

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.....	9
3.1 Алгоритм конструктора класса Cls.....	9
3.2 Алгоритм деструктора класса Cls.....	9
3.3 Алгоритм конструктора класса Cls.....	9
3.4 Алгоритм метода m1 класса Cls.....	10
3.5 Алгоритм метода m2 класса Cls.....	10
3.6 Алгоритм метода m3 класса Cls.....	11
3.7 Алгоритм метода input класса Cls.....	11
3.8 Алгоритм конструктора класса Cls.....	12
3.9 Алгоритм функции main.....	12
3.10 Алгоритм функции func.....	13
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	14
5 КОД ПРОГРАММЫ.....	21
5.1 Файл Cls.cpp.....	21
5.2 Файл Cls.h.....	22
5.3 Файл main.cpp.....	23
6 ТЕСТИРОВАНИЕ.....	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	25

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 obj класса Cls предназначен для Объект для решения задачи;
- функция func для Демонстрация работы деструктора и копирования;
- cin/cout - объекты стандартного потока ввода вывода;
- if..else - условный оператор;
- new/delete - операторы выделения/освобождения динамической памяти.

Класс Cls:

- свойства/поля:
 - поле Массив чисел:
 - наименование — arr;
 - тип — int*;
 - модификатор доступа — private;
 - поле Длина массива:
 - наименование — length;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод Cls — Конструктор по умолчанию;
 - метод Cls — Параметризованный конструктор;
 - метод ~Cls — Деструктор;
 - метод Cls — Конструктор копирования;
 - метод input — Метод ввода данных в массив;
 - метод m1 — Метод сложения чисел в массиве;
 - метод m2 — Метод умножения чисел в массиве;
 - метод m3 — Метод возврата суммы чисел в массиве.

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 Cls

Функционал: Конструктор по умолчанию.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Cls

№	Предикат	Действия	№ перехода
1		Вывод Default constructor	Ø

3.2 Алгоритм деструктора класса Cls

Функционал: Деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса Cls

№	Предикат	Действия	№ перехода
1		Вывод Destructor	2
2		Освобождение памяти по указателю arg	Ø

3.3 Алгоритм конструктора класса Cls

Функционал: Параметризованный конструктор.

Параметры: int n - размер массива.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса Cls

№	Предикат	Действия	№ перехода
1		Вывод Constructor set	2
2		length = n	3
3		Выделение памяти по указателю arr размером n	∅

3.4 Алгоритм метода m1 класса Cls

Функционал: Метод сложения чисел в массиве.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода m1 класса Cls

№	Предикат	Действия	№ перехода
1		Инициализация i = 0 типа int	2
2	i < length	arr[i] = arr[i] + arr[i + 1]	3
		Возврат значения вызова метода m3	∅
3		i += 2	2

3.5 Алгоритм метода m2 класса Cls

Функционал: Метод умножения чисел в массиве.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода m2 класса Cls

№	Предикат	Действия	№ перехода
1		Инициализация i = 0 типа int	2
2	i < length	arr[i] = arr[i] * arr[i + 1]	3
		Возврат значения вызова метода m3	Ø
3		i += 2	2

3.6 Алгоритм метода m3 класса Cls

Функционал: Метод возврата суммы чисел в массиве.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода m3 класса Cls

№	Предикат	Действия	№ перехода
1		Инициализация sum = 0 типа int	2
2		Инициализация i = 0 типа int	3
3	i < length	sum += arr[i]	4
		Возврат sum	Ø
4		i++	3

3.7 Алгоритм метода input класса Cls

Функционал: Метод ввода данных в массив.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода *input* класса *Cls*

№	Предикат	Действия	№ перехода
1		Инициализация $i = 0$ типа <code>int</code>	2
2	$i < \text{length}$	Ввод значения <code>arr[i]</code>	3
			Ø
3		$i++$	2

3.8 Алгоритм конструктора класса *Cls*

Функционал: Конструктор копирования.

Параметры: `const Cls &ob`.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса *Cls*

№	Предикат	Действия	№ перехода
1		Вывод <code>Copy constructor</code>	2
2		<code>length = ob.length</code>	3
3		Выделение памяти по указателю <code>arr</code> размером <code>ob.length</code>	4
4		Инициализация $i = 0$ типа <code>int</code>	5
5	$i < \text{length}$	<code>arr[i] = ob.arr[i]</code>	6
			Ø
6		$i++$	5

3.9 Алгоритм функции *main*

Функционал: Основная функция программы.

Параметры: нет.

Возвращаемое значение: `int` - код ошибки.

Алгоритм функции представлен в таблице 9.

Таблица 9 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление <i>n</i> типа <i>int</i>	2
2		Ввод значения <i>n</i>	3
3	$n \% 2 \neq 0 \parallel n \leq 2$	Вывод <i>n</i> ?	10
			4
4		Вывод <i>n</i>	5
5		Создание объекта <i>obj</i> класса <i>Cls</i> с передачей аргумента <i>n</i>	6
6		Вызов метода <i>input()</i> объекта <i>obj</i>	7
7		Вызов функции <i>func</i> с передачей аргумента <i>obj</i>	8
8		Вызов метода <i>m1()</i> объекта <i>obj</i>	9
9		Вывод значения вызова метода <i>m3()</i> объекта <i>obj</i>	10
10		Возврат значения 0	Ø

3.10 Алгоритм функции *func*

Функционал: Вызов метода у *obj* для демонстрации копирования.

Параметры: *Cls obj*.

Возвращаемое значение: *void*.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции *func*

№	Предикат	Действия	№ перехода
1		Инициализация <i>s</i> типа <i>int</i> значением вызова метода <i>obj.m2()</i> ;	2
2		Вывод <i>s</i>	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

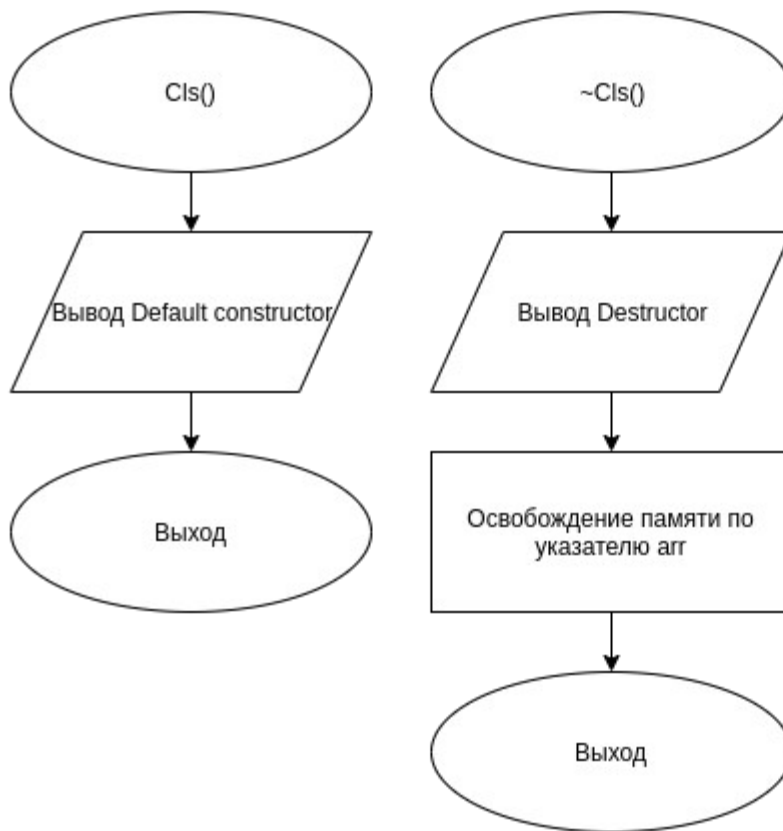


Рисунок 1 – Блок-схема алгоритма

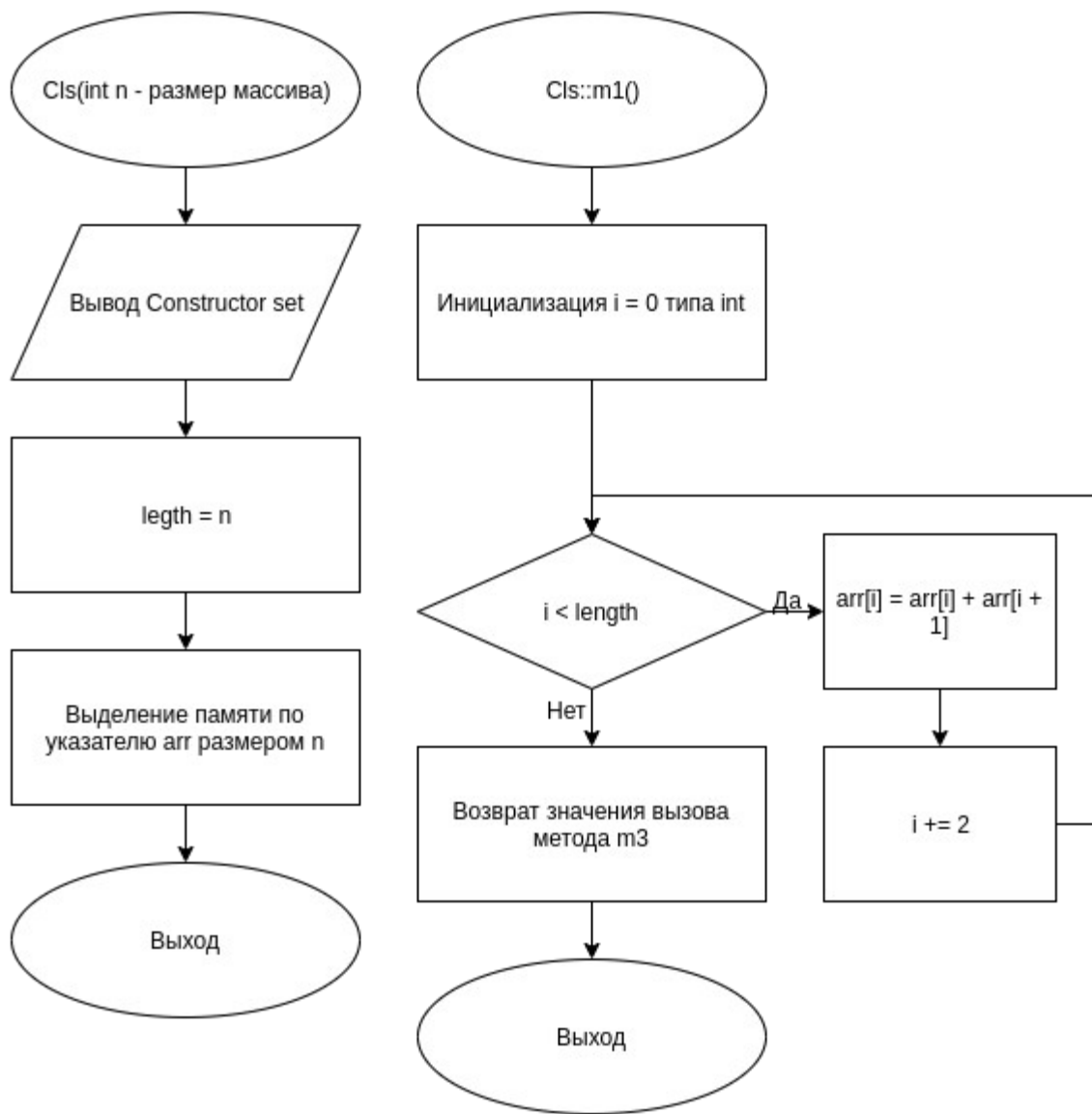


Рисунок 2 – Блок-схема алгоритма

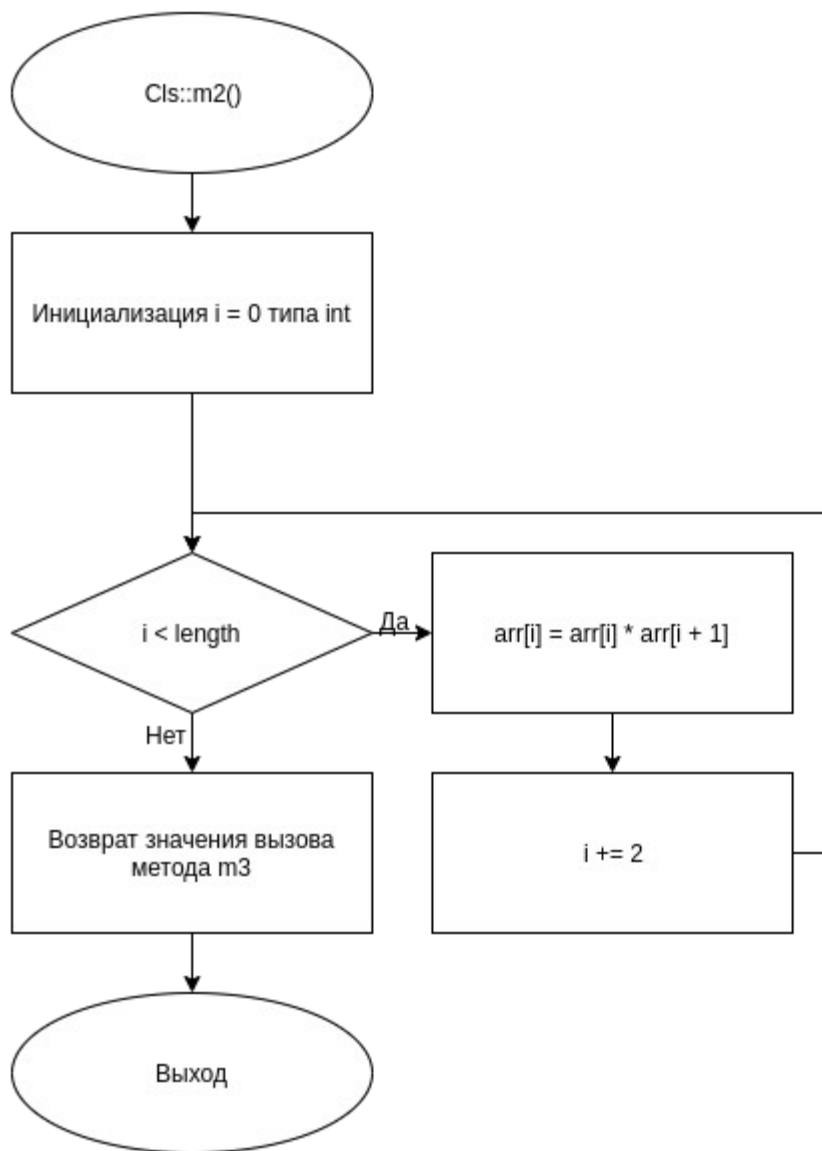


Рисунок 3 – Блок-схема алгоритма

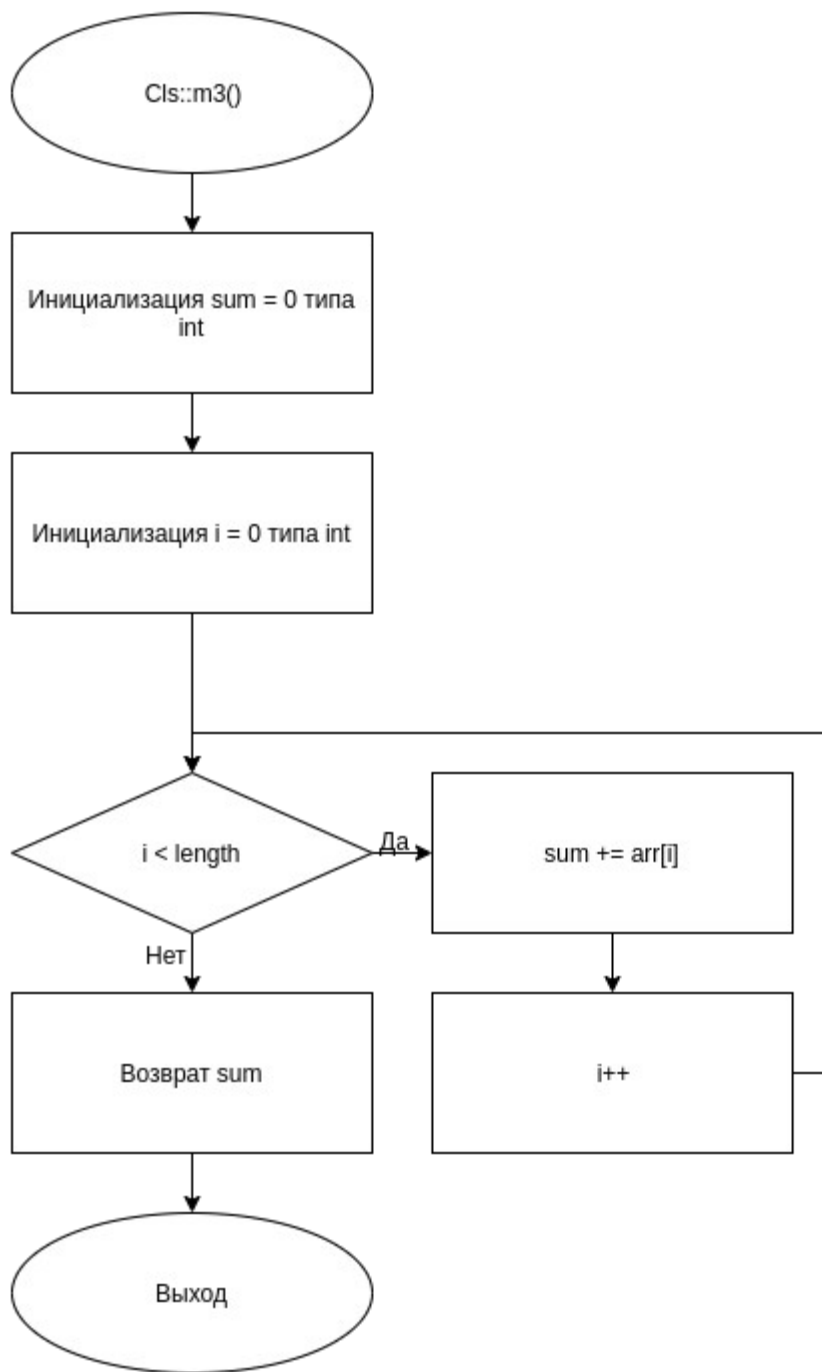


Рисунок 4 – Блок-схема алгоритма

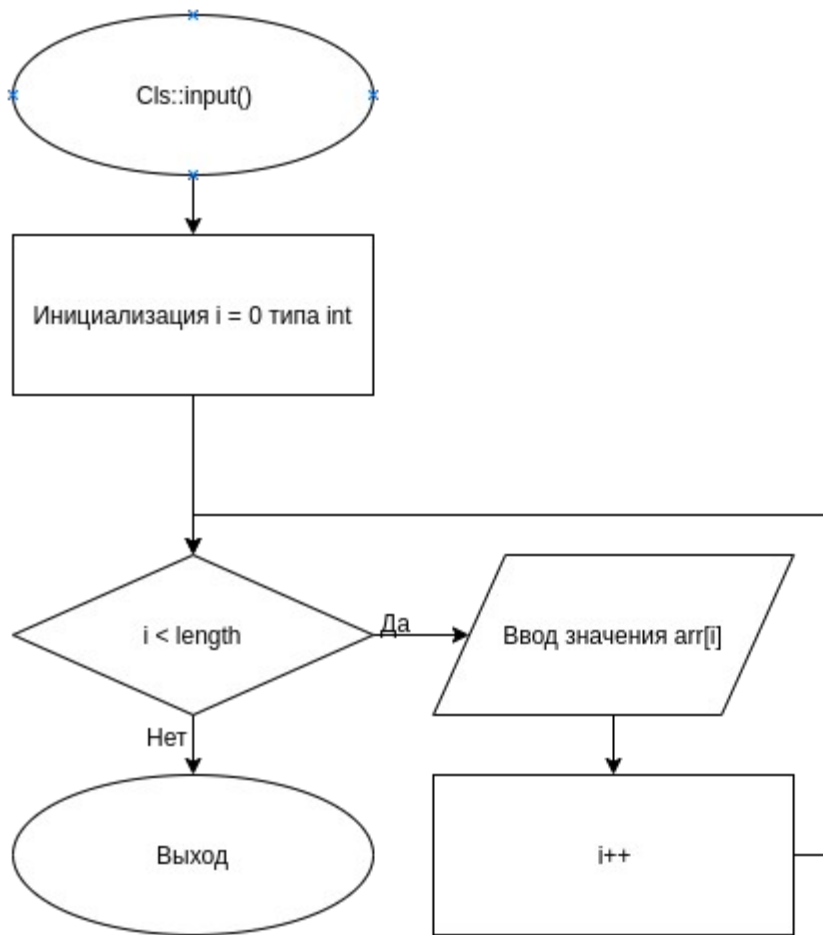


Рисунок 5 – Блок-схема алгоритма

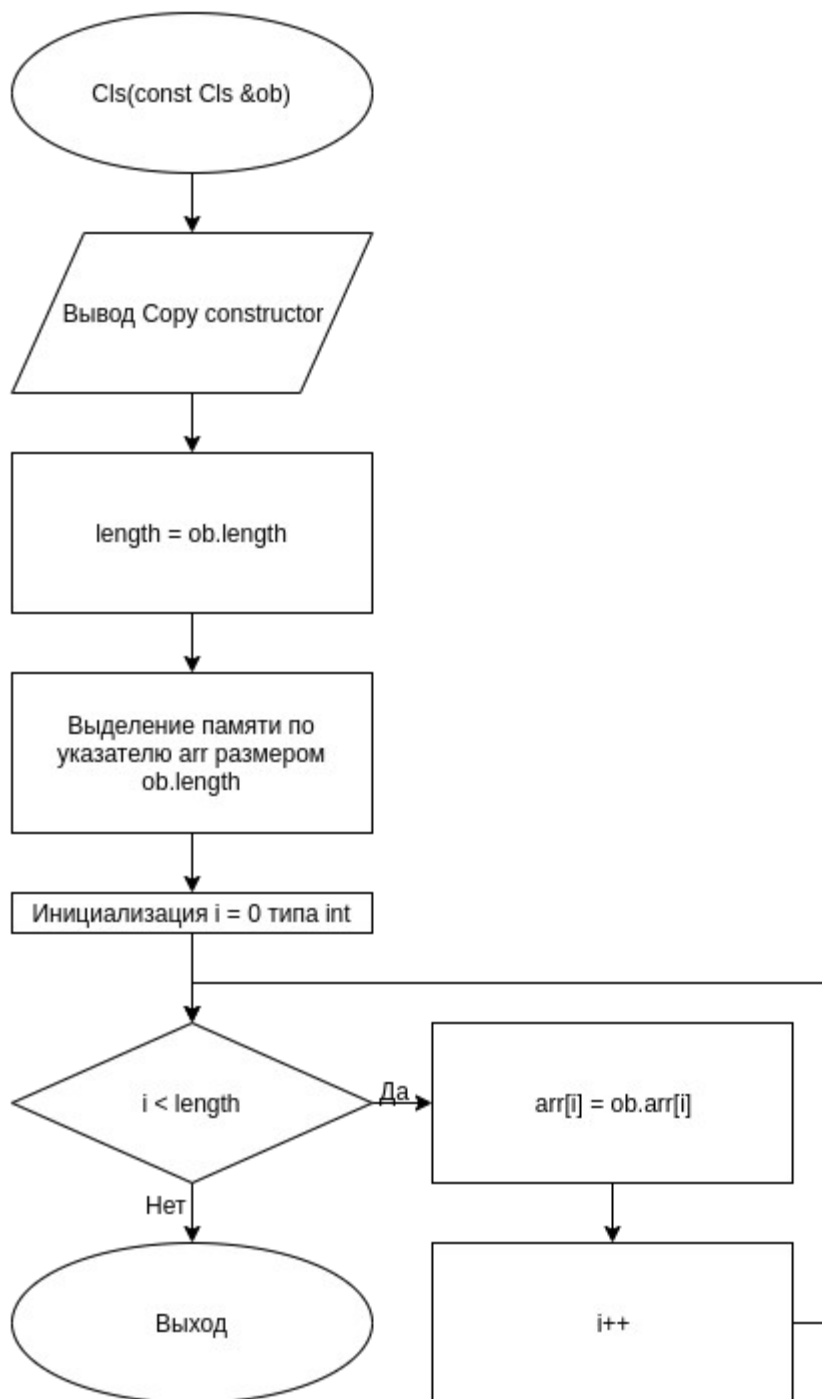


Рисунок 6 – Блок-схема алгоритма

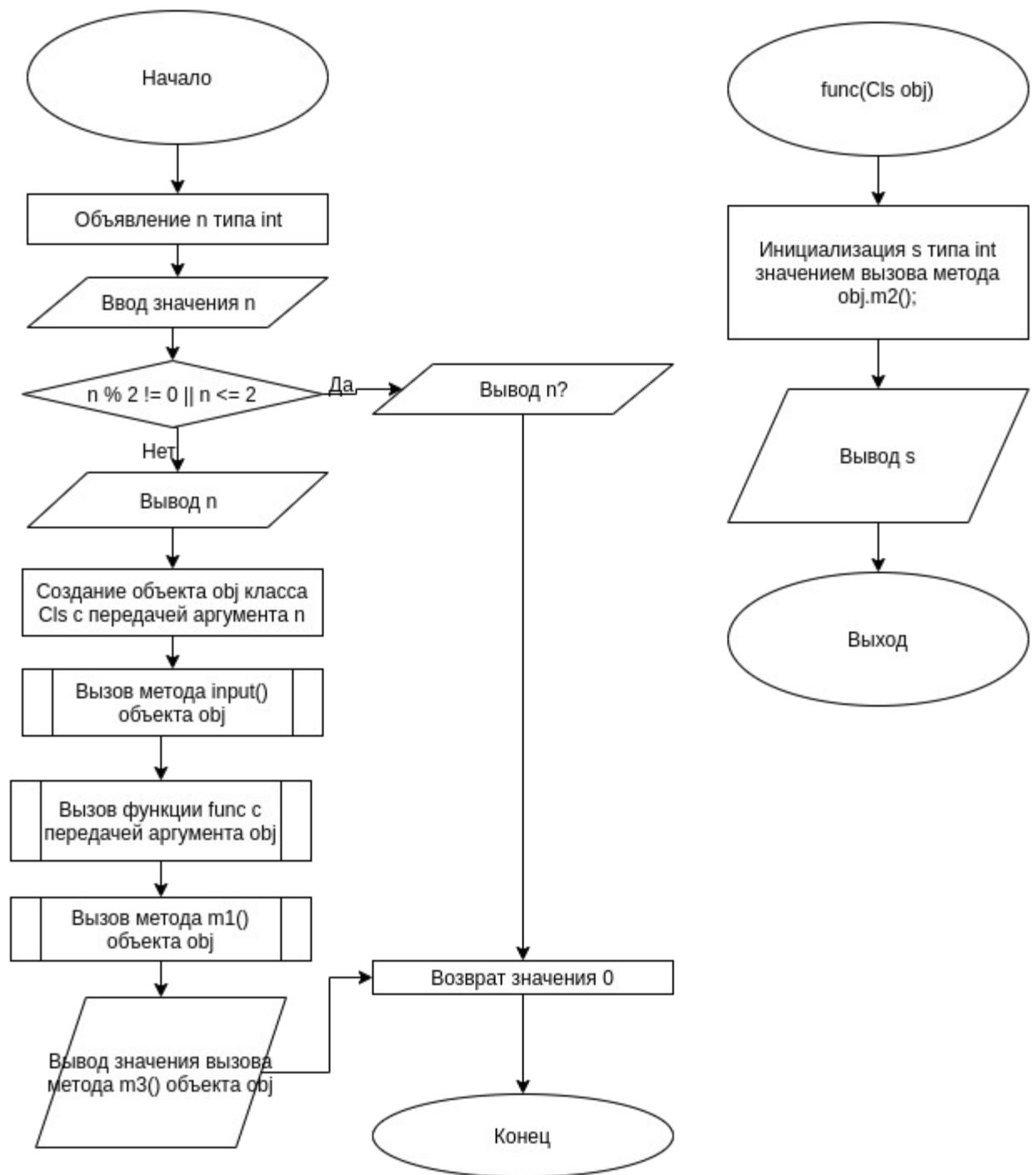


Рисунок 7 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Cls.cpp

Листинг 1 – Cls.cpp

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

using namespace std;

Cls::Cls()
{
    cout << endl << "Default constructor";
}
Cls::Cls(int n)
{
    cout << endl << "Constructor set";
    length = n;
    arr = new int[n];
}
Cls::~Cls()
{
    cout << endl << "Destructor";
    delete arr;
}
Cls::Cls(const Cls & ob)
{
    cout << endl << "Copy constructor";

    length = ob.length;
    arr = new int[ob.length];
    for(int i = 0; i < length; i++)
    {
        arr[i] = ob.arr[i];
    }
}
void Cls::input()
{
    for(int i = 0; i < length; i++)
    {
        cin >> arr[i];
    }
}
int Cls::m1()
```

```

{
    for(int i = 0; i < length; i+=2)
    {
        arr[i] = arr[i] + arr[i + 1];
    }
    return m3();
}
int Cls::m2()
{
    for(int i = 0; i < length; i+=2)
    {
        arr[i] = arr[i] * arr[i + 1];
    }
    return m3();
}
int Cls::m3()
{
    int sum = 0;
    for(int i = 0; i < length; i++)
    {
        sum += arr[i];
    }
    return sum;
}

```

5.2 Файл Cls.h

Листинг 2 – Cls.h

```

#ifndef __CLS__H
#define __CLS__H

class Cls
{
private:
    int* arr;
    int length;
public:
    Cls();
    Cls(int n);
    ~Cls();
    Cls(const Cls &ob);
    void input();
    int m1();
    int m2();
    int m3();
};

#endif

```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "Cls.h"
#include <iostream>

using namespace std;

void func(Cls obj){
    int s = obj.m2();
    cout << endl << s;
}

int main()
{
    int n;
    cin >> n;
    if(n % 2 != 0 || n <= 2){
        cout << n << "?";
        return(0);
    }

    cout << n;
    Cls obj(n);
    obj.input();
    func(obj);
    obj.m1();
    cout << endl << obj.m3();

    // program here
    return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 11.

Таблица 11 – 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
7	7?	7?

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).