp

```
#include <iostream>
#include <stdio.h>
#include "mass.h"

mass::mass(int n){
        N=n;
        arr = new int[N];
}
void mass::setArr(){
        for (int i=0;i<N;i++) std::cin>>arr[i];
}
void mass::reverseArr(){
        for (int i=0;i<N/2;i++) std::swap(arr[i],arr[N-i-1]);
}
void mass::getArr(){
        for (int i=0;i<N;i++) printf("%5d",arr[i]);
}
```

Файл mass.h

```
#ifndef MASS_H
#define MASS_H

class mass{
private:
```

```
        int N;
        int *arr;
public:
        mass(int n);
        void setArr();
        void reverseArr();
        void getArr();
};

#endif
```

Тестирование

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
7 2 3 4 5 6 7 8	N = 7 2 3 4 5 6 7 8 8 7 6 5 4 3 2	N = 7 2 3 4 5 6 7 8 8 7 6 5 4 3 2
10 23 46 321 532 1 63 90 6723 108 0	N = 10 23 46 321 532 1 63 90 6723 108 0 0 108 6723 90 63 1 532 321 46 23	N = 10 23 46 321 532 1 63 90 6723 108 0 0 108 6723 90 63 1 532 321 46 23
4 1 2 3 4	N = 4 1 2 3 4 4 3 2 1	N = 4 1 2 3 4 4 3 2 1