Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

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;

- параметризированный конструктор, передается целочисленный параметр. Параметр должен иметь значение больше 2 и быть четным. По значению параметра определяется размерность целочисленного массива из закрытой области. Вначале 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. Вывод значения размерности массива.

3. Если размерность массива некорректная, вывод сообщения и завершить работу алгоритма.

4. Вывод значения размерности массива.

5. Создание объекта.

6. Вызов метода для ввода значений элементов массива.

7. Вызов функции передача в качестве аргумента объекта.

8. Вызов метода 1 от имени объекта.

9. 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 по умолчанию 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

Операторы ввода/вывода cin/cout

Операторы new/delete

Условные операторs if/else

Конструкторы

Деструктор

Методы Input, Met_a, Met_b, Summ

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 MyClass

Функционал: Конструктор по умолчанию.

Параметры: отсутствуют.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса MyClass

№	Предикат	Действия	№ перехода
1		Вывод "Default constructor"	∅

3.2 Алгоритм конструктора класса MyClass

Функционал: Конструктор с.

Параметры: int l, длинна массива.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса MyClass

№	Предикат	Действия	№ перехода
1		Вывод "Constructor set"	2
2		len = l	3
3		Создание динамического массива с длинной len	∅

3.3 Алгоритм конструктора класса MyClass

Функционал: Конструктор для копирования.

Параметры: const MyClass& obj, .

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса MyClass

№	Предикат	Действия	№ перехода
1		Вывод "Copy constructor"	2
2		len = obj.len	3
3		Создание динамического массива с длинной len	4
4		Копирование значений из изначального динамического массива	∅

3.4 Алгоритм деструктора класса MyClass

Функционал: Деструктор по умолчанию.

Параметры: отсутствуют.

Алгоритм деструктора представлен в таблице 4.

Таблица 4 – Алгоритм деструктора класса MyClass

№	Предикат	Действия	№ перехода
1		Вывод	2
2	Динамический массив существует?	Отчистить память выделенную под него	∅
			∅

3.5 Алгоритм метода Input класса MyClass

Функционал: Ввод значений массива.

Параметры: отсутствуют.

Возвращаемое значение: целый тип.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода Input класса MyClass

№	Предикат	Действия	№ перехода
1		Ввод значений массива	∅

3.6 Алгоритм метода Met_a класса MyClass

Функционал: Сложение каждого второго элемента с следующим.

Параметры: отсутствуют.

Возвращаемое значение: целый тип.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода Met_a класса MyClass

№	Предикат	Действия	№ перехода
1		Сложение каждого второго элемента с следующим	∅

3.7 Алгоритм метода Met_b класса MyClass

Функционал: Умножение каждого второго элемента с следующим.

Параметры: отсутствуют.

Возвращаемое значение: целый тип.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода Met_b класса MyClass

№	Предикат	Действия	№ перехода
1		Умножение каждого второго элемента с следующим	∅

3.8 Алгоритм метода Summ класса MyClass

Функционал: Вывод суммы всех элементов массива.

Параметры: отсутствуют.

Возвращаемое значение: целый тип.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода Summ класса MyClass

№	Предикат	Действия	№ перехода
1		Ввод суммы всех элементов массива	∅

3.9 Алгоритм функции func

Функционал: Создание объектра, вызов методов.

Параметры: MyClass& obj, копируймый объект.

Возвращаемое значение: целый тип.

Алгоритм функции представлен в таблице 9.

Таблица 9 – Алгоритм функции func

№	Предикат	Действия	№ перехода
1		Создание объекта с ссылкой на объект поданной на параметр	2
2		Вызов метода Met_b()	3
3		Вывод значения метода Summ()	∅

3.10 Алгоритм функции main

Функционал: основная функция.

Параметры: отсутствуют.

Возвращаемое значение: целый тип.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Интилизация, ввод, вывод l	2
2	l > 2 && l % 2 == 0		3
		Вывод "?"	∅
3		Создание объекта с l поданным на параметр	4
4		Вызов метода Input()	5
5		Вызов функции func(obj_a)	6
6		Вызов метода Met_a()	7
7		Вывод значения метода Summ()	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-4.

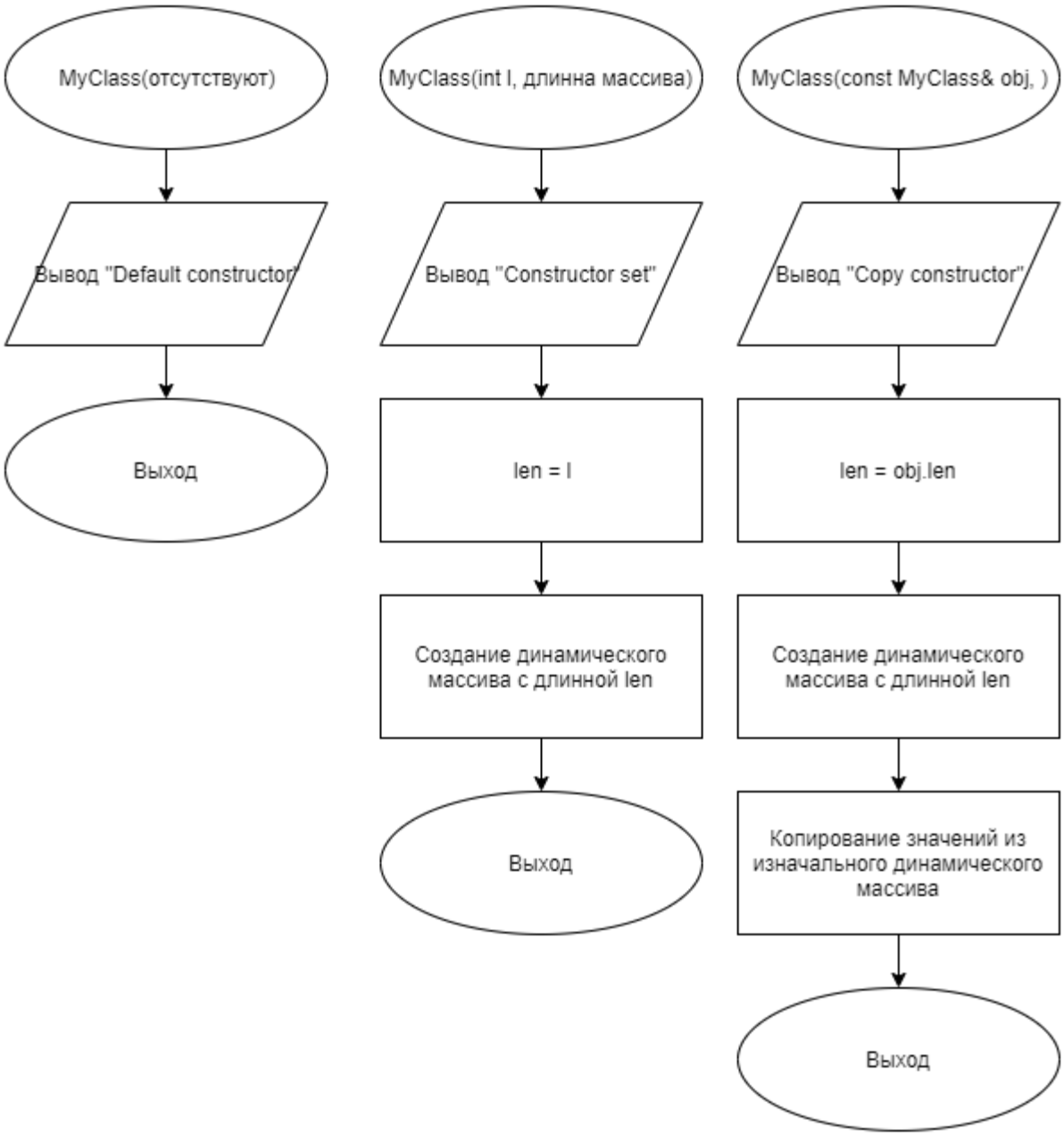

Рисунок 1 – Блок-схема алгоритма

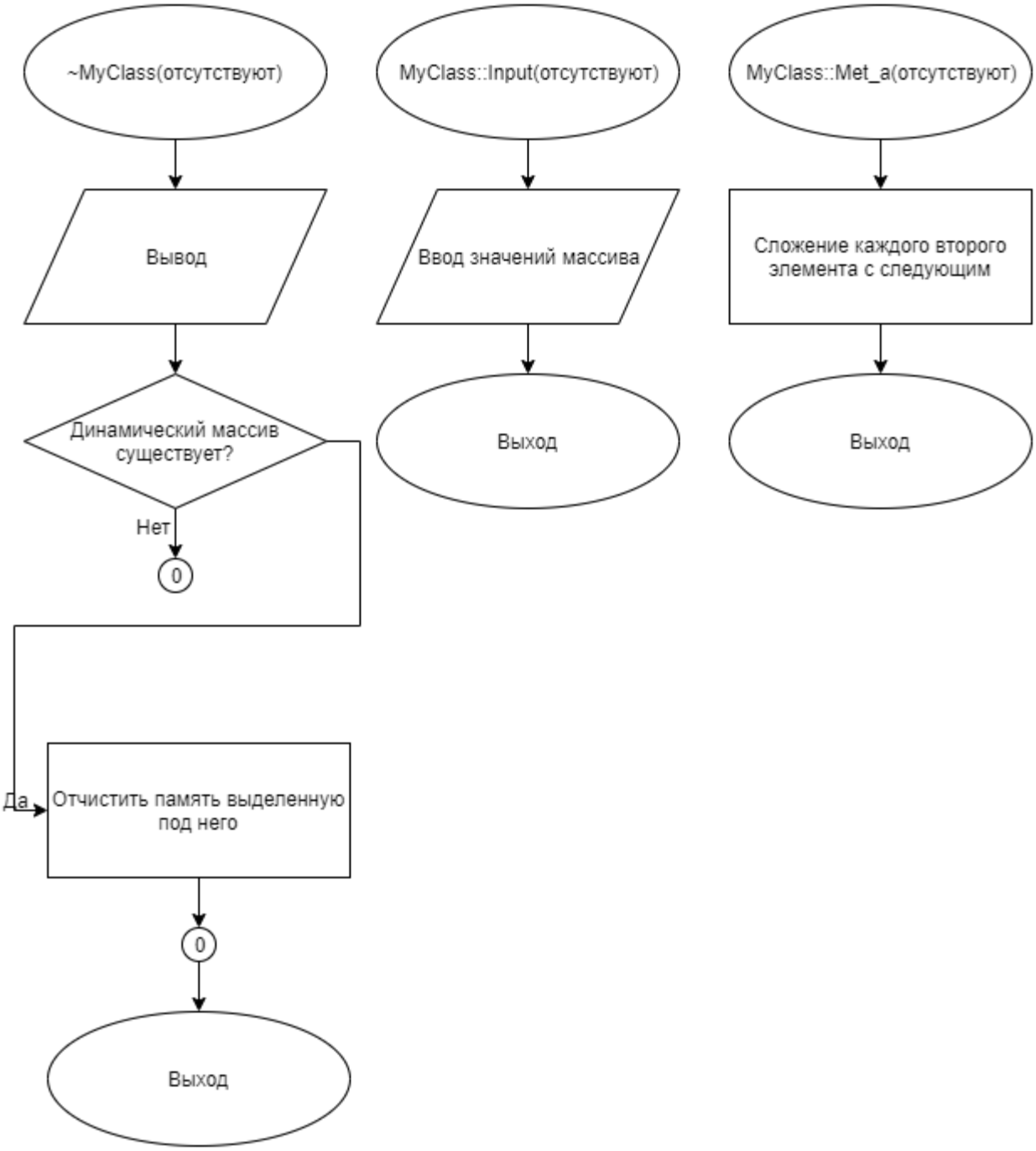

Рисунок 2 – Блок-схема алгоритма

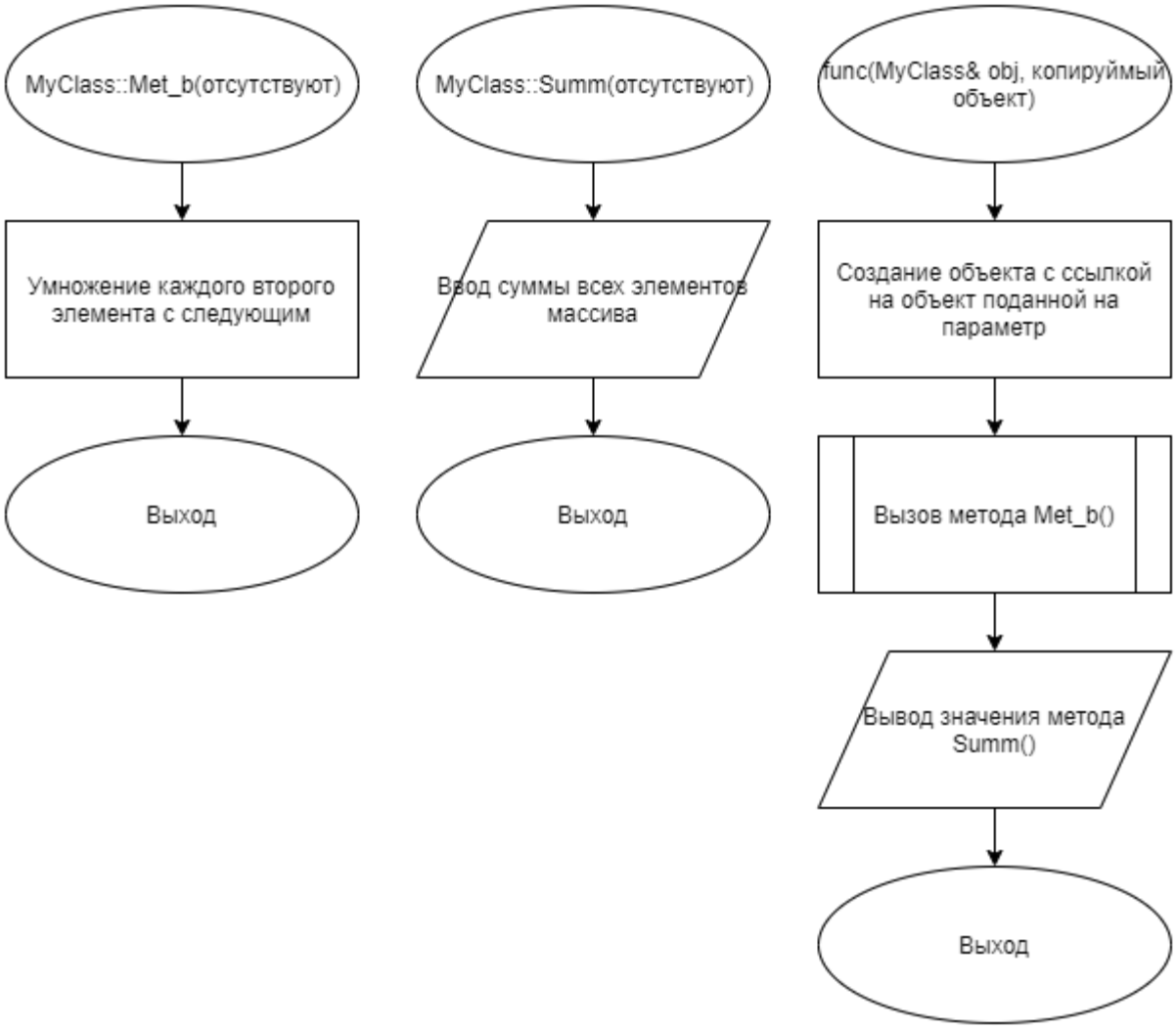

Рисунок 3 – Блок-схема алгоритма

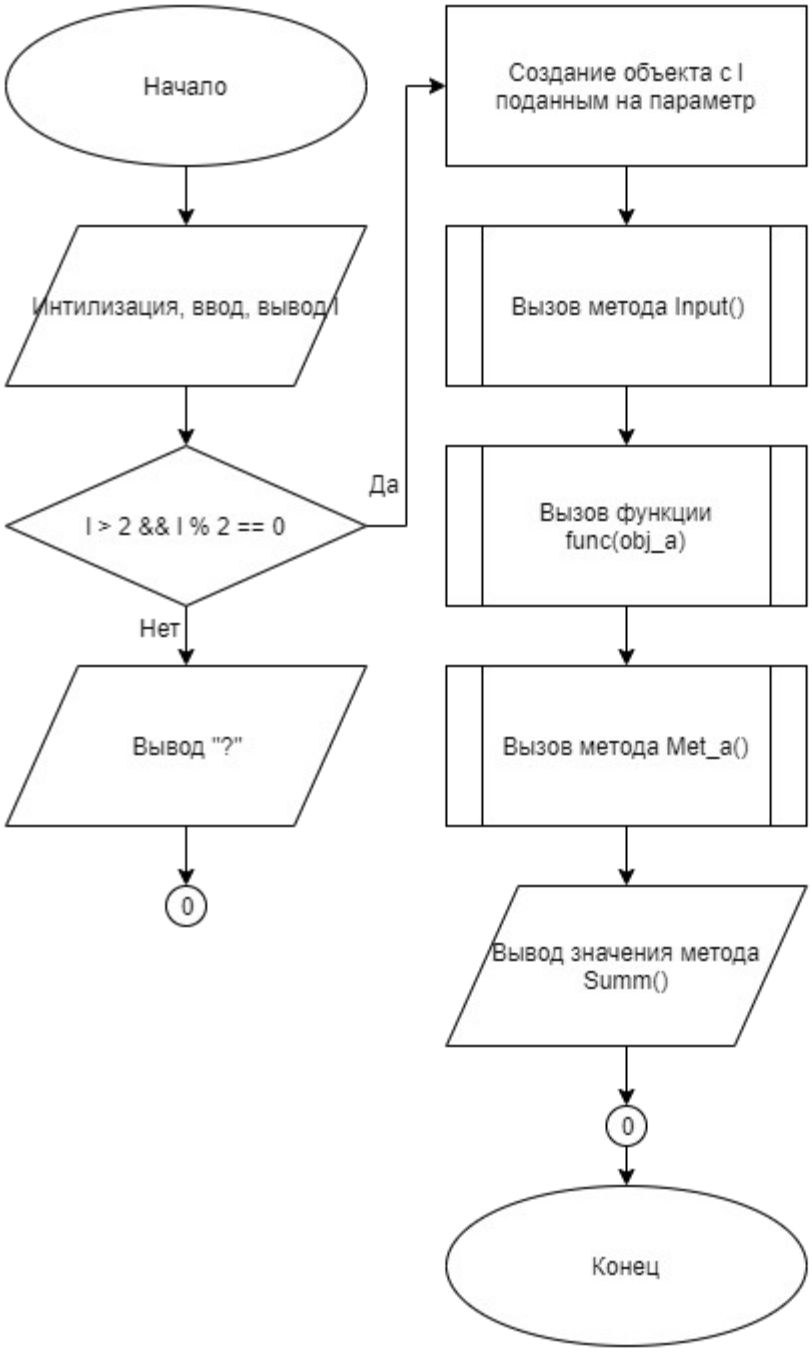

Рисунок 4 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include "MyClass.h"

void func(MyClass& obj)
{
	MyClass obj_func(obj);
	obj_func.Met_b();
	cout << endl << obj_func.Summ();
}

int main()
{
	int l;
	cin >> l;
	cout << l;
	if ((l > 2) && (l % 2 == 0))
	{
		MyClass obj_a(l);
		obj_a.Input();
		func(obj_a);
		obj_a.Met_a();
		cout << endl << obj_a.Summ();
	}
	else
	{
		cout << "?";
	}
}
```

5.2 Файл MyClass.cpp

Листинг 2 – MyClass.cpp

```
#include "MyClass.h"


MyClass::MyClass()
{
```

```
	cout << "\nDefault constructor";
}

MyClass::MyClass(int l)
{
	cout << "\nConstructor set";
	len = l;
	mas = new int[len];
}

MyClass::~MyClass()
{
	cout << "\nDestructor";
	if (mas != nullptr)
	{
		delete []mas;
	}
}

MyClass::MyClass(MyClass& obj)
{
	cout << "\nCopy constructor";
	len = obj.len;
	mas = new int[len];
	for (int i = 0; i < len; i++)
	{
		mas[i] = obj.mas[i];
	}
}

void MyClass::Input()
{
	for (int i = 0; i < len; i++)
	{
		cin >> mas[i];
	}
}

void MyClass::Met_a()
{
	for (int i = 0; i+1 < len; i = i + 2)
	{
		mas[i] = mas[i] + mas[i+1];
	}
}

void MyClass::Met_b()
{
	for (int i = 0; i+1 < len; i = i + 2)
	{
		mas[i] = mas[i] * mas[i+1];
	}
}

int MyClass::Summ()
{
	int summ = 0;
```

```
	for (int i = 0; i < len; i++)
	{
		summ = summ + mas[i];
	}
	return summ;
}
```

5.3 Файл MyClass.h

Листинг 3 – MyClass.h

```
#ifndef __MYCLASS_H__
#define __MYCLASS_H__
#include <iostream>
using namespace std;
class MyClass
{
public:
	MyClass();
	MyClass(int l);
	MyClass(MyClass& obj);
	~MyClass();
	void Input();
	void Met_a();
	void Met_b();
	int Summ();
private:
	int *mas, len = 0;
};
#endif
```

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
2	2?	2?
7	7?	7?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.

2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.

3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodichescoe_posobie_dlya_laboratorny h_rabot_3.pdf (дата обращения 05.05.2021).

4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).

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).