

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..... | 11 |
| 3.1 Алгоритм конструктора класса cl1..... | 11 |
| 3.2 Алгоритм деструктора класса cl1..... | 11 |
| 3.3 Алгоритм метода fillarray класса cl1..... | 12 |
| 3.4 Алгоритм метода SummAll класса cl1..... | 12 |
| 3.5 Алгоритм метода SummBy1 класса cl1..... | 13 |
| 3.6 Алгоритм метода MultiplyBy1 класса cl1..... | 14 |
| 3.7 Алгоритм метода CreateArray класса cl1..... | 14 |
| 3.8 Алгоритм метода OutputArray класса cl1..... | 15 |
| 3.9 Алгоритм конструктора класса cl1..... | 16 |
| 3.10 Алгоритм функции function..... | 16 |
| 3.11 Алгоритм функции main..... | 17 |
| 4 БЛОК-СХЕМЫ АЛГОРИТМОВ..... | 19 |
| 5 КОД ПРОГРАММЫ..... | 22 |
| 5.1 Файл cl1.cpp..... | 22 |
| 5.2 Файл cl1.h..... | 23 |
| 5.3 Файл main.cpp..... | 24 |
| 6 ТЕСТИРОВАНИЕ..... | 25 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... | 26 |

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 Object2 класса cl1 предназначен для объект с аргументом в виде объекта Object1;
- объект Object3 класса cl1 предназначен для объект с аргументом размерности массива;
- объект Object1 класса cl1 предназначен для объект с аргументом размерности массива;
- функция function для создание и возврат объекта Object3 класса cl1 с аргументом размерности массива;
- функция main для главная функция программы;
- библиотека iostream;
- пространство имён std;
- оператор цикла со счётчиком;
- условная конструкция if else;
- Объект cin стандартного потока ввода с клавиатуры;
- Объект cout стандартного потока вывода на экран.

Класс cl1:

- свойства/поля:
 - поле объявление указателя на динамический массив:
 - наименование — array;
 - тип — int*;
 - модификатор доступа — private;
 - поле хранение размера массива array (скрытого поля класса cl1):
 - наименование — sizeofarray;
 - тип — int;

- модификатор доступа — `private`;
- функционал:
 - о метод `cl1` — конструктор по умолчанию. 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` — деструктор. Выводит сообщение о работе на экран. Уничтожает объект;
 - о метод `fillarray` — заполнение массива `array` значениями, введёнными пользователем с клавиатуры;
 - о метод `SummAll` — суммирование значений элементов массива `array` и возврат результата суммирования;
 - о метод `SummBy1` — суммирование значений очередной пары элементов массива `array` и присваивание результата суммы первому элементу пары;
 - о метод `MultiplyBy1` — умножение значений очередной пары элементов массива `array` и присваивание результата суммы первому элементу пары;
 - о метод `CreateArray` — создание целочисленного массива `array` в закрытой области;
 - о метод `OutputArray` — последовательный вывод элементов массива `array`, разделённых тремя пробелами.

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. 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: деструктор. Выводит сообщение о работе на экран. Уничтожает объект. 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 | | Переход на новую строку | 2 |
| 2 | | Вывод: "Destructor" | Ø |

3.3 Алгоритм метода fillarray класса cl1

Функционал: заполнение массива array значениями, введёнными пользователем 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 возврат результата суммирования.

Параметры: нет.

Возвращаемое значение: целое, сумма всех элементов массива.

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 значения 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 массива *array* и присваивание результата суммы первому элементу пары.

Параметры: нет.

Возвращаемое значение: Отсутствует.

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 переменной <i>sizeofarray</i> | Присваивание <i>i</i> -ому элементу массива <i>array</i> значения суммы <i>i</i> -го и (<i>i</i> +1)-го элементов массива <i>array</i> | 3 |

| № | Предикат | Действия | № перехода |
|---|--|---|---------------|
| | Значение переменной i не меньше значения переменной <code>sizeofarray</code> | | \emptyset |
| 3 | | Увеличение значения переменной i на 2 | 2 |

3.6 Алгоритм метода `MultiplyBy1` класса `cl1`

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

Параметры: нет.

Возвращаемое значение: Отсутствует.

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 <code>sizeofarray</code> | Присваивание i -ому элементу массива <code>array</code> значения произведения i -го и $(i+1)$ -го элементов массива <code>array</code> | 3 |
| | Значение переменной i меньше значения переменной <code>sizeofarray</code> | | \emptyset |
| 3 | | Увеличение значения переменной i на 2 | 2 |

3.7 Алгоритм метода `CreateArray` класса `cl1`

Функционал: создание целочисленного массива `array` в закрытой области.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода *CreateArray* класса *cl1*

| № | Предикат | Действия | № перехода |
|---|----------|--|---------------|
| 1 | | Выделение памяти под динамический массив <i>array</i> размером <i>sizeofarray</i> посредством оператора <i>new</i> | Ø |

3.8 Алгоритм метода *OutputArray* класса *cl1*

Функционал: последовательный вывод элементов массива *array*, разделённых тремя пробелами.

Параметры: нет.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода *OutputArray* класса *cl1*

| № | Предикат | Действия | № перехода |
|---|--|--|---------------|
| 1 | | Вывод: <<значение 1-го элемента массива <i>array</i> >> | 2 |
| 2 | | Инициализация целочисленной переменной <i>i</i> значением 1 | 3 |
| 3 | Значение переменной <i>i</i> меньше значения <i>sizeofarray</i> | Вывод: " ", <<значение <i>i</i> -го элемента массива <i>array</i> >> | 4 |
| | Значение переменной <i>i</i> не меньше значения <i>sizeofarray</i> | Переход на новую строку | Ø |
| 4 | | Инкремент значения переменной <i>i</i> | 3 |

3.9 Алгоритм конструктора класса cl1

Функционал: параметризированный 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 | | Переход на новую строку | 2 |
| 2 | | Вывод: "Copy constructor" | 3 |
| 3 | | Присваивание полю sizeofarray создаваемого объекта значения поля sizeofarray объекта копирования | 4 |
| 4 | | Выделение памяти под динамический массив array создаваемого объекта размером sizeofarray посредством оператора new | 5 |
| 5 | | Инициализация целочисленной переменной i значением 0 | 6 |
| 6 | Значение переменной i меньше значения переменной sizeofarray | Присваивание i-ому элементу массива array создаваемого объекта значение i-ой ячейки массива array объекта копирования Object1 | 7 |
| | Значение переменной i не меньше значения переменной sizeofarray | | Ø |
| 7 | | Инкремент значения переменной i | 6 |

3.10 Алгоритм функции function

Функционал: создание и возврат объекта Object3 класса cl1 с аргументом

размерности массива.

Параметры: нет.

Возвращаемое значение: создаваемый объект Object3 класса cl1 с аргументом размерности массива array.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции function

| № | Предикат | Действия | № перехода |
|---|----------|---|------------|
| 1 | | Создание объекта Object3 класса cl1, путём вызова параметризованного конструктора с аргументом в виде параметра sizeofarray | 2 |
| 2 | | Возврат значения созданного объекта Object3 класса cl1 | Ø |

3.11 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое, индикация корректности работы программы.

Алгоритм функции представлен в таблице 11.

Таблица 11 – 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 не больше 2 или | Вывод: <<значение sizeofarray>>"?" | Ø |

| № | Предикат | Действия | № перехода |
|----|----------|---|---------------|
| | нечётное | | |
| 4 | | Создание объекта Object1 класса cl1 посредством вызова конструктора по умолчанию | 5 |
| 5 | | Присваивание объекту Object1 значение выполнения функции function с аргументом в виде значения переменной sizeofarray | 6 |
| 6 | | Вызов метода CreateArray() объекта Object1 | 7 |
| 7 | | Вызов метода fillArray() объекта Object1 | 8 |
| 8 | | Вызов метода MultiplyBy1() объекта Object1 | 9 |
| 9 | | Создание объекта Object2 класса cl1 посредством вызова конструктора копирования с аргументом в виде объекта Object1 | 10 |
| 10 | | Вызов метода SummBy1() объекта Object2 | 11 |
| 11 | | Вызов метода OutputArray() объекта Object1 | 12 |
| 12 | | Вызов метода SummAll() объекта Object1 | 13 |
| 13 | | Вывод: <<результат выполнения метода SummAll() объекта Object1>> | 14 |
| 14 | | Вызов метода OutputArray() объекта Object2 | 15 |
| 15 | | Вызов метода SummAll() объекта Object2 | 16 |
| 16 | | Вывод: <<результат выполнения метода SummAll() объекта Object2>> | ∅ |

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

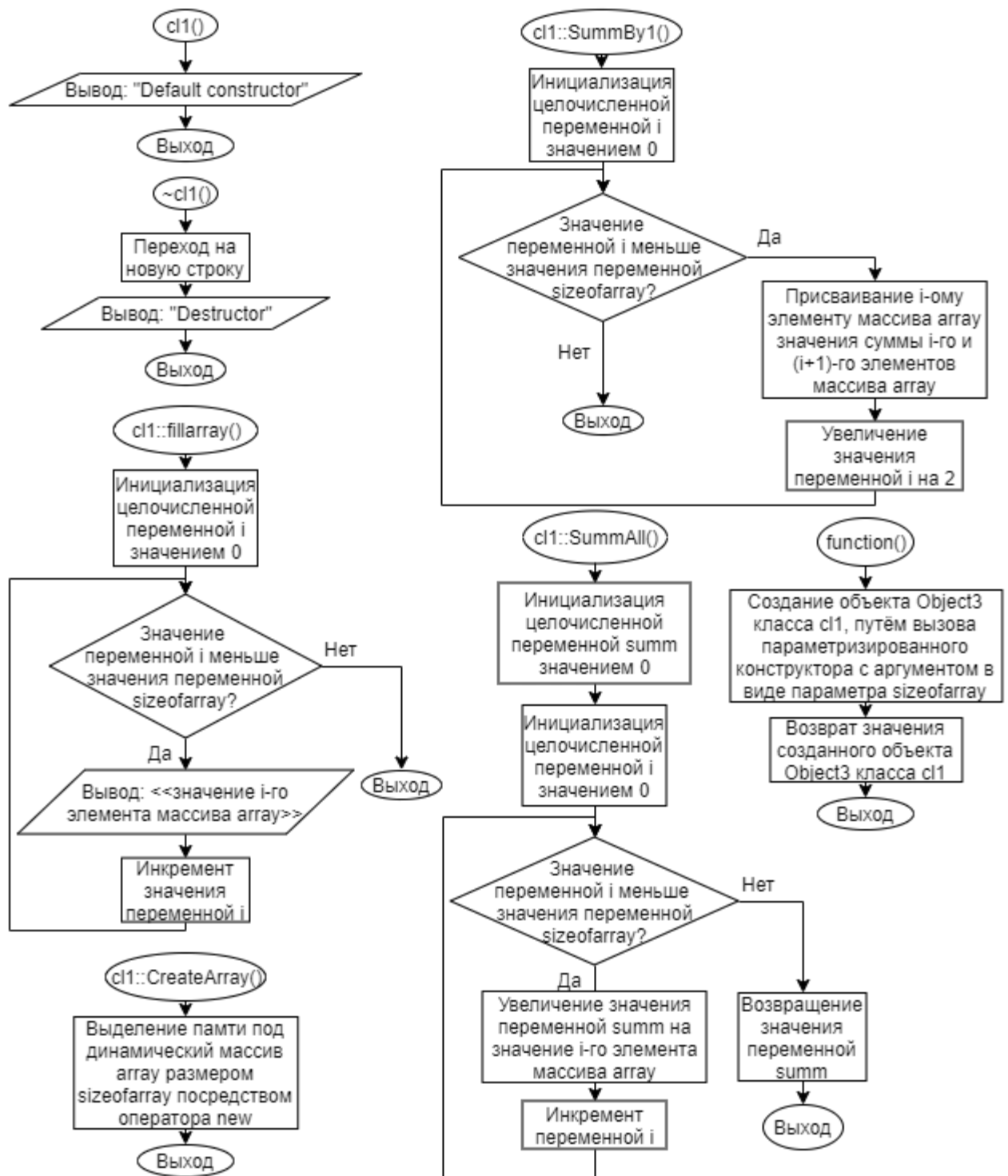


Рисунок 1 – Блок-схема алгоритма

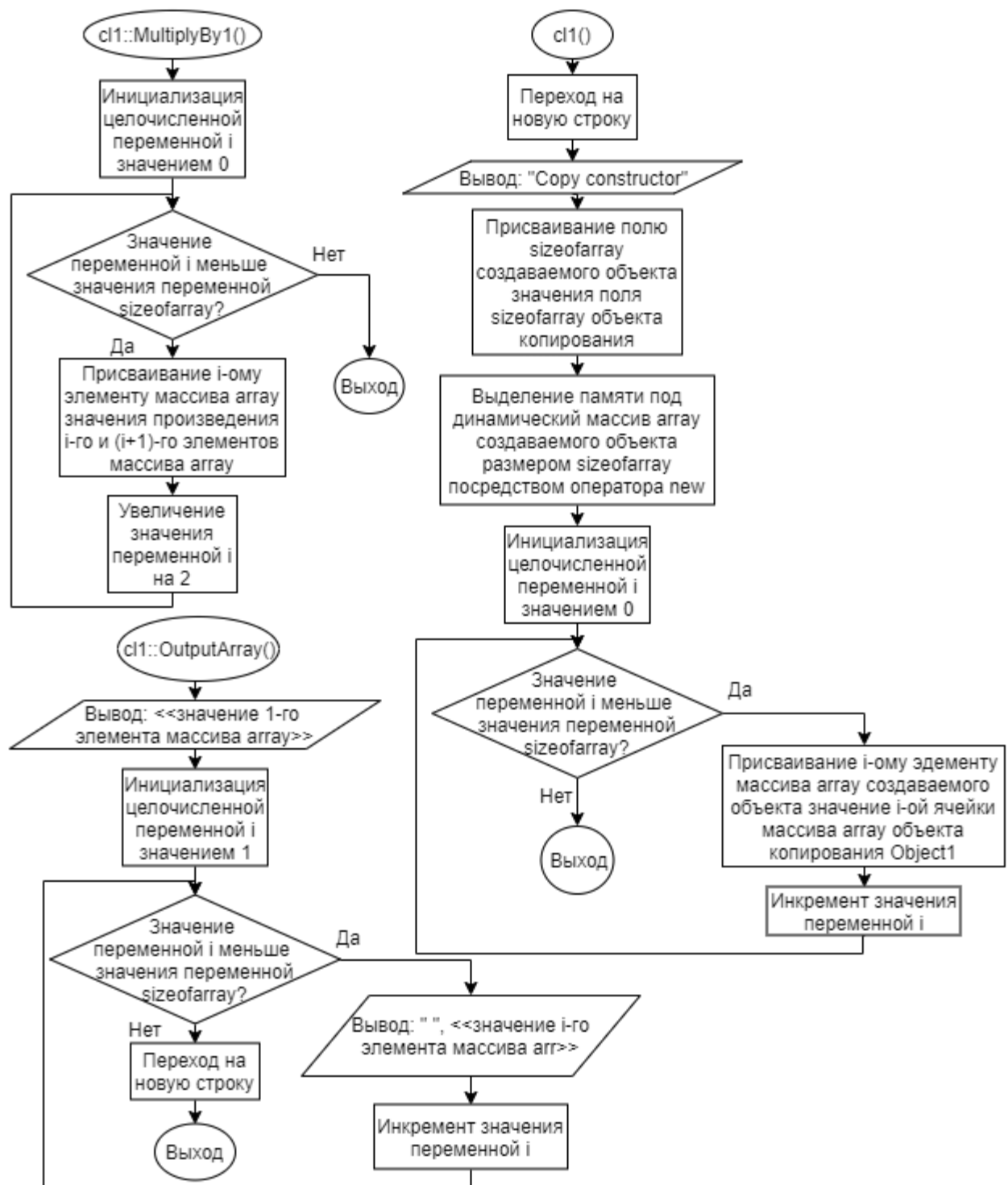


Рисунок 2 – Блок-схема алгоритма

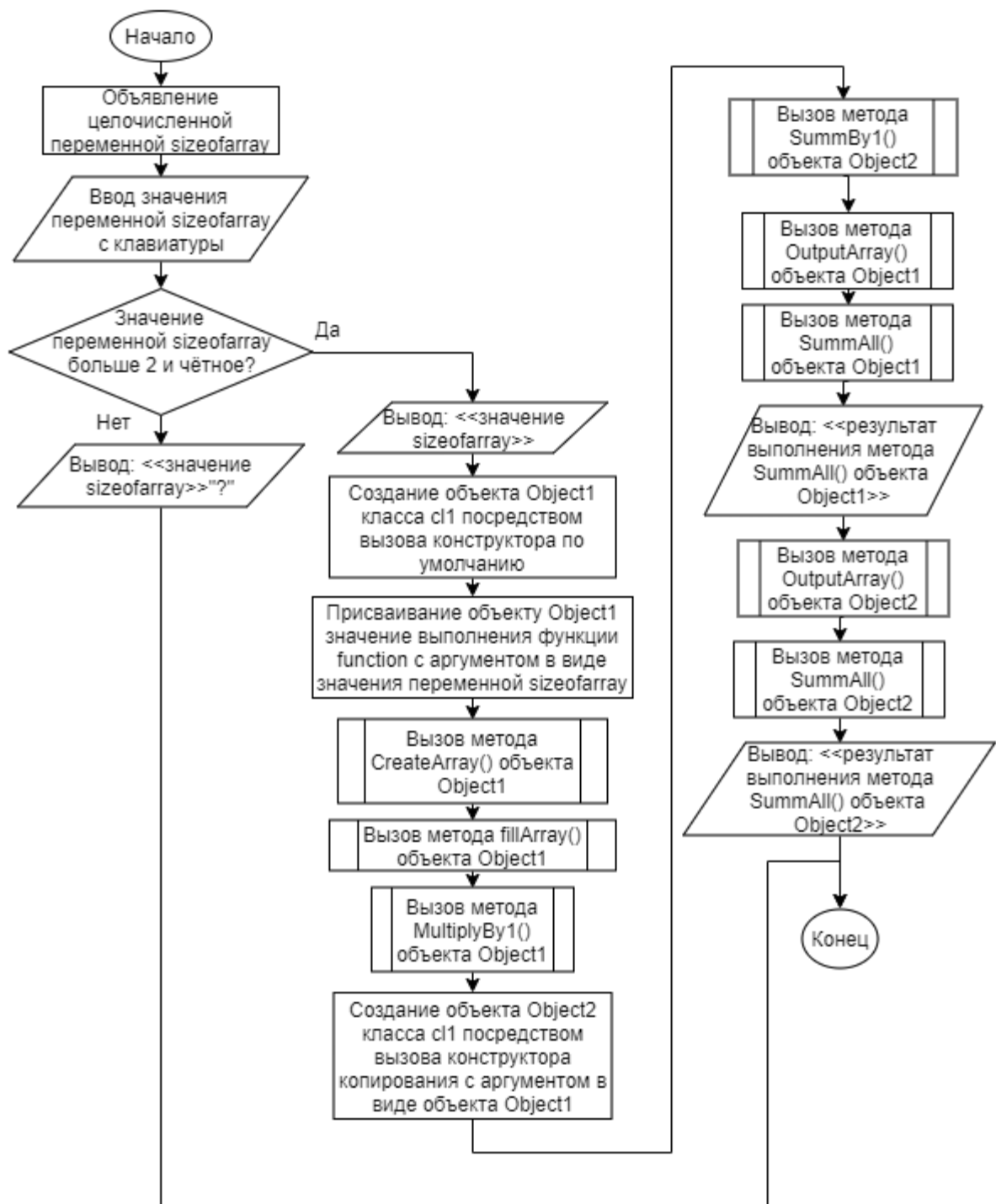


Рисунок 3 – Блок-схема алгоритма

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";
    this -> sizeofarray = sizeofarray;
}
cl1::cl1(const cl1 &Object1)
{
    cout << "\nCopy constructor" << endl;
    sizeofarray = Object1(sizeofarray);
    array = new int[sizeofarray];
    for (int i = 0; i < sizeofarray; i++)
    {
        array[i] = Object1.array[i];
    }
}
cl1::~cl1()
{
    cout << "\nDestructor";
}
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;
}
void cl1::SummBy1()
{
    for (int i = 0; i < sizeofarray; i += 2)
    {
        array[i] += array[i+1];
    }
}
void cl1::MultiplyBy1()
{
    for (int i = 0; i < sizeofarray; i+= 2)
    {
        array[i] *= array[i+1];
    }
}
void cl1::CreateArray()
{
    array = new int[sizeofarray];
}
void cl1::OutputArray()
{
    cout << array[0];
    for (int i = 1; i < sizeofarray; i++)
    {
        cout << "    " << array[i];
    }
    cout << endl;
}

```

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();
    void SummBy1();
    void MultiplyBy1();
}

```

```
        void CreateArray();
        void OutputArray();
    };
#endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

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

cl1 function(int arraysize)
{
    cl1 Object3(arraysize);
    return Object3;
}

int main()
{
    int arraysize;
    cin >> arraysize;
    if(arraysize > 2 && arraysize%2 == 0)
    {
        cout << arraysize << endl;
        cl1 Object1;
        Object1 = function(arraysize);
        Object1.CreateArray();
        Object1.fillarray();
        Object1.MultiplyBy1();
        cl1 Object2(Object1);
        Object2.SummBy1();
        Object1.OutputArray();
        cout << Object1.SummAll() << endl;
        Object2.OutputArray();
        cout << Object2.SummAll();
    }
    else
    {
        cout << arraysize << "?";
    }
    return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 12.

Таблица 12 – Результат тестирования программы

| Входные данные | Ожидаемые выходные данные | Фактические выходные данные |
|----------------------|---|---|
| 7 9 8 2 6 | 7? | 7? |
| 8 6 5 4 3 2 1 7 8 | 8 Default constructor Constructor set Destructor Copy constructor 30 5 12 3 2 1 56 8 117 35 5 15 3 3 1 64 8 134 Destructor Destructor | 8 Default constructor Constructor set Destructor Copy constructor 30 5 12 3 2 1 56 8 117 35 5 15 3 3 1 64 8 134 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/methodichescoc_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).