

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 cl1.....	10
3.2 Алгоритм деструктора класса cl1.....	10
3.3 Алгоритм метода fillarray класса cl1.....	11
3.4 Алгоритм метода SummAll класса cl1.....	11
3.5 Алгоритм метода SummBy1 класса cl1.....	12
3.6 Алгоритм метода MultiplyBy1 класса cl1.....	13
3.7 Алгоритм функции main.....	14
3.8 Алгоритм функции voidfunction.....	15
3.9 Алгоритм конструктора класса cl1.....	15
3.10 Алгоритм конструктора класса cl1.....	16
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	17
5 КОД ПРОГРАММЫ.....	20
5.1 Файл cl1.cpp.....	20
5.2 Файл cl1.h.....	21
5.3 Файл main.cpp.....	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, в начале работы выдает сообщение;
- Параметризованный конструктор, передается целочисленный параметр. Параметр должен иметь значение больше 2 и быть четным. По значению параметра определяется размерность целочисленного массива из закрытой области. 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.
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 `Object1` класса `cl1` предназначен для объект с аргументом размерности массива;
- функция `main` для главная функция программы;
- функция `voidfunction` для вызов и вывод на экран возвращаемого значения метода `MultiplyBy1()` класса `cl1`;
- библиотека `iostream`;
- пространство имён `std`;
- оператор цикла со счётчиком `for`;
- условная конструкция `if else`;
- Объект стандартного потока вывода `cout` на экран;
- Объект стандартного потока ввода `cin` с клавиатуры.

Класс `cl1`:

- свойства/поля:
 - поле `Отвечает за объявление указателя на динамический массив`:
 - наименование — `array`;
 - тип — `int*`;
 - модификатор доступа — `private`;
 - поле `Отвечает за хранение размера массива array (скрытого поля класса cl1)`:
 - наименование — `sizeofarray`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - метод `cl1` — конструктор по умолчанию (не параметризованный).

Выводит сообщение о работе на экран;

- о метод `cl1` — параметризованный конструктор с целочисленным параметром, по значению которого определяется размер целочисленного массива `array` из закрытой области данных. Выводит сообщение о работе на экран;

- о метод `cl1` — конструктор копии. Копирует объект класса `cl1` для корректного выполнения вычислений в методах самого класса;

- о метод `~cl1` — деструктор. Выводит сообщение о работе на экран, уничтожает объект и очищает память, выделенную под динамический массив `array`;

- о метод `fillarray` — заполнение массива `array` значениями, введёнными с клавиатуры;

- о метод `SummAll` — суммирование значений элементов массива `array` и возвращает результат суммирования;

- о метод `SummBy1` — суммирование значений очередной пары элементов и присваивание результата суммы первому элементу пары;

- о метод `MultiplyBy1` — умножение значений очередной пары элементов и присваивание результата суммы первому элементу пары.

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 *cl1*

Функционал: конструктор по умолчанию (не параметризованный).
Выводит сообщение о работе на экран.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса *cl1*

№	Предикат	Действия	№ перехода
1		Вывод: "Default constructor"	Ø

3.2 Алгоритм деструктора класса *cl1*

Функционал: Выводит сообщение о работе на экран, уничтожает объект и очищает память, выделенную под динамический массив *array*.

Параметры: нет.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса *cl1*

№	Предикат	Действия	№ перехода
1		Вывод: "Destructor"	2
2	Значение поля <i>array</i> объекта не равно значению <i>nullptr</i>	Очищение выделенной памяти для поля <i>array</i> посредством оператора <i>delete</i>	Ø

№	Предикат	Действия	№ перехода
	Значение поля array объекта не равно значению nullptr		Ø

3.3 Алгоритм метода fillarray класса cl1

Функционал: заполнение массива array значениями, введенными с клавиатуры.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода fillarray класса cl1

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной i значением 0	2
2	Значение переменной i меньше значения переменной sizeofarray	Вывод: <<значение i-го элемента массива array>>	3
	Значение переменной i не меньше значения переменной sizeofarray		Ø
3		Инкремент значения переменной i	2

3.4 Алгоритм метода SummAll класса cl1

Функционал: суммирование значений элементов массива array и возвращает результат суммирования.

Параметры: нет.

Возвращаемое значение: целое, сумма всех элементов массива array.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *SummAll* класса *cl1*

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной <i>summ</i> значением 0	2
2		Инициализация целочисленной переменной <i>i</i> значением 0	3
3	Значение переменной <i>i</i> меньше значения переменной <i>sizeofarray</i>	Увеличение значение переменной <i>summ</i> на значение <i>i</i> -го элемента массива <i>array</i>	4
	Значение переменной <i>i</i> не меньше значения переменной <i>sizeofarray</i>		5
4		Инкремент переменной <i>i</i>	3
5		Возвращение значения переменной <i>summ</i>	∅

3.5 Алгоритм метода *SummBy1* класса *cl1*

Функционал: суммирование значений очередной пары элементов и присваивание результата суммы первому элементу пары.

Параметры: нет.

Возвращаемое значение: целое, сумма всех элементов массива *array*.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *SummBy1* класса *cl1*

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной <i>i</i> значением 0	2
2	Значение переменной <i>i</i> меньше значения	Присваивание <i>i</i> -му элементу массива <i>array</i> значение суммы <i>i</i> -го и (<i>i</i> +1)-го элементов массива	3

№	Предикат	Действия	№ перехода
	переменной sizeofarray	array	
	Значение переменной i не меньше значения переменной sizeofarray		4
3		Увеличение значения переменной i на 2	2
4		Вызов метода SummAll() и возвращение результата его выполнения	∅

3.6 Алгоритм метода MultiplyBy1 класса cl1

Функционал: умножение значений очередной пары элементов и присваивание результата суммы первому элементу пары.

Параметры: нет.

Возвращаемое значение: целое, сумма элементов массива array.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода MultiplyBy1 класса cl1

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной i значением 0	2
2	Значение переменной i меньше значения переменной sizeofarray	Присваивание i-му элементу массива array значение произведения i-го и (i+1)-го элементов массива array	3
	Значение переменной i не меньше значения переменной sizeofarray		4
3		Увеличение значения переменной i на 2	2
4		Вызов метода SummAll() и возвращение результата его выполнения	∅

3.7 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое, индикация корректности работы программы.

Алгоритм функции представлен в таблице 7.

Таблица 7 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной sizeofarray	2
2		Ввод значения переменной sizeofarray с клавиатуры	3
3	Значение переменной sizeofarray больше 2 и чётное	Вывод: <<значение sizeofarray>>	4
	Значение переменной sizeofarray меньше или равно 2 и нечётное	Вывод: <<значение sizeofarray>>"?"	Ø
4		Создание объекта Object1 класса Object посредством вызова параметризованного конструктора с передаваемым аргументом sizeofarray	5
5		Вызов метода fillarray() объекта Object1	6
6		Вызов функции voidfunction() с передаваемым аргументом Object1	7
7		Переход на новую строку	8
8		Вызов метода SummBy1() объекта Object1	9
9		Вывод: <<возвращаемое значение метода SummBy1() объекта Object1>>	Ø

3.8 Алгоритм функции voidfunction

Функционал: создание нового объекта через конструктор копии и вызов методов данного объекта.

Параметры: cl1 Object1, копия объекта класса cl1, созданная посредством конструктора копии.

Возвращаемое значение: Отсутствует.

Алгоритм функции представлен в таблице 8.

Таблица 8 – Алгоритм функции voidfunction

№	Предикат	Действия	№ перехода
1		Вызов метода MultiplyBy1 объекта Object1	2
2		Вывод: <<возвращаемое значение метода MultiplyBy1 объекта Object1>>	Ø

3.9 Алгоритм конструктора класса cl1

Функционал: параметризованный конструктор с целочисленным параметром, по значению которого определяется размер целочисленного массива array из закрытой области данных. Выводит сообщение о работе на экран.

Параметры: нет.

Алгоритм конструктора представлен в таблице 9.

Таблица 9 – Алгоритм конструктора класса cl1

№	Предикат	Действия	№ перехода
1		Вывод: "Constructor set"	2
2		Выделение памяти под динамический массив array размером sizeofarray через оператор new	3
3		Присваивание полю array класса cl1 значения параметра sizeofarray	Ø

3.10 Алгоритм конструктора класса cl1

Функционал: конструктор копии. Копирует объект класса cl1 для корректного выполнения вычислений в методах самого класса.

Параметры: const cl1 &Object1, объект, значение полей которого используются при создании копии данного объекта.

Алгоритм конструктора представлен в таблице 10.

Таблица 10 – Алгоритм конструктора класса cl1

№	Предикат	Действия	№ перехода
1		Вывод: "Copy constructor"	2
2		Присваивание полю sizeofarray создаваемого объекта значение поля sizeofarray объекта копирования	3
3		Выделение памяти под динамический массив array создаваемого объекта размером sizeofarray посредством оператора new	4
4		Инициализация целочисленной переменной i значением 0	5
5	Значение переменной i меньше значения переменной sizeofarray	Присваивание i-ому элементу массива array создаваемого объекта значение i-той ячейки массива array объекта копирования Object1	6
	Значение переменной i не меньше значения переменной sizeofarray		Ø
6		Инкремент значения переменной i	5

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

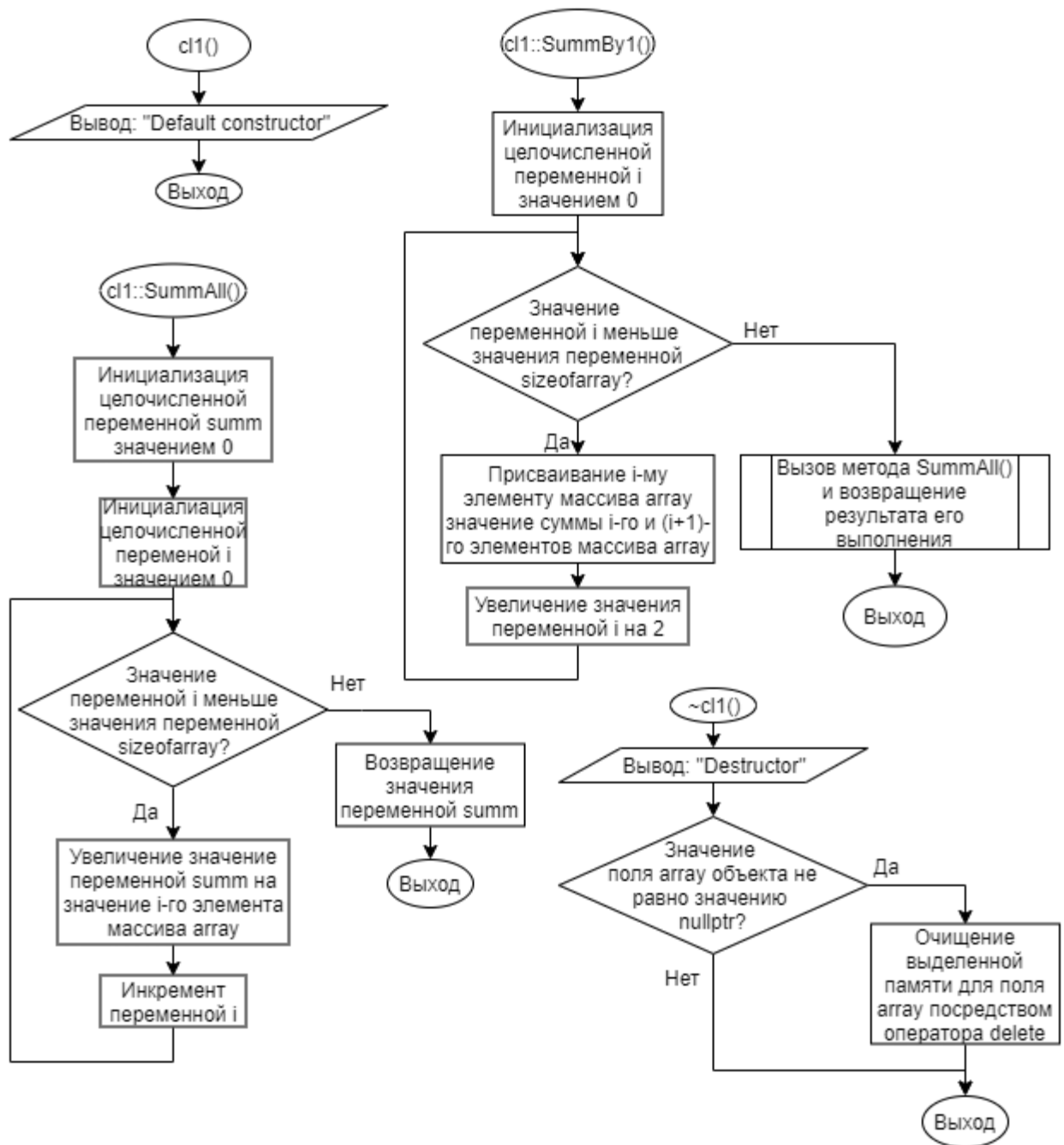


Рисунок 1 – Блок-схема алгоритма

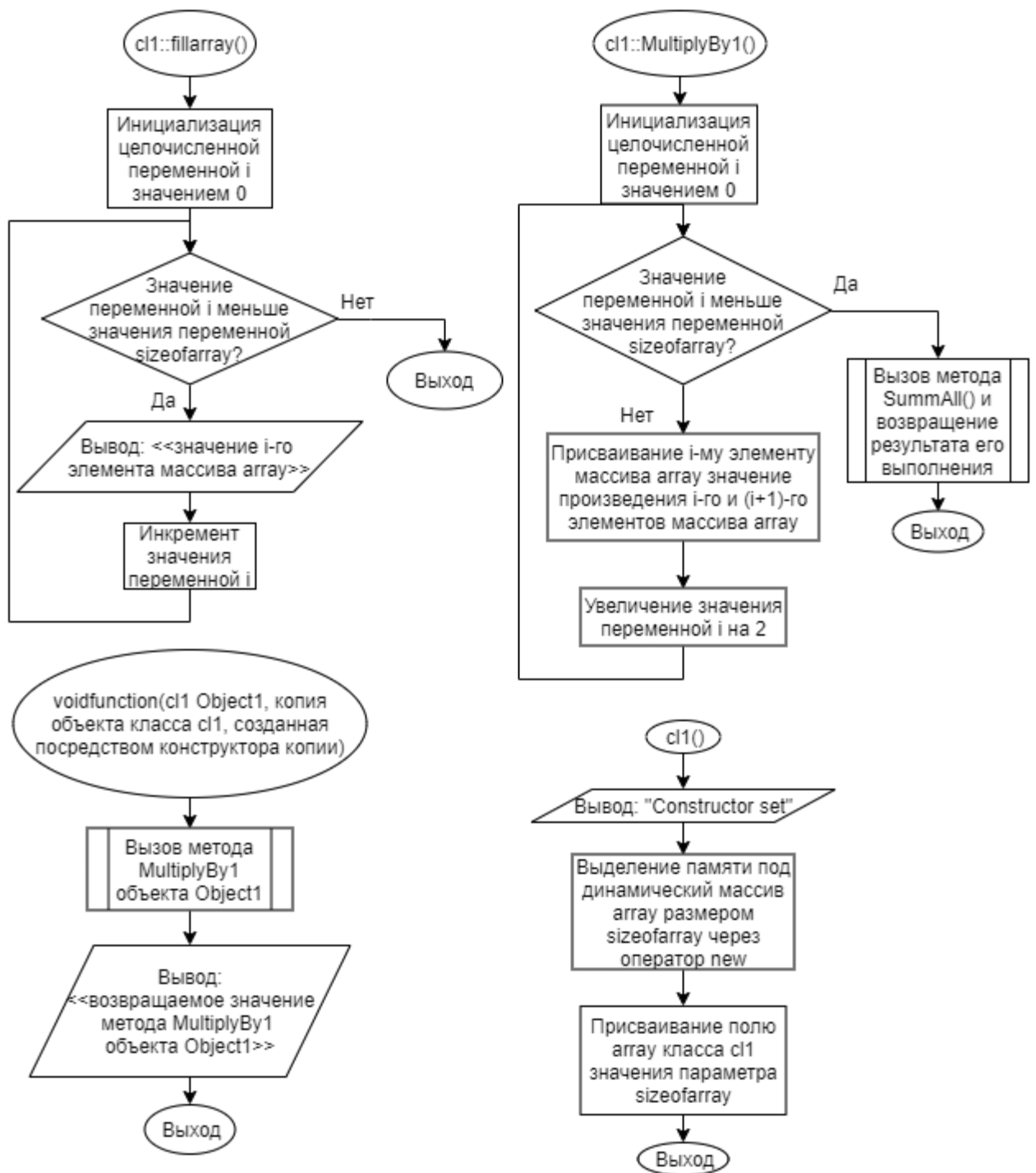


Рисунок 2 – Блок-схема алгоритма

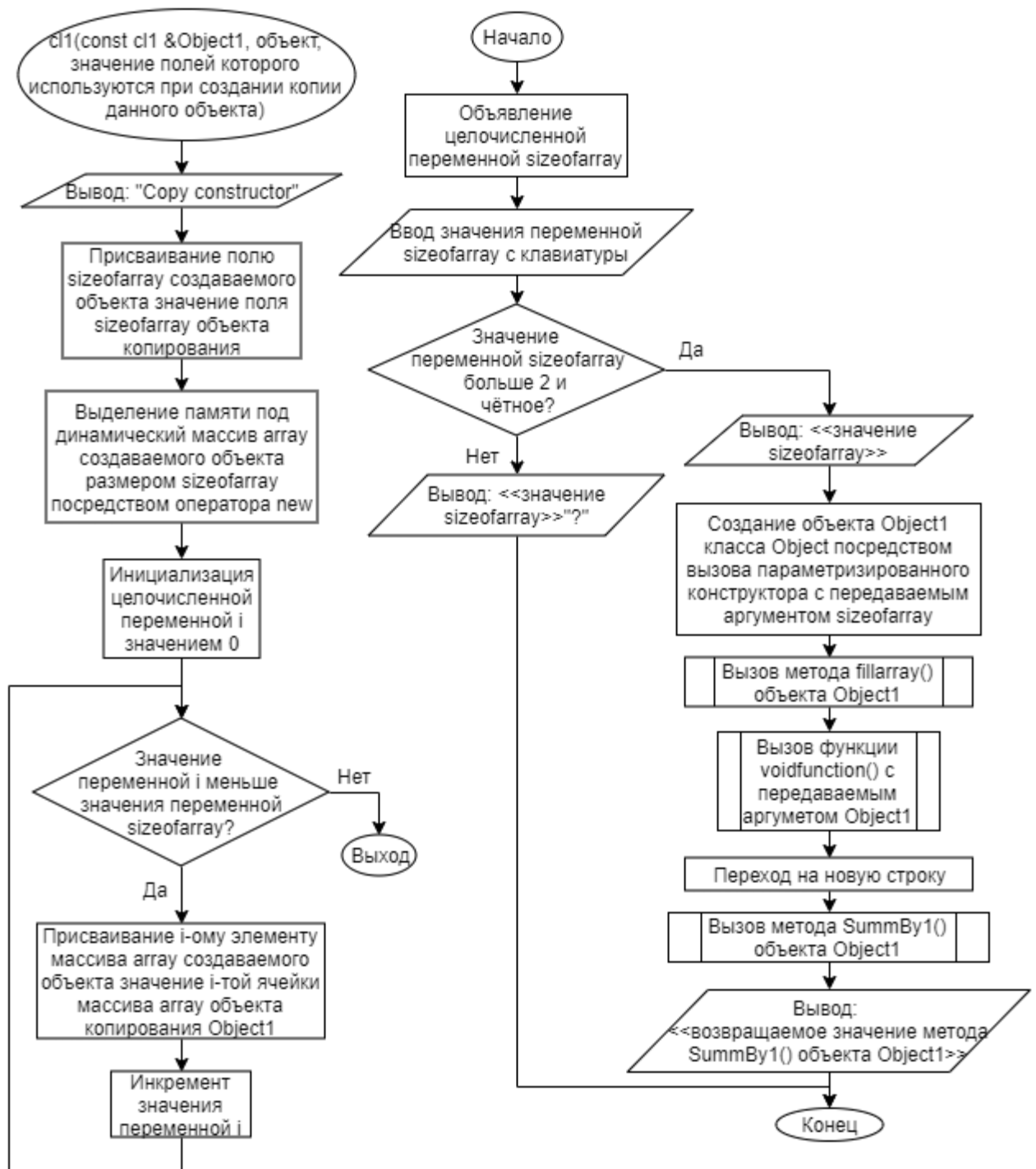


Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл cl1.cpp

Листинг 1 – cl1.cpp

```
#include <iostream>
#include "cl1.h"
using namespace std;
cl1::cl1()
{
    cout << "Default constructor" << endl;
}
cl1::cl1(int sizeofarray)
{
    cout << "Constructor set" << endl;
    array = new int[sizeofarray];
    this -> sizeofarray = sizeofarray;
}
cl1::cl1(const cl1 &Object1)
{
    cout << "Copy constructor" << endl;
    sizeofarray = Object1(sizeofarray);
    array = new int[sizeofarray];
    for (int i = 0; i < sizeofarray; i++)
    {
        array[i] = Object1.array[i];
    }
}
cl1::~cl1()
{
    cout << "Destructor";
    if(array != nullptr)
    {
        delete[] array;
    }
}
void cl1::fillarray()
{
    for (int i = 0; i < sizeofarray; i++)
    {
        cin >> array[i];
    }
}
int cl1::SummAll()
```

```

{
    int summ = 0;
    for (int i = 0; i < sizeofarray; i++)
    {
        summ += array[i];
    }
    return summ;
}
int cl1::SummBy1()
{
    for (int i = 0; i < sizeofarray; i += 2)
    {
        array[i] += array[i+1];
    }
    return SummAll();
}
int cl1::MultiplyBy1()
{
    for (int i = 0; i < sizeofarray; i+= 2)
    {
        array[i] *= array[i+1];
    }
    return SummAll();
}

```

5.2 Файл cl1.h

Листинг 2 – cl1.h

```

#ifndef __CL1__H
#define __CL1__H
class cl1
{
    int* array;
    int sizeofarray;
public:
    cl1();
    cl1(int sizeofarray);
    cl1(const cl1 &Object1);
    ~cl1();
    void fillarray();
    int SummAll();
    int SummBy1();
    int MultiplyBy1();
};
#endif

```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <iostream>
#include "cl1.h"
using namespace std;

void voidfunction(cl1 Object1)
{
    cout << Object1.MultiplyBy1() << endl;
}
int main()
{
    int arraysize;
    cin >> arraysize;
    if(arraysize > 2 && arraysize%2 == 0)
    {
        cout << arraysize << endl;
        cl1 Object1(arraysize);
        Object1.fillarray();
        voidfunction(Object1);
        cout << endl;
        cout << Object1.SumBy1() << endl;
    }
    else
    {
        cout << arraysize << "?";
    }
    return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 11.

Таблица 11 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
10 1 2 3 4 5 6 7 8 9 10	10 Constructor set Copy constructor 220 Destructor 85 Destructor	10 Constructor set Copy constructor 220 Destructor 85 Destructor
3 1 2 3	3?	3?
5 1 2 3 9 6	5?	5?
6 1 5 20 9 5 4	6 Constructor set Copy constructor 223 Destructor 62 Destructor	6 Constructor set Copy constructor 223 Destructor 62 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).