

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.....	9
3 ОПИСАНИЕ АЛГОРИТМОВ.....	10
3.1 Алгоритм функции creationFunction.....	10
3.2 Алгоритм метода ArrayOutput класса MeineKlasse.....	10
3.3 Алгоритм метода ArrayCreate класса MeineKlasse.....	11
3.4 Алгоритм функции main.....	11
3.5 Алгоритм метода PairMult класса MeineKlasse.....	12
3.6 Алгоритм метода PairSumm класса MeineKlasse.....	13
3.7 Алгоритм конструктора класса MeineKlasse.....	13
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	15
5 КОД ПРОГРАММЫ.....	20
5.1 Файл main.cpp.....	20
5.2 Файл MeineKlasse.cpp.....	21
5.3 Файл MeineKlasse.h.....	22
6 ТЕСТИРОВАНИЕ.....	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	24

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, вначале работы выдает сообщение;
- Параметризованный конструктор, передается целочисленный параметр. Параметр должен иметь значение больше 2 и быть четным. Вначале работы выдает сообщение;
- Конструктор копии, обеспечивает создание копии объекта в новой области памяти. Вначале работы выдает сообщение;
- Метод деструктор, который в начале работы выдает сообщение;
- Метод который создает целочисленный массив в закрытой области, согласно ранее заданной размерности.
- Метод ввода данных для созданного массива;
- Метод 1, который суммирует значения очередной пары элементов и сумму присваивает первому элементу пары. 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. 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;
- Метод последовательного вывода содержимого элементов массива,

которые разделены тремя пробелами.

Разработать функцию func, которая имеет один целочисленный параметр, содержащий размерность массива. В функции должен быть реализован алгоритм:

1. Создание локального объекта с использованием параметризованного конструктора.
2. Возврат созданного локального объекта.

В основной функции реализовать алгоритм:

1. Ввод размерности массива.
2. Если размерность массива некорректная, вывод сообщения и завершить работу алгоритма.
3. Вывод значения размерности массива.
4. Создание первого объекта.
5. Присвоение первому объекту результата работы функции func с аргументом, содержащим значение размерности массива.
6. Для первого объекта вызов метода создания массива.
7. Для первого объекта вызов метода ввода данных массива.
8. Для первого объекта вызов метода 2.
9. Инициализация второго объекта первым объектом.
10. Вызов метода 1 для второго объекта.
11. Вывод содержимого массива первого объекта.
12. Вывод суммы элементов массива первого объекта.
13. Вывод содержимого массива второго объекта.
14. Вывод суммы элементов массива второго объекта.

1.1 Описание входных данных

Первая строка:

«Целое число»

Вторая строка:

«Целое число» «Целое число» . . .

Пример:

4
3 5 1 2

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

Параметризованный конструктор в начале работы с новой строки выдает сообщение:

Constructor set

Конструктор копии в начале работы с новой строки выдает сообщение:

Copy constructor

Деструктор в начале работы с новой строки выдает сообщение:

Destructor

Метод последовательного вывода содержимого элементов массива, с новой строки выдает:

«Целое число» «Целое число» «Целое число» . . .

Пример вывода:

```
4
Default constructor
Constructor set
Destructor
Copy constructor
15  5  2  2
24
20  5  4  2
31
Destructor
Destructor
```

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект obj1 класса MeineKlasse предназначен для предназначени для инициализации себя через функцию creationFunction;
- объект obj2 класса MeineKlasse предназначен для предназначен для инициализации себя через конструктор копии;
- функция creationFunction для создание и возврат локализованного объекта.

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 `creationFunction`

Функционал: создание и возврат локализованного объекта класса `MeineKlasse`.

Параметры: `int size` - размер массива.

Возвращаемое значение: объект класса `MeineKlasse`.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции `creationFunction`

№	Предикат	Действия	№ перехода
1		возврат объекта с параметром <code>size</code>	Ø

3.2 Алгоритм метода `ArrayOutput` класса `MeineKlasse`

Функционал: поэлементный вывод массива с тремя пробелами в качестве разделителя.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `ArrayOutput` класса `MeineKlasse`

№	Предикат	Действия	№ перехода
1		инициализация целочисленной перменной <code>i = 0</code>	2

№	Предикат	Действия	№ перехода
2	$i < \text{size} - 1$	вывод элемента с индексом i по указателю Array	3
			4
3		увеличение значения i на 1	4
4		вывод элемента массива с индексом $\text{size} - 1$ по указателю Array и переход на новую строку	\emptyset

3.3 Алгоритм метода ArrayCreate класса MeineKlasse

Функционал: создание массива по указателю Array в закрытой области размером size.

Параметры: int size - размер массива .

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода ArrayCreate класса MeineKlasse

№	Предикат	Действия	№ перехода
1		присваивание Array адрес массива размером size	\emptyset

3.4 Алгоритм функции main

Функционал: выполнение поставленной задачи.

Параметры: нет.

Возвращаемое значение: int - код успешности выполнения программы.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление целочисленного параметра size	2

№	Предикат	Действия	№ перехода
2		ввод size	3
3	size > 2 и size чётный	вывод size и переход на новую строку	4
		вывод "size?"	15
4		создание объекта obj1 через базовый конструктор	5
5		присвоение объекту obj1 результат работы функции creationFunction	6
6		вызов метода ArrayCreation для obj1	7
7		вызов метода ArrayInput для obj1	8
8		вызов метода PairMult для obj1	9
9		инициализация объекта obj2 объектом obj1	10
10		вызов метода PairSumm для obj2	11
11		вызов метода ArrayOutput для obj1	12
12		вывод результата вызова метода ArraySumm объекта obj1 и переход на новую строку	13
13		вызов метода ArrayOutput для obj2	14
14		вывод результата вызова метода ArraySumm объекта obj2 и переход на новую строку	15
15		возврат 0	∅

3.5 Алгоритм метода PairMult класса MeineKlasse

Функционал: подсчёт произведение пар из массива.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода PairMult класса MeineKlasse

№	Предикат	Действия	№ перехода
1		инициализация целочисленной переменной i = 1	2

№	Предикат	Действия	№ перехода
2	$i < \text{size}$	присваивание $i-1$ элементу значение произведения $i-1$ и i элементов	3
			\emptyset
3		увеличение значения i на 2	2

3.6 Алгоритм метода PairSumm класса MeineKlasse

Функционал: подсчёт суммы пар из массива.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода PairSumm класса MeineKlasse

№	Предикат	Действия	№ перехода
1		инициализация целочисленной переменной $i = 1$	2
2	$i < \text{size}$	присваивание $i-1$ элементу значение произведения $i-1$ и i элементов	3
			\emptyset
3		увеличение значения i на 2	2

3.7 Алгоритм конструктора класса MeineKlasse

Функционал: вывод сообщения и сохранения размера массива.

Параметры: `int comes_size` - размер массива.

Алгоритм конструктора представлен в таблице 7.

Таблица 7 – Алгоритм конструктора класса MeineKlasse

№	Предикат	Действия	№ перехода
1		вывод "Constructor set"	2

№	Предикат	Действия	№ перехода
2		приваивание Array значения 0	3
3		приваивание size значения comes_size	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

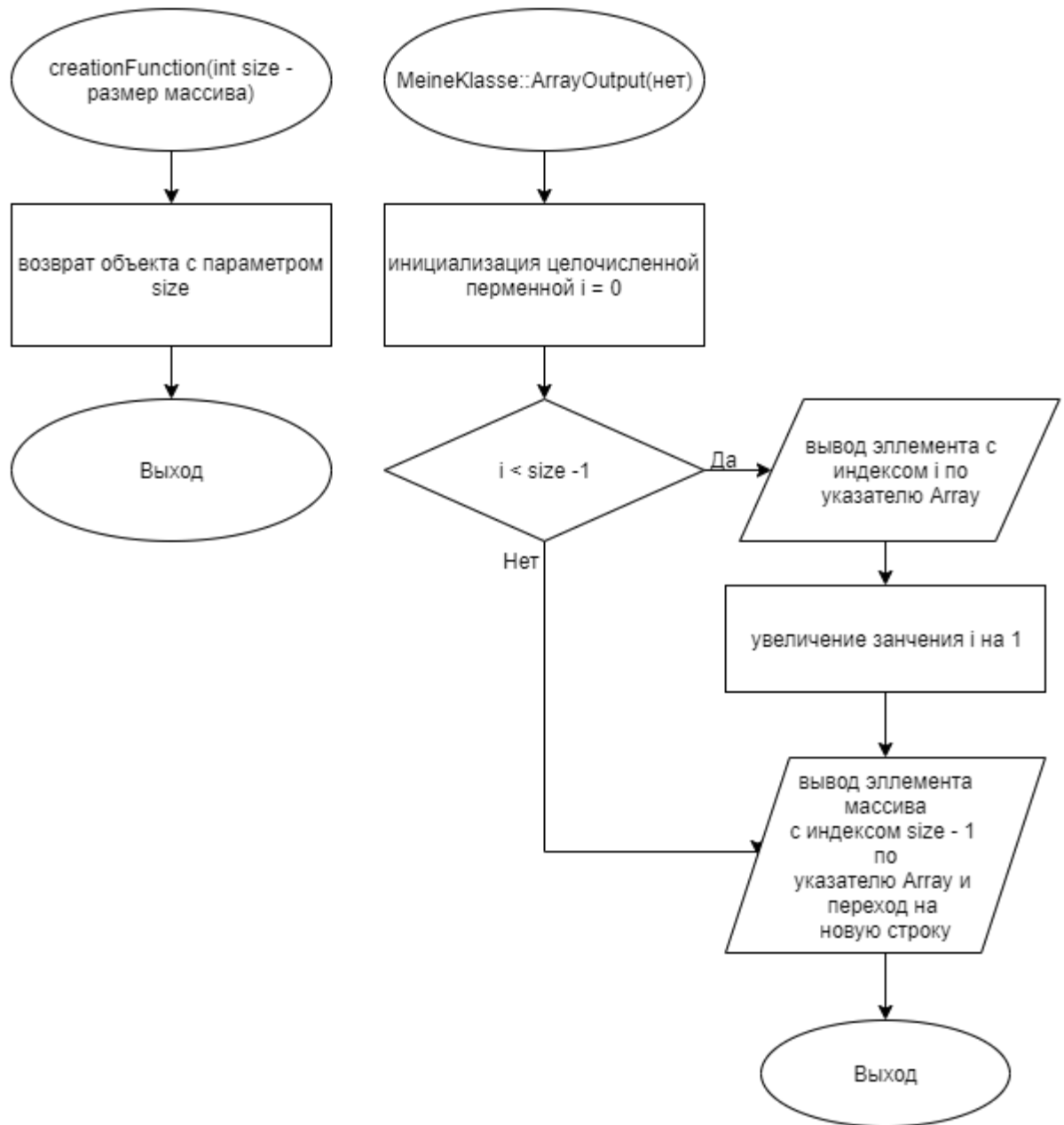


Рисунок 1 – Блок-схема алгоритма

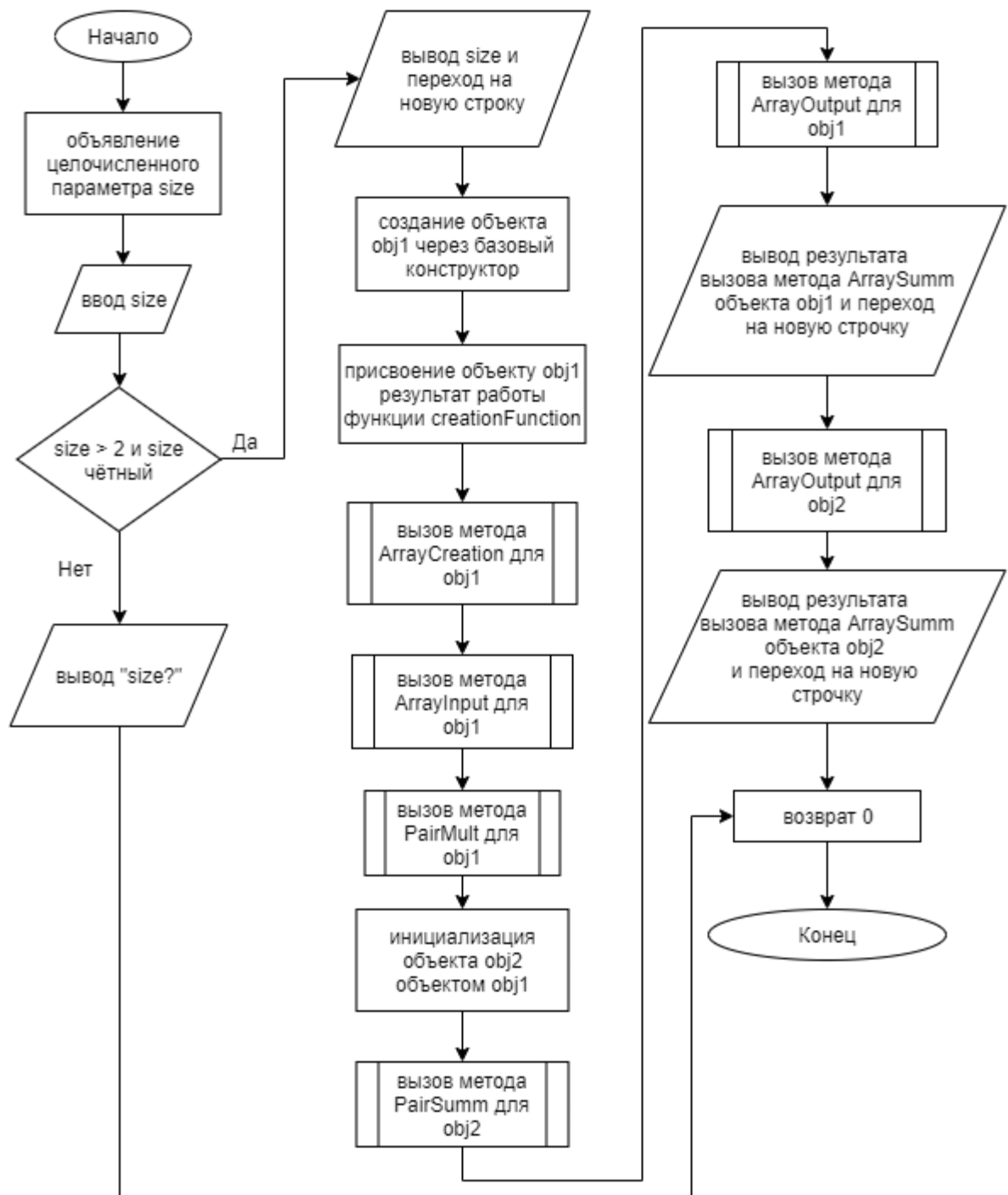


Рисунок 2 – Блок-схема алгоритма

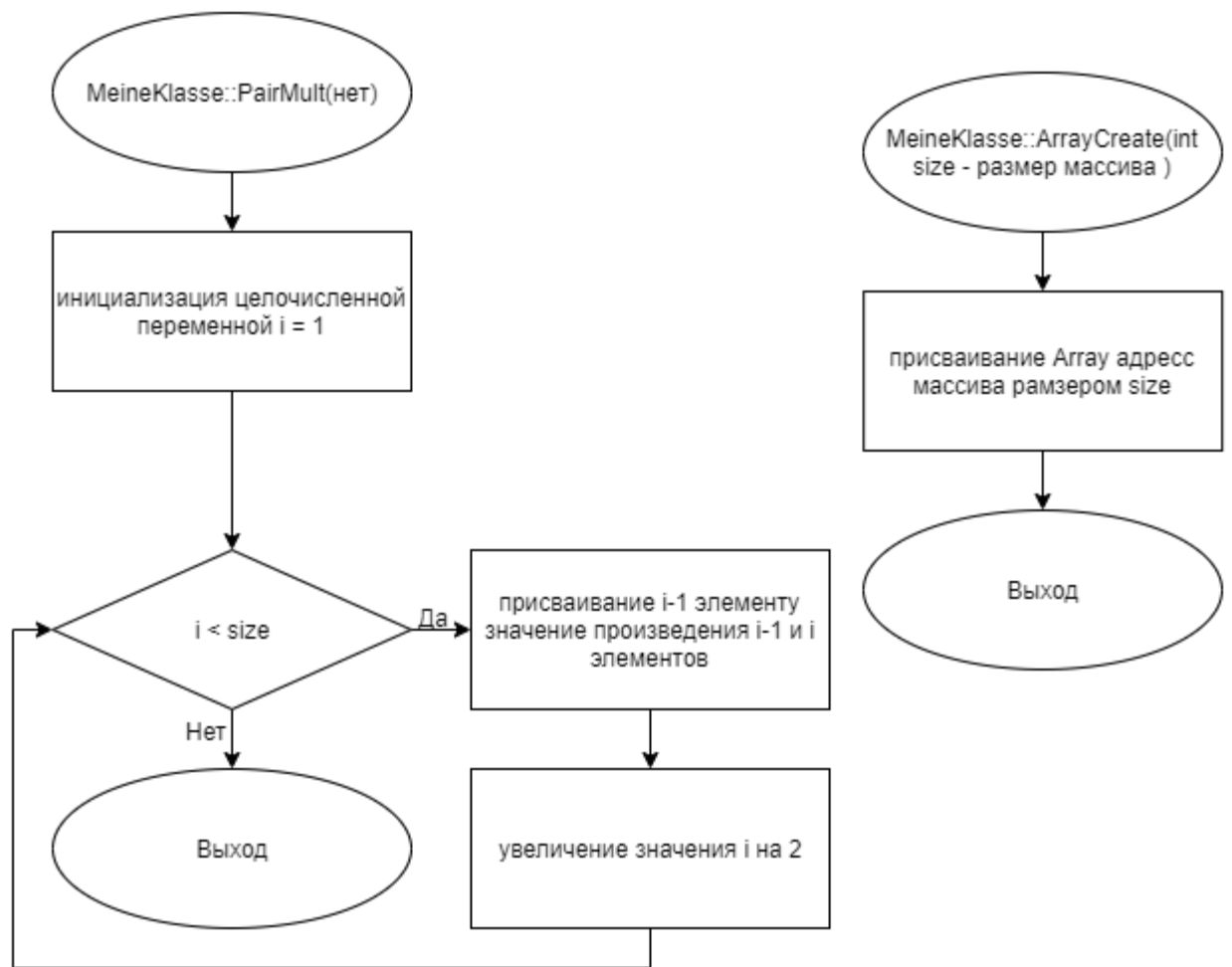


Рисунок 3 – Блок-схема алгоритма

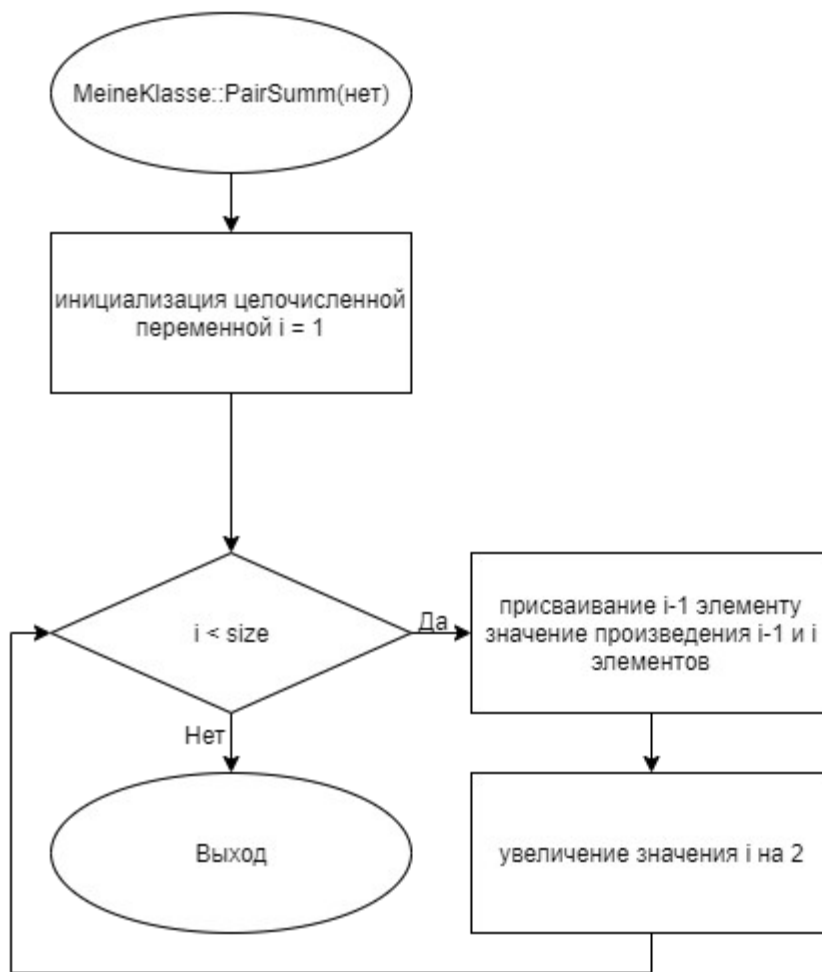


Рисунок 4 – Блок-схема алгоритма



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 <iostream>
#include "MeineKlasse.h"

MeineKlasse creationFunction(int size){
    return MeineKlasse(size);
}

int main()
{
    int size;
    std::cin >> size;
    if ((size > 2) && (size % 2 == 0)){
        std::cout << size << "\n";
        MeineKlasse obj1;
        obj1 = creationFunction(size);
        obj1.ArrayCreate();
        obj1.ArrayInput();
        obj1.PairMult();
        MeineKlasse obj2(obj1);
        obj2.PairSumm();
        obj1.ArrayOutput();
        std::cout << obj1.ArraySumm() << "\n";
        obj2.ArrayOutput();
        std::cout << obj2.ArraySumm() << "\n";
    }
    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 = nullptr;
    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];
    }
}
void MeineKlasse::PairSumm(){
    for (int i = 1; i < size; i+=2){
        Array[i-1] = Array[i] + Array[i-1];
    }
}
void MeineKlasse::PairMult(){
    for (int i = 1; i < size; i+=2){
        Array[i-1] = Array[i] * Array[i-1];
    }
}
int MeineKlasse::ArraySumm(){
    int summ = 0;
    for (int i = 0; i < size; i++){
        summ += Array[i];
    }
    return summ;
}
void MeineKlasse::ArrayCreate(){
    Array = new int[size];
}
void MeineKlasse::ArrayOutput(){
```

```
    for (int i = 0; i < size - 1; ++i){
        std::cout << Array[i] << "  ";
    }
    std::cout << Array[size-1] << "\n";
}
```

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();
    void PairSumm();
    void PairMult();
    int ArraySumm();
    void ArrayOutput();
    void ArrayCreate();
private:
    int size;
    int* Array;
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 8.

Таблица 8 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
4 3 5 1 2	4 Default constructor Constructor set Destructor Copy constructor 15 5 2 2 24 20 5 4 2 31 Destructor Destructor	4 Default constructor Constructor set Destructor Copy constructor 15 5 2 2 24 20 5 4 2 31 Destructor 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).