

Здесь будет титульник, листай ниже

# СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	6
1.2 Описание выходных данных.....	7
2 МЕТОД РЕШЕНИЯ.....	8
3 ОПИСАНИЕ АЛГОРИТМОВ.....	10
3.1 Алгоритм конструктора класса MyClass.....	10
3.2 Алгоритм метода printArray класса MyClass.....	10
3.3 Алгоритм метода change класса MyClass.....	11
3.4 Алгоритм метода freeMemory класса MyClass.....	11
3.5 Алгоритм деструктора класса MyClass.....	12
3.6 Алгоритм конструктора класса MyClass.....	12
3.7 Алгоритм функции main.....	13
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	14
5 КОД ПРОГРАММЫ.....	19
5.1 Файл main.cpp.....	19
5.2 Файл MyClass.cpp.....	19
5.3 Файл MyClass.h.....	20
6 ТЕСТИРОВАНИЕ.....	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	23

# 1 ПОСТАНОВКА ЗАДАЧИ

Сконструировать систему, которая демонстрирует особенности при передаче объекта в функцию в качестве параметра. Показывает, какого характера ошибки могут возникнуть при выполнении такой передачи и как их обойти.

Спроектировать объект (разработать описание класса), который имеет свойства:

- - переменную целого типа в закрытом доступе, содержит количество элементов массива;
- - указатель на массив целого типа в открытом доступе.

Функционал объекта:

- - параметризованный конструктор, с одним целочисленным параметром. С новой строки выводит сообщение, что работает параметризованный конструктор. Конструктор создает массив целочисленного типа размерности согласно значению параметра. Каждому элементу присваивает значение равное квадрату соответствующего индекса;
- - конструктор по умолчанию. Выдает сообщение, что работает конструктор по умолчанию;
- - метод, в открытом доступе, который выводит значения элементов массива с новой строки, разделенным тремя пробелами;
- - метод, в открытом доступе, который меняет знак каждого второго элемента массива;
- - метод, в открытом доступе, который освобождает память, выделенную для массива;
- - деструктор, выдает с новой строки сообщение, что работает деструктор.

Разработать функцию, которой в качестве параметра передается объект спроектированного класса. В описании функции реализовать алгоритм:

1. Для объекта, переданного в качестве параметра вызов метода, который меняет знак значений части элементов массива.

2. Вызов метода вывода содержимого массива.

Исходный алгоритм сборки и отработки системы:

1. Вводит целочисленное, положительное значение, размещает в целочисленной переменной.

2. Объявляет объект с использованием параметризованного конструктора, в качестве аргумента передает целочисленную переменную, которая содержит введенное значение.

3. Вызывает метод вывода значений элементов объявленного объекта.

4. Вызывает функцию, которой в качестве аргумента передает объявленный объект.

5. Вызывает метод вывода значений элементов объявленного объекта.

6. Вызов метода для освобождения памяти для массива.

7. Завершение работы системы.

Доработайте конструкцию (не меняя описание класса), алгоритм сборки и отработки системы так, чтобы выводимые значения во второй и пятой строке совпали.

## **1.1 Описание входных данных**

Первая строка:

«целое число»

Пример ввода

5

## 1.2 Описание выходных данных

Конструктор по умолчанию выводит сообщение:

```
Default constructor
```

Параметризованный конструктор выводит сообщение:

```
Parameterized constructor
```

Деструктор выводит сообщение:

```
Destructor
```

Метод вывода значений элементов массива выводит последовательность чисел:

```
«целое число»  «целое число»  «целое число» . . .
```

Пример вывода

До доработки системы

```
Parameterized constructor
0  1  4  9 16
0 -1  4 -9 16
Destructor
0 -1  4 -9 16
Destructor
```

После доработки системы

```
Parameterized constructor
0  1  4  9 16
0 -1  4 -9 16
Destructor
0  1  4  9 16
Destructor
```

## 2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект `myobj` класса `MyClass` предназначен для демонстрации особенности при передаче объекта в функцию в качестве параметра;
- функция `main` для главная функция программы;
- функция `processArray` для повторной смены знаков каждой второй переменной в массиве и вывод этого массива;
- библиотека `iostream`;
- операторы сравнения `<`, `>`;
- арифметические операторы сложения и вычитания `+`, `-`;
- оператор присвоения `=`;
- цикл со счетчиком `for`;
- объекты стандартного потока ввода/вывода данных `cin`, `cout`;
- оператор переноса строки `endl`;
- целочисленные переменные;
- массивы;
- указатель `*`;
- операторы динамического управления памяти `new` `delete`.

Класс `MyClass`:

- свойства/поля:
  - о поле переменная целого типа:
    - наименование — `size`;
    - тип — `int`;
    - модификатор доступа — `private`;
  - о поле указатель на массив целого типа:
    - наименование — `array`;

- тип — `int*`;
  - модификатор доступа — `public`;
- функционал:
  - o метод `MyClass` — параметризированный конструктор;
  - o метод `MyClass` — конструктор по умолчанию;
  - o метод `printArray` — вывод массива в консоль;
  - o метод `change` — изменение знака каждого второго элемента массива;
  - o метод `freeMemory` — освобождение памяти, которую занимал массив;
  - o метод `~MyClass` — деструктор по умолчанию.

## 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 MyClass

Функционал: параметризованный конструктор.

Параметры: int n, целое число - размер массива.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса MyClass

№	Предикат	Действия	№ перехода
1		вывод "Parameterized constructor"	2
2		присвоение переменной size значения n	3
3		инициализация динамического массива array с размером size	4
4		инициализация переменной целого типа i со значением 0	5
5	i < size?	присвоить значение i^2 элементу динамического массива по индексу i	6
			Ø
6		инкремент i	5

### 3.2 Алгоритм метода printArray класса MyClass

Функционал: вывод массива в консоль.

Параметры: нет.

Возвращаемое значение: ничего.



Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода *printArray* класса *MyClass*

№	Предикат	Действия	№ перехода
1		инициализация переменной целого типа <i>i</i> со значением 0	2
2	<i>i</i> < size?	вывод значения элемента динамического массива по индексу <i>i</i> и " "	3
		перенос строки	∅
3		инкремент <i>i</i>	2

### 3.3 Алгоритм метода *change* класса *MyClass*

Функционал: изменение знака каждого второго элемента массива.

Параметры: нет.

Возвращаемое значение: ничего.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода *change* класса *MyClass*

№	Предикат	Действия	№ перехода
1		инициализация переменной целого типа <i>i</i> со значением 1	2
2	<i>i</i> < size?	присвоение элементу динамического массива <i>array</i> по индексу <i>i</i> значения 0 - значение динамического массива по индексу <i>i</i>	3
			∅
3		присвоение переменной <i>i</i> значения <i>i</i> +2	2

### 3.4 Алгоритм метода *freeMemory* класса *MyClass*

Функционал: освобождение памяти, которую занимал массив.

Параметры: нет.

Возвращаемое значение: ничего.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *freeMemory* класса *MyClass*

№	Предикат	Действия	№ перехода
1		удаление памяти, выделенной под массив array	2
2		присвоение указателю переменной пустого значения	3
3		присвоение переменной size значения 0	∅

### 3.5 Алгоритм деструктора класса *MyClass*

Функционал: деструктор по умолчанию.

Параметры: нет.

Алгоритм деструктора представлен в таблице 5.

Таблица 5 – Алгоритм деструктора класса *MyClass*

№	Предикат	Действия	№ перехода
1		Вывод "Destructor"	∅

### 3.6 Алгоритм конструктора класса *MyClass*

Функционал: конструктор по умолчанию.

Параметры: нет.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса *MyClass*

№	Предикат	Действия	№ перехода
1		вывод "Constructor"	2
2		присвоение переменной size значения 0	3

№	Предикат	Действия	№ перехода
3		присвоение указателю переменной пустого значения	Ø

### 3.7 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое число - индикатор работоспособности программы.

Алгоритм функции представлен в таблице 7.

Таблица 7 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление переменной целого типа n	2
2		присвоение значения переменной n	3
3		создание объекта myobj с передачей параметра n для параметризованного конструктора	4
4		вызов метода printArray	5
5		вызов функции processArray с параметром myobj	6
6		вызов метода change	7
7		вызов метода printArrat	8
8		вызов метода freeMemory	Ø

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-5.

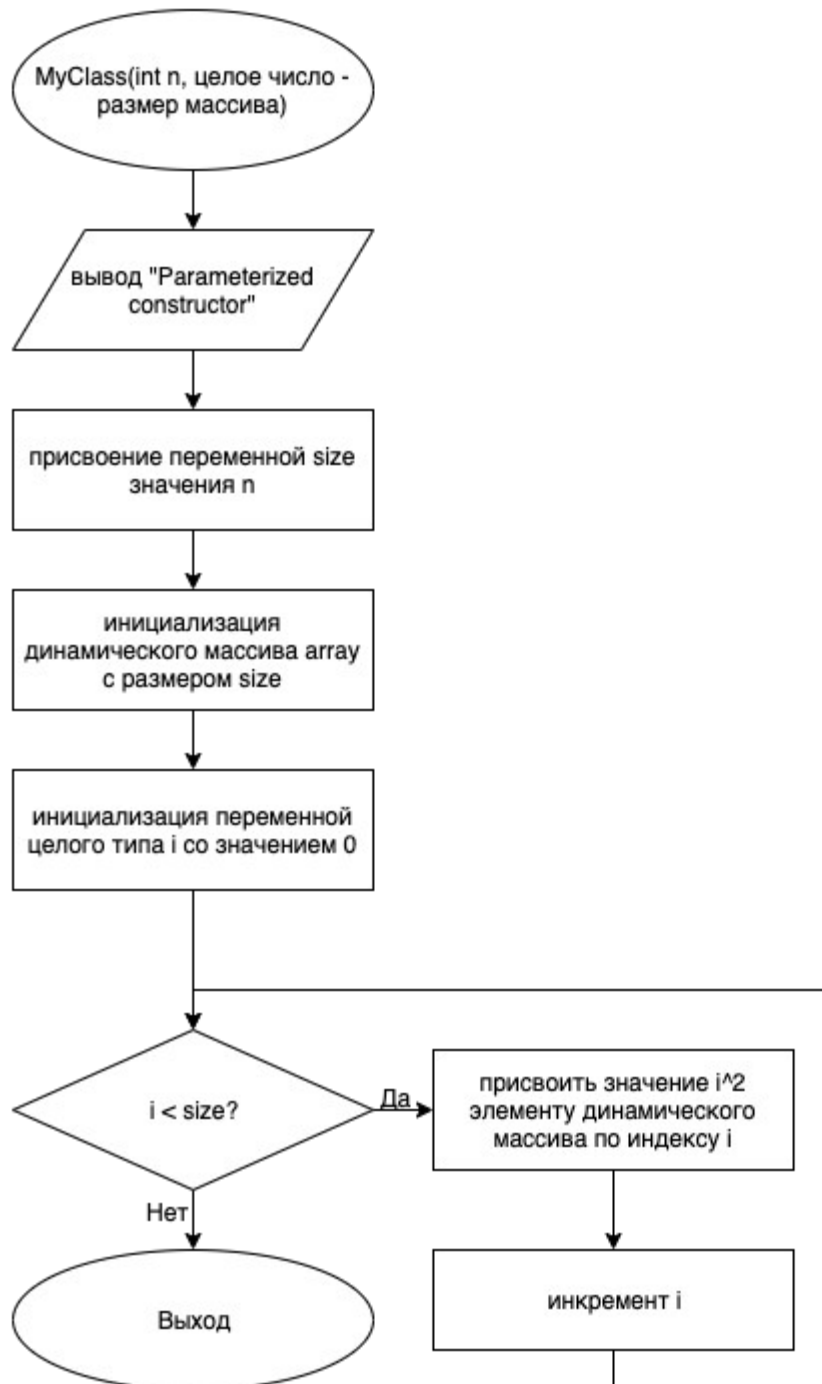


Рисунок 1 – Блок-схема алгоритма

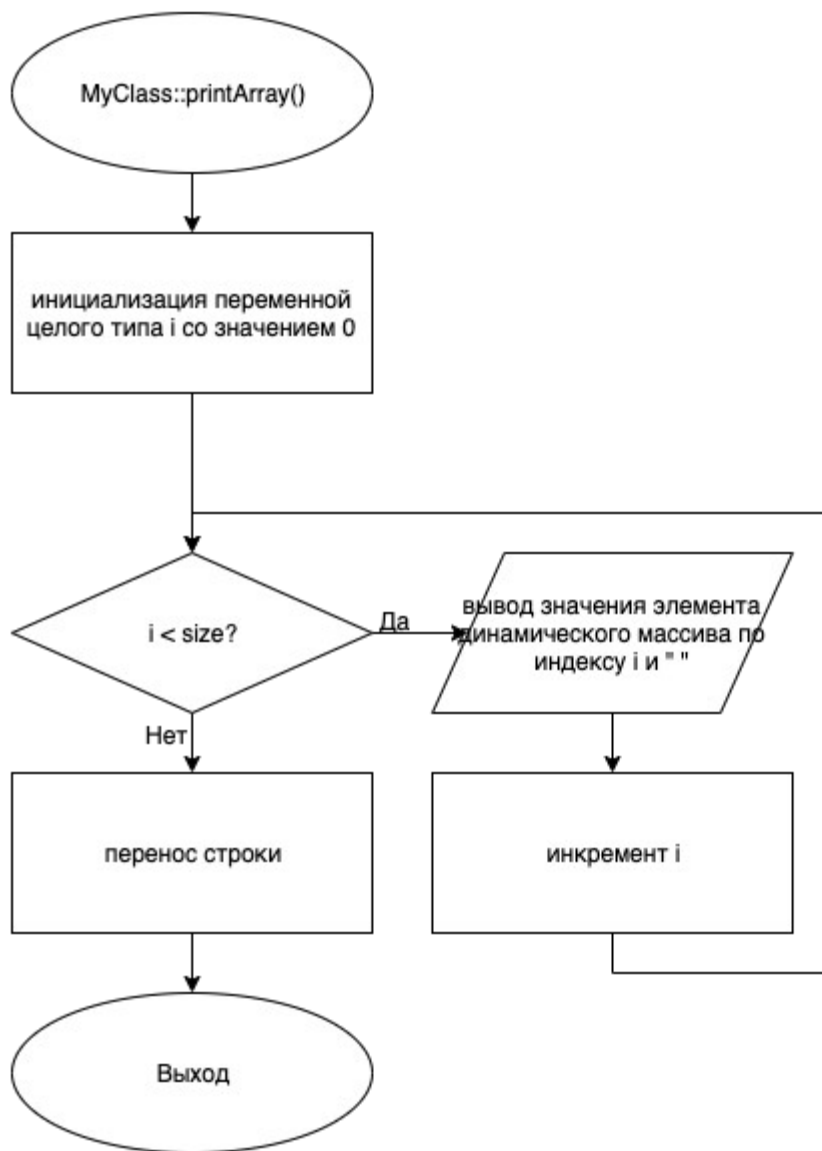


Рисунок 2 – Блок-схема алгоритма

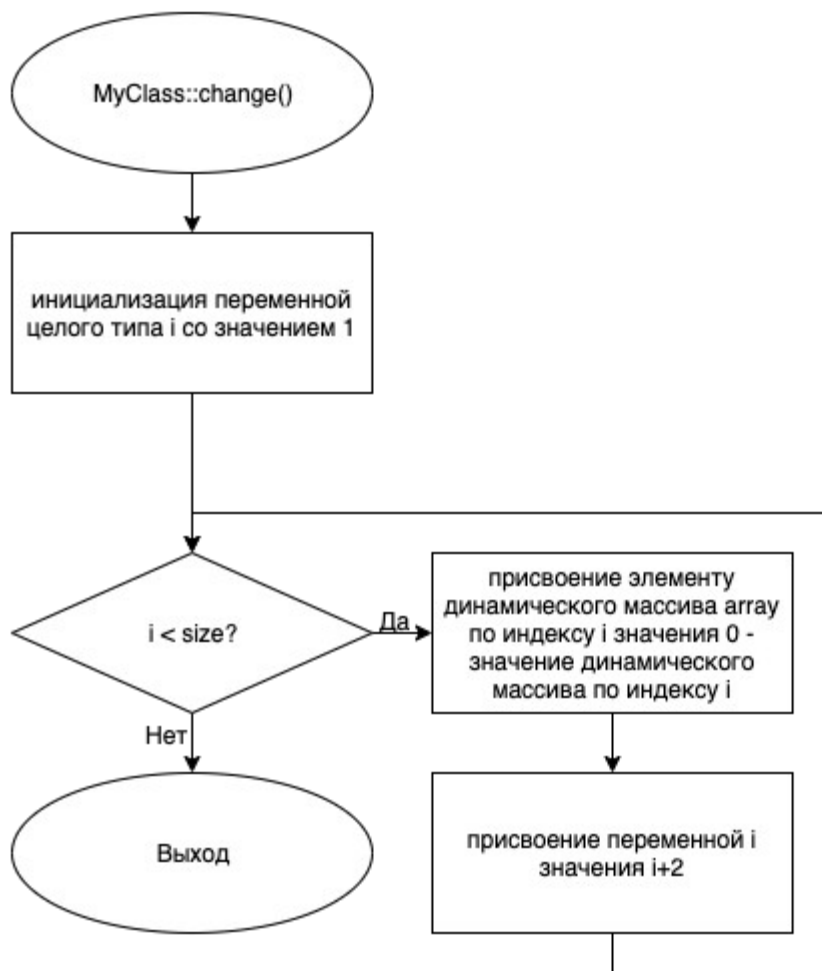


Рисунок 3 – Блок-схема алгоритма

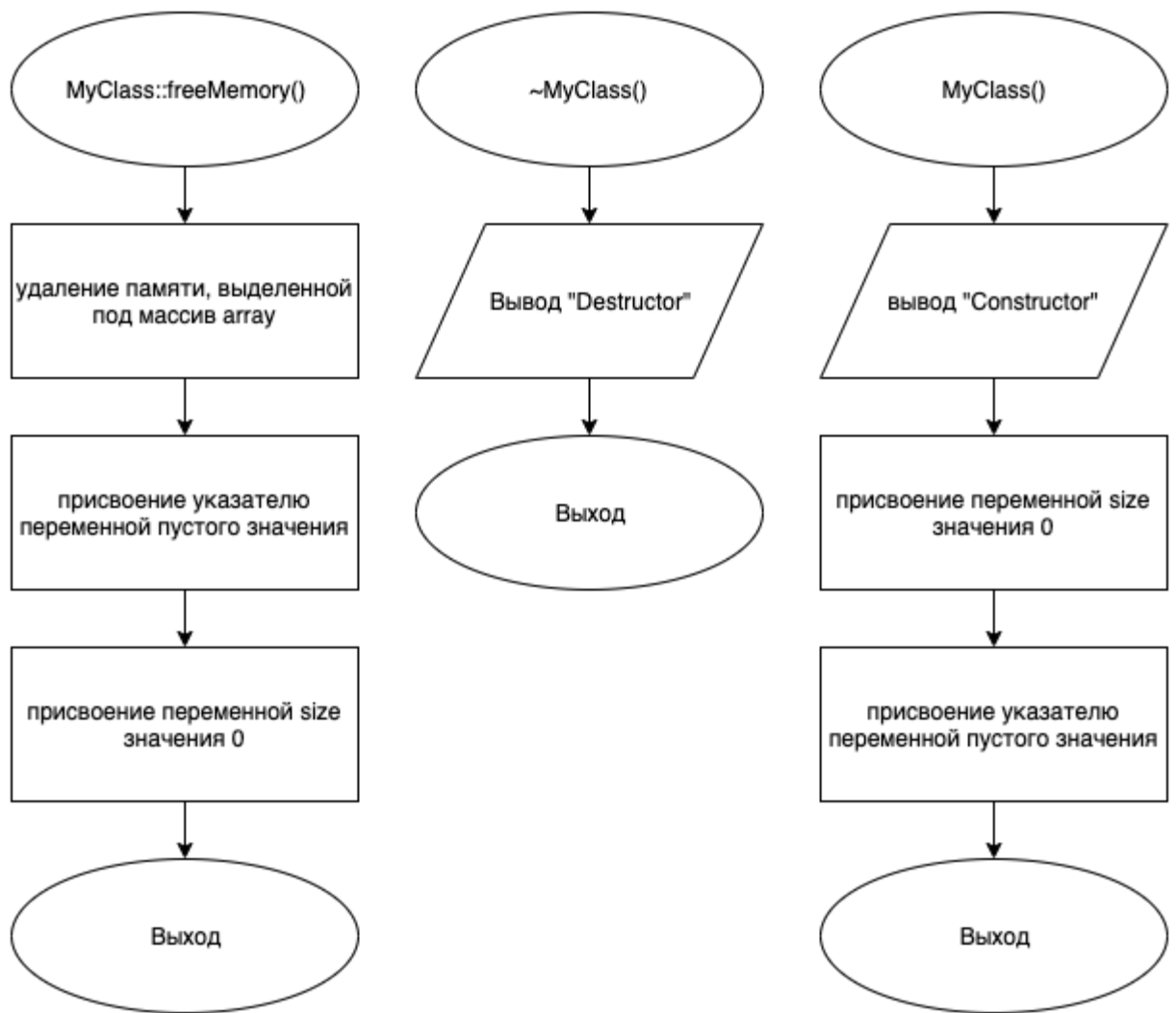


Рисунок 4 – Блок-схема алгоритма

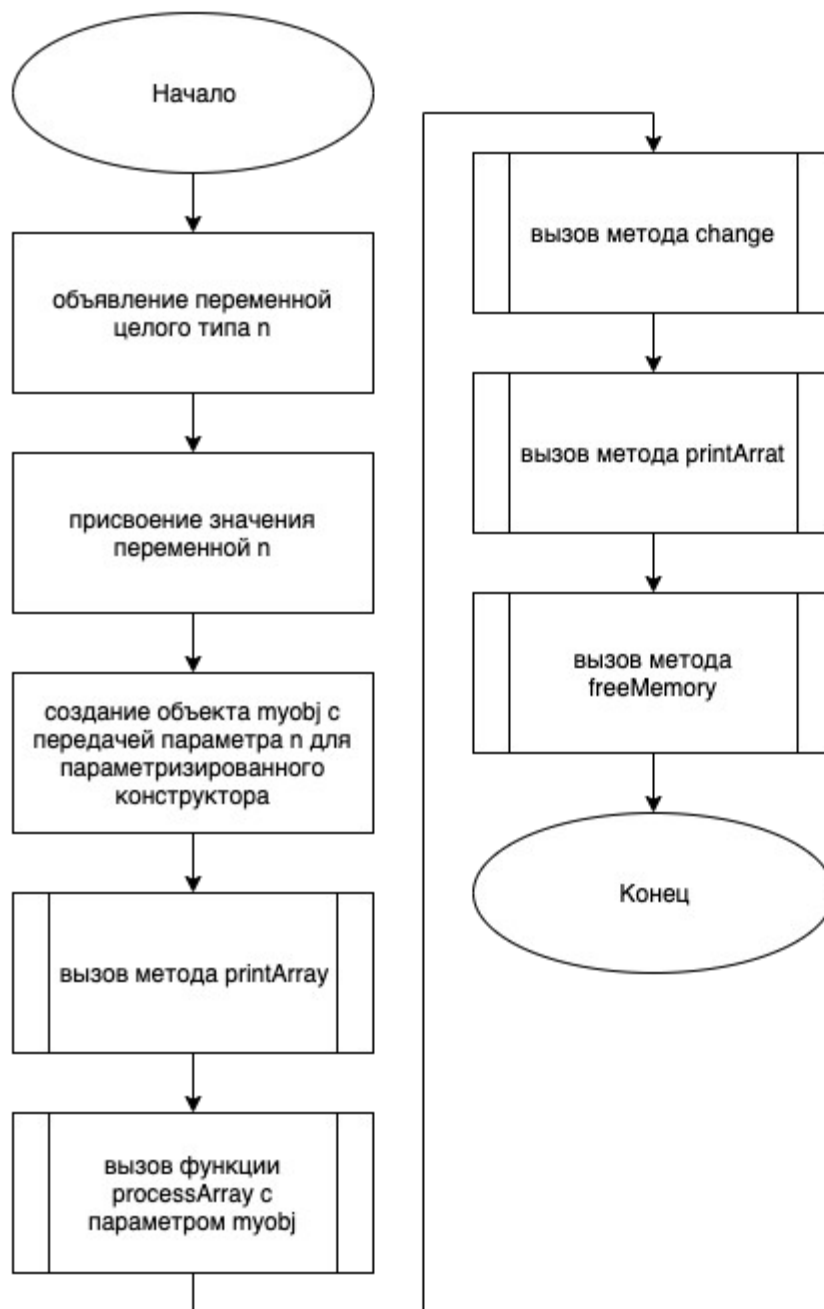


Рисунок 5 – Блок-схема алгоритма



## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл main.cpp

*Листинг 1 – main.cpp*

```
#include <stdlib.h>
#include <stdio.h>
#include "MyClass.h"
#include <iostream>
using namespace std;

void processArray(MyClass obj){
    obj.change();
    obj.printArray();
}

int main(){
    int n;
    cin >> n;

    MyClass myobj(n);

    myobj.printArray();
    processArray(myobj);
    myobj.change();
    myobj.printArray();
    myobj.freeMemory();

    return(0);
}
```

### 5.2 Файл MyClass.cpp

*Листинг 2 – MyClass.cpp*

```
#include "MyClass.h"
#include <iostream>
using namespace std;
```

```

MyClass::MyClass(int n){
    cout << "Parameterized constructor" << endl;
    size = n;
    array = new int[size];
    for (int i = 0; i < size; ++i){
        array[i] = i*i;
    }
}

MyClass::MyClass(){
    cout << "Constructor" << endl;
    size = 0;
    array = nullptr;
}

MyClass::~MyClass(){
    cout << "Destructor" << endl;
}

void MyClass::printArray(){
    for (int i = 0; i < size; ++i){
        if (i == size-1){
            cout << array[i];
        } else {
            cout << array[i] << " ";
        }
    }
    cout << endl;
}

void MyClass::change(){
    for (int i = 1; i < size; i+=2){
        array[i] = -array[i];
    }
}

void MyClass::freeMemory(){
    delete[] array;
    array = nullptr;
    size = 0;
}

```

### 5.3 Файл MyClass.h

*Листинг 3 – MyClass.h*

```

#ifndef __MYCLASS__H
#define __MYCLASS__H

class MyClass{

```

```
private:
    int size;

public:
    int* array;
    MyClass(int n);
    MyClass();
    ~MyClass();

    void printArray();
    void change();
    void freeMemory();
};

#endif
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 8.

Таблица 8 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5	Parameterized constructor 0 1 4 9 16 0 -1 4 -9 16 Destructor 0 1 4 9 16 Destructor	Parameterized constructor 0 1 4 9 16 0 -1 4 -9 16 Destructor 0 1 4 9 16 Destructor

## 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](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](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).