

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 MeineKlasse.....	10
3.2 Алгоритм деструктора класса MeineKlasse.....	10
3.3 Алгоритм метода ArrayInput класса MeineKlasse.....	11
3.4 Алгоритм метода PairSumm класса MeineKlasse.....	11
3.5 Алгоритм метода PairMult класса MeineKlasse.....	11
3.6 Алгоритм метода ArraySumm класса MeineKlasse.....	12
3.7 Алгоритм конструктора класса MeineKlasse.....	13
3.8 Алгоритм конструктора класса MeineKlasse.....	13
3.9 Алгоритм функции main.....	14
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	16
5 КОД ПРОГРАММЫ.....	23
5.1 Файл main.cpp.....	23
5.2 Файл MeineKlasse.cpp.....	23
5.3 Файл MeineKlasse.h.....	24
6 ТЕСТИРОВАНИЕ.....	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	27

1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В закрытом доступе имеется массив целого типа и поле его длины. Количество элементов массива четное и больше двух. Объект имеет функциональность:

- Конструктор по умолчанию, в начале работы выдает сообщение;
- Параметризованный конструктор, передается целочисленный параметр. По значению параметра определяется размерность целочисленного массива из закрытой области. Массив создается. В начале работы выдает сообщение;
- Метод деструктор, который выдает сообщение что он отработал;
- Метод ввода данных для созданного массива;
- Метод 1, который суммирует значения очередной пары элементов и сумму присваивает первому элементу пары. Далее суммирует элементы полученного массива и возвращает это значение. Например, пусть массив состоит из элементов {1,2,3,4}. В результате суммирования пар получим массив {3,2,7,4};
- Метод 2, который умножает значения очередной пары элементов и результат присваивает первому элементу пары. Далее суммирует элементы полученного массива и возвращает это значение. Например, пусть массив состоит из элементов {1,2,3,4}. В результате умножения пар получим массив {2,2,12,4};
- Метод который, суммирует значения элементов массива и возвращает это значение.

Разработать функцию, которая в качестве параметра получает объект по значению. Функция вызывается метод 2, далее выводит сумму элементов массива с новой строки.

В основной функции реализовать алгоритм:

1. Ввод размерности массива. Размер должен иметь значение больше 2 и быть четным.
2. Если размерность массива некорректная, вывод сообщения и завершить работу алгоритма.
3. Вывод значения размерности массива.
4. Создание объекта с аргументом размерности массива.
5. Вызов метода для ввода значений элементов массива.
6. Вызов функции передача в качестве аргумента объекта.
7. Вызов метода 1 от имени объекта.
8. Вывод суммы элементов массива объекта с новой строки.

Разработать конструктор копии объекта для корректного выполнения вычислений. В начале работы конструктор копии выдает сообщение с новой строки.

1.1 Описание входных данных

Первая строка:

«Целое число»

Вторая строка:

«Целое число» «Целое число» . . .

Пример:

8
1 2 3 4 5 6 7 8

1.2 Описание выходных данных

Если введенная размерность массива допустима, то в первой строке выводится это значение:

«Целое число»

Если введенная размерность массива не больше двух или нечетная, то в первой строке выводится некорректное значение и вопросительный знак:

«Целое число»?

Конструктор по умолчанию в начале работы с новой строки выдает сообщение:

Default constructor

Параметризированный конструктор в начале работы с новой строки выдает сообщение:

Constructor set

Конструктор копирования в начале работы с новой строки выдает сообщение:

Copy constructor

Деструктор в начале работы с новой строки выдает сообщение:

Destructor

Пример вывода:

```
8
Constructor set
Copy constructor
120
Destructor
56
Destructor
```

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект obj класса MeineKlasse предназначен для создания полей объекта класса;
- функция MultiplyFunction для вызова метода копии объекта, дальнейшего вызова метода копии объекта, дальнейшего вызова метода перемножения парных элементов;
- cin - оператор стандартного потока ввода;
- cout - оператор стандартного потока вывода.

Класс MeineKlasse:

- свойства/поля:
 - поле размер целочисленного массива:
 - наименование — size;
 - тип — int;
 - модификатор доступа — private;
 - поле указатель на целочисленный массив:
 - наименование — Array;
 - тип — int*;
 - модификатор доступа — private;
- функционал:
 - метод MeineKlasse — стандартный конструктор;
 - метод MeineKlasse — параметризованный конструктор;
 - метод MeineKlasse — конструктор копии;
 - метод ~MeineKlasse — деструктор;
 - метод ArrayInput — ввод значений массива;
 - метод PairSumm — нахождение суммы пар значений массива;

- o метод PairMult — нахождение разности пар значений массива;
- o метод ArraySumm — сумма всех 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 MeineKlasse

Функционал: создание объекта класса MeineKlasse, вывод сообщения об отработке.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса MeineKlasse

№	Предикат	Действия	№ перехода
1		создание указателя Array на массив с целого типа со значением 0	2
2		вывод сообщения "Default Constructor"	3
3		переход на новую строку	Ø

3.2 Алгоритм деструктора класса MeineKlasse

Функционал: деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса MeineKlasse

№	Предикат	Действия	№ перехода
1		удаление указателя Array	2
2		вывод сообщения "Destructor"	Ø

3.3 Алгоритм метода `ArrayInput` класса `MeineKlasse`

Функционал: ввод значений массива, затем подсчёт суммы элементов.

Параметры: нет.

Возвращаемое значение: `int` - сумма значений.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода `ArrayInput` класса `MeineKlasse`

№	Предикат	Действия	№ перехода
1		инициализация целочисленной переменной $i = 0$	2
2	$i < \text{size}$	ввод значения i массива по указателю <code>Array</code>	3
			\emptyset
3		увеличение значения переменной i на 1	2

3.4 Алгоритм метода `PairSumm` класса `MeineKlasse`

Функционал: нахождение суммы пар значений массива.

Параметры: нет.

Возвращаемое значение: `int` - сумма элементов массива.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода `PairSumm` класса `MeineKlasse`

№	Предикат	Действия	№ перехода
1		инициализация целочисленной переменной $i = 1$	2
2	$i < \text{size}$	присвоение $i-1$ элементу по указателю <code>Array</code> значения равному сумме парных элементов i и $i-1$	3
		возврат значения метода <code>ArraySumm</code>	\emptyset
3		увеличение значения переменной i на 2	2

3.5 Алгоритм метода PairMult класса MeineKlasse

Функционал: нахождение разности пар значений массива.

Параметры: нет.

Возвращаемое значение: int - сумма элементов массива.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода PairMult класса MeineKlasse

№	Предикат	Действия	№ перехода
1		инициализация целочисленной переменной $i = 1$	2
2	$i < \text{size}$	присвоение $i-1$ элементу по указателю Array значения равному произведению парных элементов i и $i-1$	3
		возврат значения метода ArraySumm	Ø
3		увеличение значения переменной i на 2	2

3.6 Алгоритм метода ArraySumm класса MeineKlasse

Функционал: подсчёт суммы всех элементов массива.

Параметры: нет.

Возвращаемое значение: int - сумма элементов массива.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода ArraySumm класса MeineKlasse

№	Предикат	Действия	№ перехода
1		инициализация целочисленной переменной $\text{sum} = 0$	2
2		инициализация целочисленной переменной $i = 0$	3
3	$i < \text{size}$	к значению переменной sum прибавляется значение переменной i массива по указателю	4

№	Предикат	Действия	№ перехода
		Array	
		возврат значения переменной sum	∅
4		увеличение значения i на 1	3

3.7 Алгоритм конструктора класса MeineKlasse

Функционал: создания указателя Array на целочисленный массив.

Параметры: int, size, размер массива.

Алгоритм конструктора представлен в таблице 7.

Таблица 7 – Алгоритм конструктора класса MeineKlasse

№	Предикат	Действия	№ перехода
1		вывод "Set Constructor"	2
2		переход на новую строку	3
3		создания указателя Array на целочисленный массив размером comes_size	4
4		присовение полю size, значения comes_size	∅

3.8 Алгоритм конструктора класса MeineKlasse

Функционал: копирование всех полей объекта obj.

Параметры: const MeineKlasse& obj.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса MeineKlasse

№	Предикат	Действия	№ перехода
1		создание указателя Array на целочисленный массив размером, равный полю size объекта obj	2
2		присвоение полю size, равному значению поля size	3

№	Предикат	Действия	№ перехода
		объекта obj	
3		инициализация целочисленной переменной $i = 0$	4
4	$i < \text{size}$	іму элементу по указателю Array присваивается значение, равное іму элементу массива по указателю Array объекта obj	5
			6
5		увеличение значения i на 1	6
6		вывод "Copy constructor" и переход на новую строку	7
7		возврат 0	\emptyset

3.9 Алгоритм функции main

Функционал: выполнение поставленной задачи.

Параметры: нет.

Возвращаемое значение: `int` - индикатор успешности выполнения программы.

Алгоритм функции представлен в таблице 9.

Таблица 9 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление целочисленной переменной <code>size</code>	2
2		ввод <code>size</code>	3
3	$\text{size} > 2$ и size чётное	вывод <code>size</code>	4
		вывод "size?"	9
4		создание объекта <code>obj</code> класса <code>MeineKlasse</code> с аргументом <code>size</code>	5
5		вызов метода <code>ArrayInput</code> для объекта <code>obj</code>	6
6		вызов функции <code>MultiplyFunction</code> с аргументом <code>obj</code>	7

№	Предикат	Действия	№ перехода
7		вывод результата вызова метода PairSumm объекта obj	8
8		вызов метода ArraySumm для объекта obj	9
9		возврат 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-7.

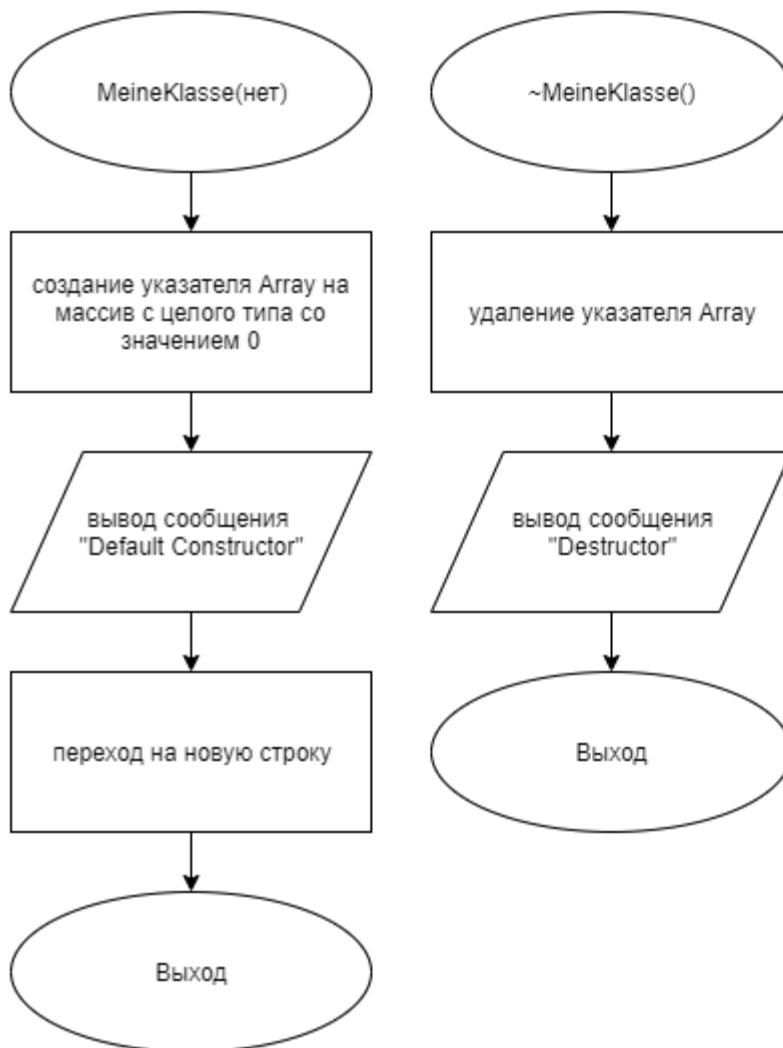


Рисунок 1 – Блок-схема алгоритма

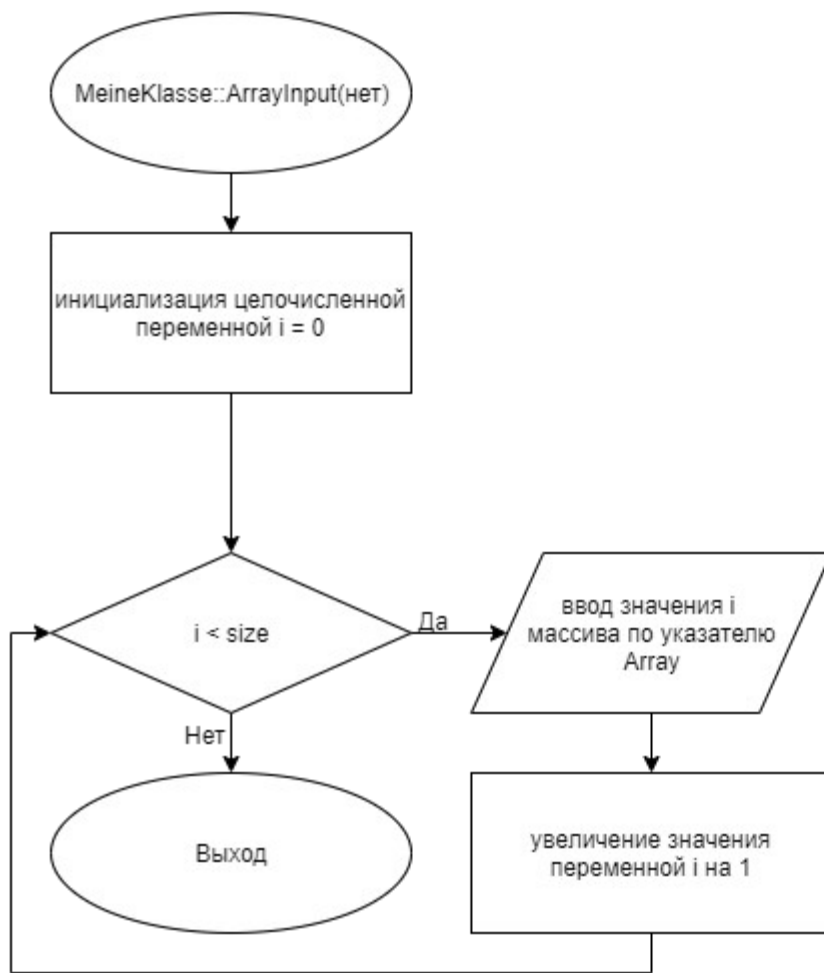


Рисунок 2 – Блок-схема алгоритма

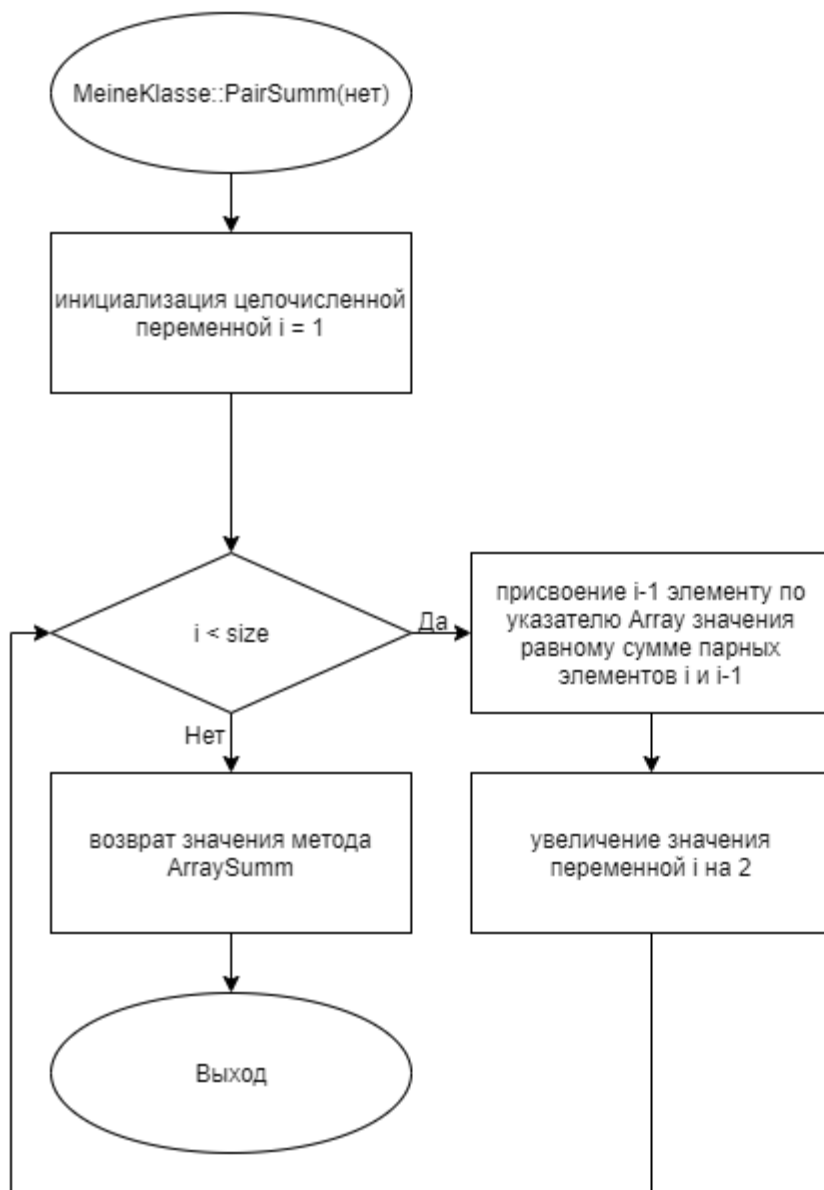


Рисунок 3 – Блок-схема алгоритма

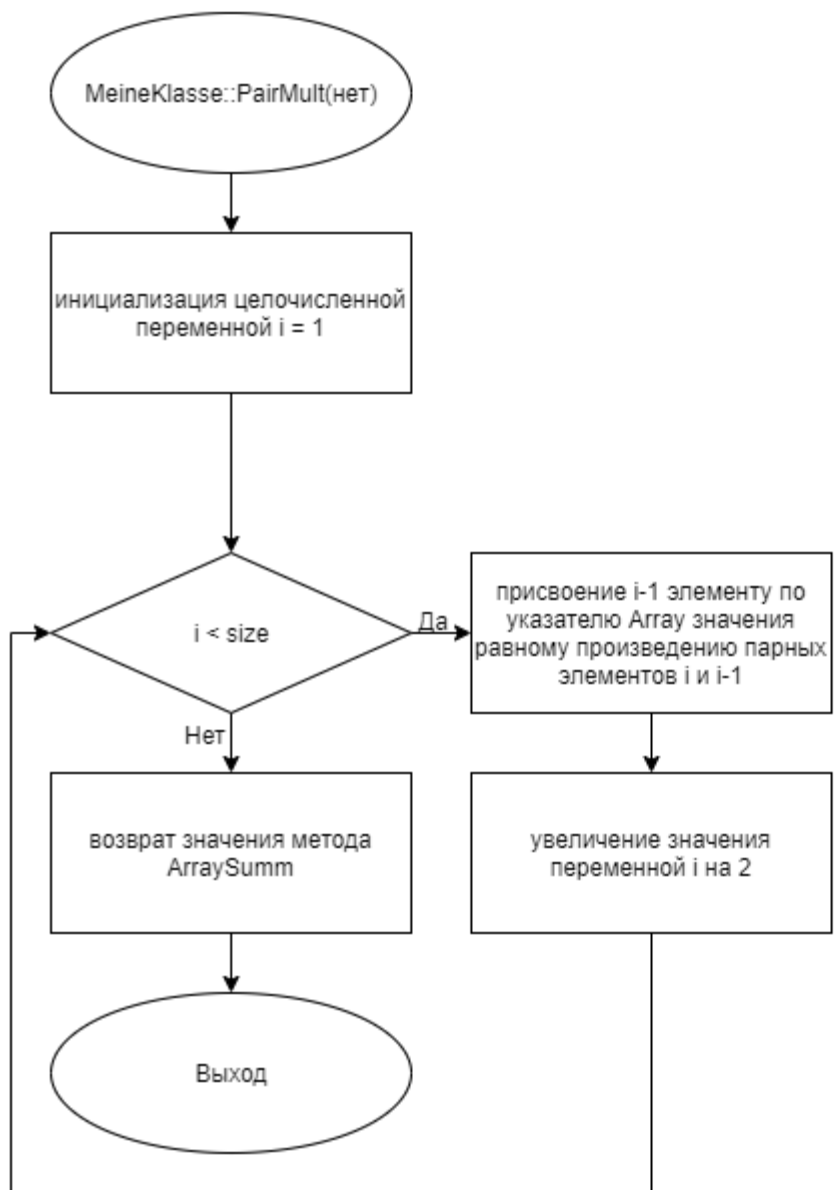


Рисунок 4 – Блок-схема алгоритма

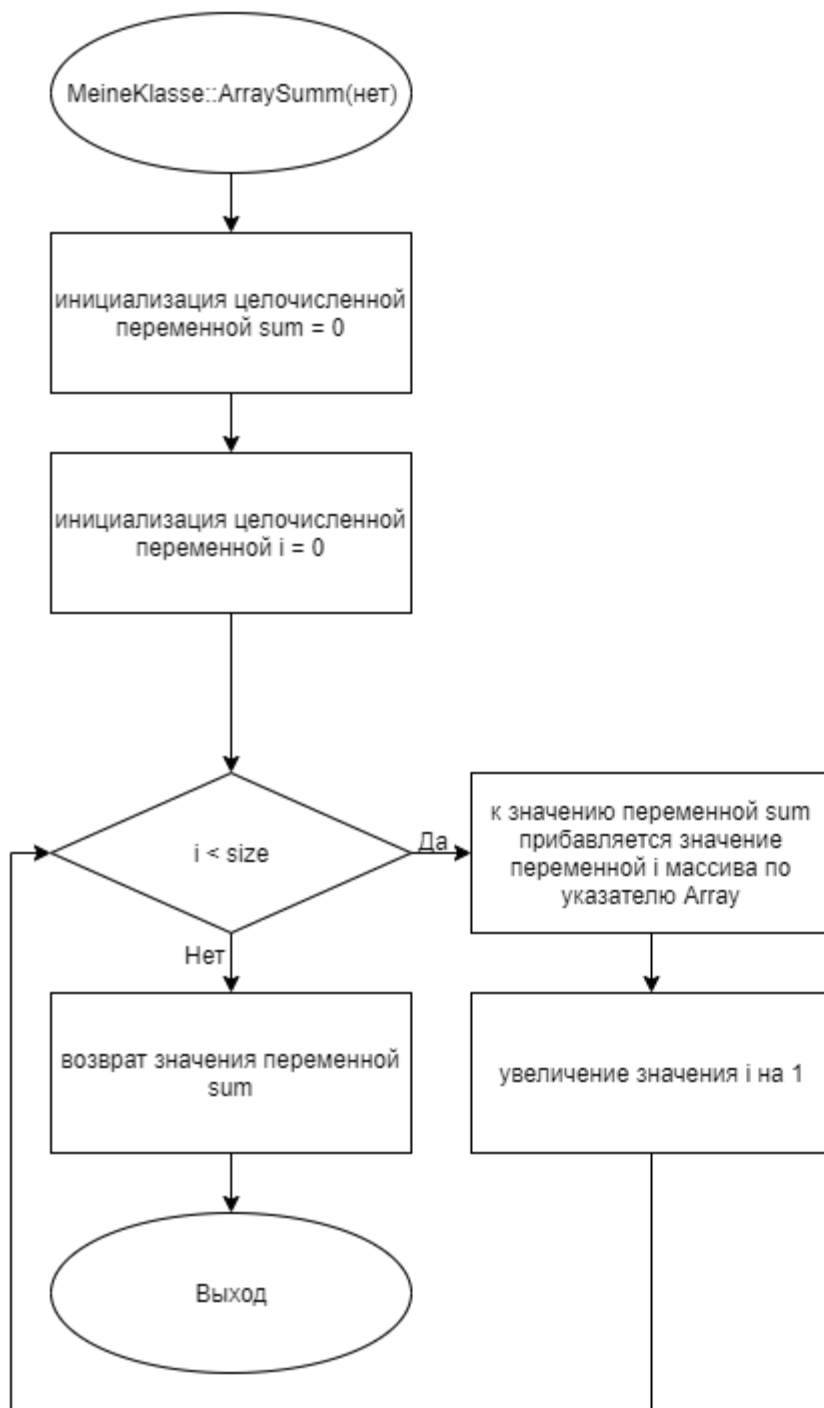


Рисунок 5 – Блок-схема алгоритма

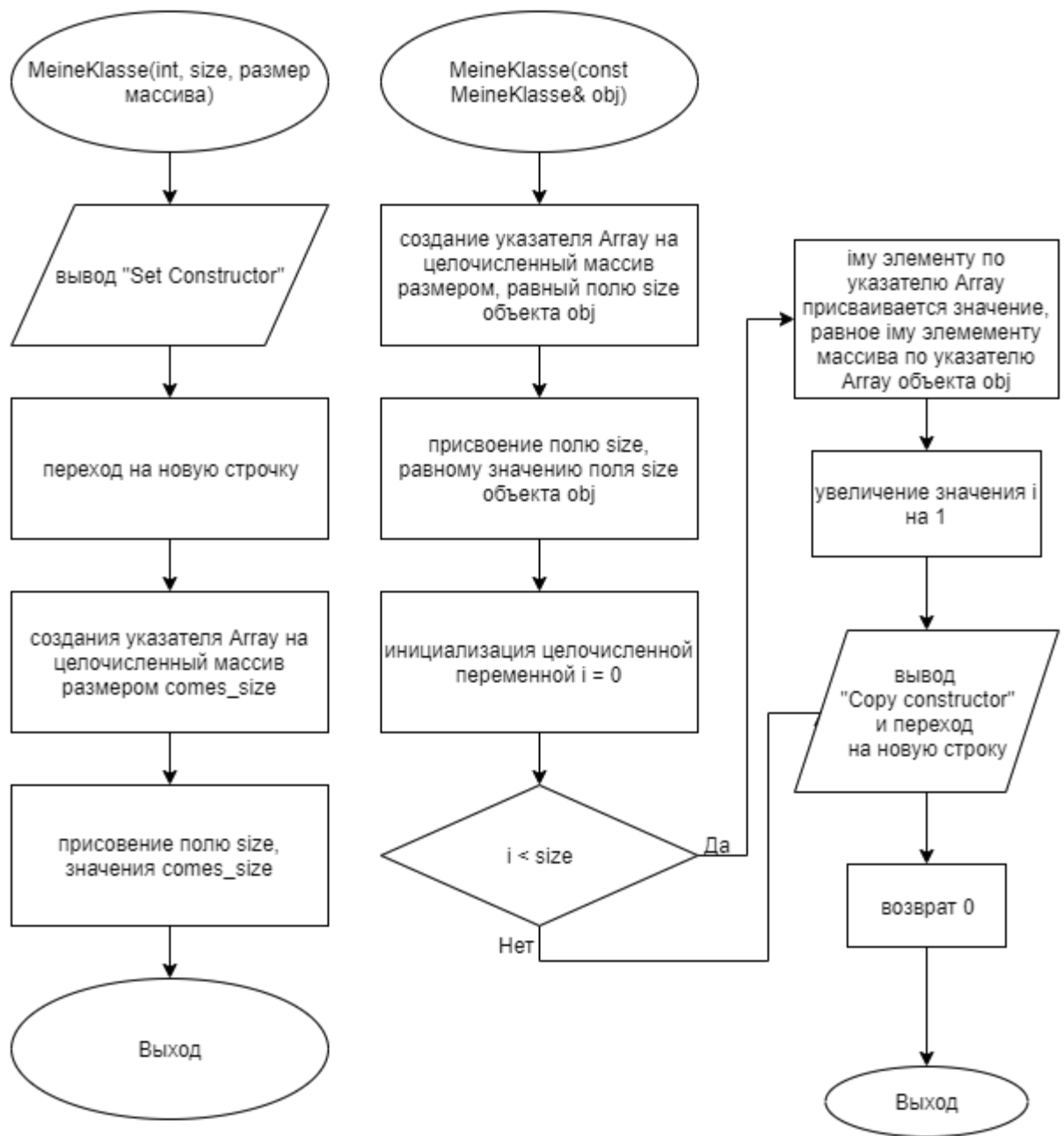


Рисунок 6 – Блок-схема алгоритма

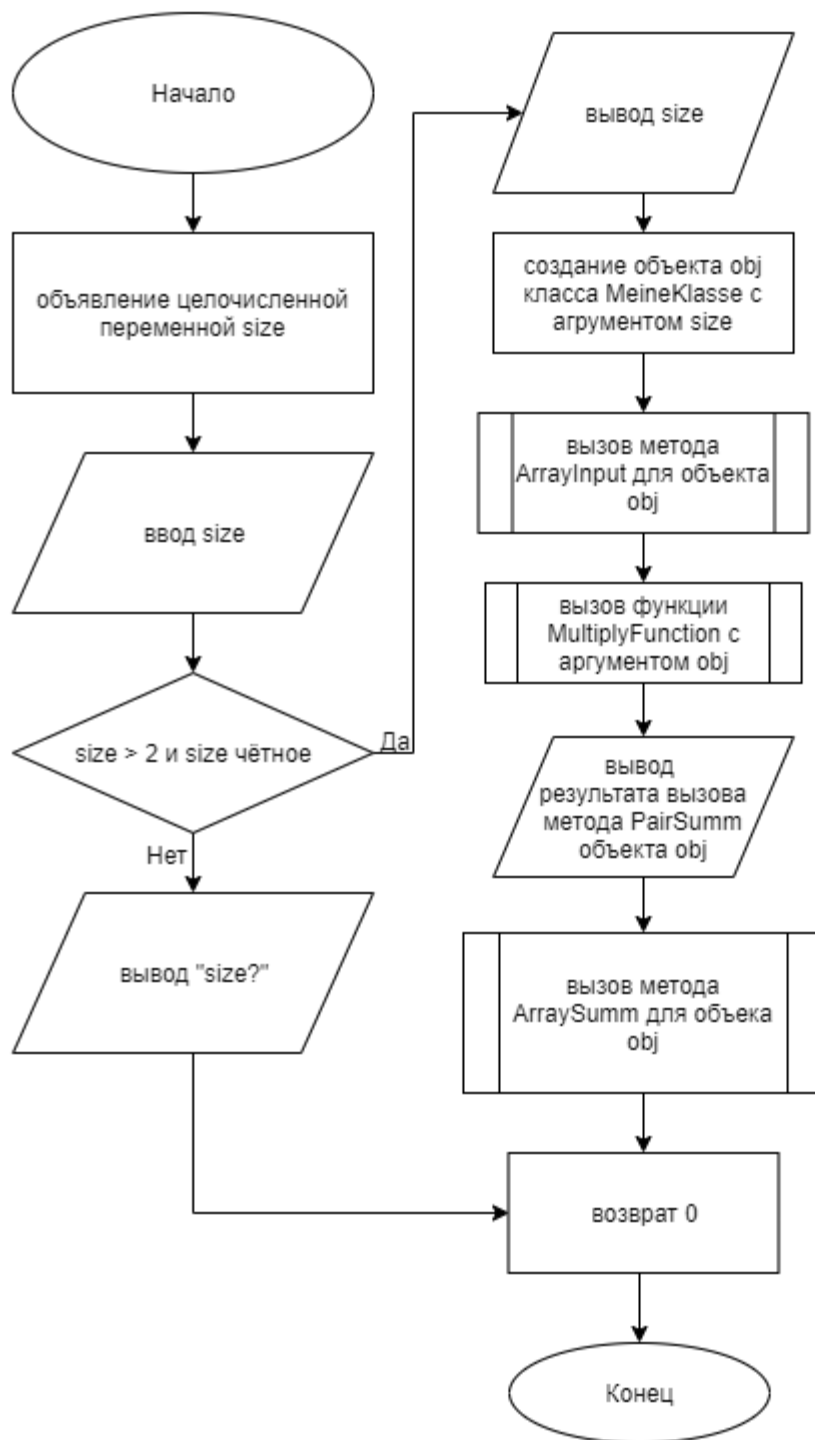


Рисунок 7 – Блок-схема алгоритма

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 <iostream>
#include "MeineKlasse.h"

void MultiplyFunction(MeineKlasse obj){
    std::cout << obj.PairMult() << "\n";
}

int main()
{
    int size;
    std::cin >> size;
    if ((size > 2) && (size % 2 == 0)){
        std::cout << size << "\n";
        MeineKlasse obj(size);
        obj.ArrayInput();
        MultiplyFunction(obj);
        std::cout << obj.PairSumm() << "\n";
        obj.ArraySumm();
    }
    else{
        std::cout << size << "?\n";
    }
    return(0);
}
```

5.2 Файл MeineKlasse.cpp

Листинг 2 – MeineKlasse.cpp

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

```

MeineKlasse::MeineKlasse(){
    Array = nullptr;
    std::cout << "Default constructor\n";
}
MeineKlasse::MeineKlasse(const MeineKlasse & obj){
    Array = new int[obj.size];
    size = obj.size;
    for (int i = 0; i < size; ++i){
        Array[i] = obj.Array[i];
    }
    std::cout << "Copy constructor\n";
}
MeineKlasse::MeineKlasse(int comes_size){
    std::cout << "Constructor set\n";
    Array = new int[comes_size];
    size = comes_size;
}
MeineKlasse::~MeineKlasse(){
    delete[] Array;
    std::cout << "Destructor\n";
}
void MeineKlasse::ArrayInput(){
    for (int i = 0; i < size; i++){
        std::cin >> Array[i];
    }
}
int MeineKlasse::PairSumm(){
    for (int i = 1; i < size; i+=2){
        Array[i-1] = Array[i] + Array[i-1];
    }
    return (ArraySumm());
}
int MeineKlasse::PairMult(){
    for (int i = 1; i < size; i+=2){
        Array[i-1] = Array[i] * Array[i-1];
    }
    return (ArraySumm());
}
int MeineKlasse::ArraySumm(){
    int summ = 0;
    for (int i = 0; i < size; i++){
        summ += Array[i];
    }
    return summ;
}

```

5.3 Файл MeineKlasse.h

Листинг 3 – MeineKlasse.h

```
#ifndef __MEINEKLASSE__H
```

```
#define __MEINEKLASSE__H

class MeineKlasse{
public:
    MeineKlasse();
    MeineKlasse(const MeineKlasse & obj);
    MeineKlasse(int size);
    ~MeineKlasse();
    void ArrayInput();
    int PairSumm();
    int PairMult();
    int ArraySumm();
private:
    int size;
    int* Array;
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 10.

Таблица 10 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
8 1 2 3 4 5 6 7 8	8 Constructor set Copy constructor 120 Destructor 56 Destructor	8 Constructor set Copy constructor 120 Destructor 56 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 (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).