

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 Test.....	10
3.2 Алгоритм деструктора класса Test.....	10
3.3 Алгоритм метода input класса Test.....	10
3.4 Алгоритм метода sum класса Test.....	11
3.5 Алгоритм метода metod1 класса Test.....	12
3.6 Алгоритм метода metod2 класса Test.....	12
3.7 Алгоритм конструктора класса Test.....	13
3.8 Алгоритм конструктора класса Test.....	13
3.9 Алгоритм функции main.....	14
3.10 Алгоритм функции func.....	15
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	16
5 КОД ПРОГРАММЫ.....	21
5.1 Файл main.cpp.....	21
5.2 Файл Test.cpp.....	21
5.3 Файл Test.h.....	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 Test предназначен для ;
- функция fun для Функция вызывается metod2, далее выводит сумму элементов массива с новой строки.;
- Объект стандартного потока ввода с клавиатуры cin;
- Объект стандартного потока вывода на экран cout;
- Условный оператор if..else;
- Оператор цикла for.

Класс Test:

- свойства/поля:
  - поле Размер массива:
    - наименование — n;
    - тип — int;
    - модификатор доступа — private;
  - поле Указатель на массив:
    - наименование — mass;
    - тип — int\*;
    - модификатор доступа — private;
- функционал:
  - метод Test — Конструктор стандартный;
  - метод Test — Конструктор параметризованный;
  - метод Test — Конструктор копии;
  - метод ~Test — Деструктор;
  - метод input — Ввод массива;
  - метод sum — Подсчет суммы элементов массива;

- о метод `metod1` — Суммирует значения очередной пары элементов и сумму присваивает первому элементу пары. Далее суммирует элементы полученного массива и возвращает это значение;
- о метод `metod2` — Умножает значения очередной пары элементов и результат присваивает первому элементу пары. Далее суммирует элементы полученного массива и возвращает это значение..

## 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 Test

Функционал: Конструктор стандартный.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Test

№	Предикат	Действия	№ перехода
1		Вывод на экран "Default constructor"	Ø

### 3.2 Алгоритм деструктора класса Test

Функционал: Деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса Test

№	Предикат	Действия	№ перехода
1		Удаление из памяти массива по адресу mass	2
2		Вывод на экран "Destructor"	Ø

### 3.3 Алгоритм метода input класса Test

Функционал: Ввод массива.



Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода *input* класса *Test*

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной <i>i</i> и инициализация 0	2
2	$i < n$	Ввод с клавиатуры значения ячейки с адресом <i>mass[i]</i>	3
			Ø
3		Увеличение <i>i</i> на 1	2

### 3.4 Алгоритм метода *sum* класса *Test*

Функционал: Подсчет суммы элементов массива.

Параметры: нет.

Возвращаемое значение: Целочисленной значение суммы элементов массива.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода *sum* класса *Test*

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной <i>summ</i> и инициализация 0 Объявление целочисленной переменной <i>i</i> и инициализация 0	2
2	$i < n$	Переменная <i>summ</i> увеличивается на значение ячейки <i>mass[i]</i>	2
		Вернуть значение <i>summ</i>	Ø

### 3.5 Алгоритм метода *metod1* класса *Test*

Функционал: Суммирует значения очередной пары элементов и сумму присваивает первому элементу пары. Далее суммирует элементы полученного массива и возвращает это значение.

Параметры: нет.

Возвращаемое значение: Целочисленное значение суммы элементов массива.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *metod1* класса *Test*

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной <i>i</i> и инициализация 0	2
2	$i < n$	Значение ячейки с адресом <i>mass[i]</i> увеличивается на значение ячейки <i>mass[i+1]</i> Увеличение <i>i</i> на 1	2
		Вернуть результат метода <i>sum</i>	∅

### 3.6 Алгоритм метода *metod2* класса *Test*

Функционал: Умножает значения очередной пары элементов и результат присваивает первому элементу пары. Далее суммирует элементы полученного массива и возвращает это значение..

Параметры: нет.

Возвращаемое значение: Целочисленное значение суммы элементов массива.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода *metod2* класса *Test*

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной <i>i</i> и инициализация 0	2
2	$i < n$	Значение ячейки с адресом <i>mass[i]</i> умножается на 2 значение ячейки <i>mass[i+1]</i> Увеличение <i>i</i> на 1	2
		Вернуть результат метода <i>sum</i>	∅

### 3.7 Алгоритм конструктора класса *Test*

Функционал: Параметризированный конструктор.

Параметры: Целочисленная переменная *x* для ввода размера массива.

Алгоритм конструктора представлен в таблице 7.

Таблица 7 – Алгоритм конструктора класса *Test*

№	Предикат	Действия	№ перехода
1		Вывод на экран "Constructor set"	2
2		Закрытой переменной <i>n</i> присваивается значение параметра <i>x</i>	3
3		Указателю <i>mass</i> присваивается адрес массива размерностью <i>n</i>	∅

### 3.8 Алгоритм конструктора класса *Test*

Функционал: Конструктор копии.

Параметры: Объект класса *Test*.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса *Test*

№	Предикат	Действия	№ перехода
1		Вывод на экран "Copy constructor"	2

№	Предикат	Действия	№ перехода
2		Закрытой переменной $p$ присваивается значение переменной $p$ исходного объекта	3
3		Указателю $mass$ присваивается значение массива с размерностью $p$ Объявление целочисленной переменной $i$ и инициализация 0	4
4	$i < p$	Ячейке с адресом $mass[i]$ присваивается значение ячейки с адресом $mass[i]$ исходного объекта Увеличение $i$ на 1	4
			Ø

### 3.9 Алгоритм функции `main`

Функционал: Выполнение действий, описанных в задаче.

Параметры: нет.

Возвращаемое значение: Целочисленное значение.

Алгоритм функции представлен в таблице 9.

Таблица 9 – Алгоритм функции `main`

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной $x$	2
2		Ввод с клавиатуры значения $x$	3
3	$x > 2$ и $x \% 2 = 0$	Вывод "(x)"	4
		Вывод "(x)?"	Ø
4		Объявление объекта $obj$ класса <code>Test</code> с параметром $x$	5
5		Вызов метода <code>input</code> объекта $obj$ класса <code>Test</code>	6
6		Вызов функции <code>func</code> с параметром $obj$	7
7		Вызов метода <code>metod1</code> объекта $obj$ класса <code>Test</code>	8

№	Предикат	Действия	№ перехода
8		Вывод результата на экран	Ø

### 3.10 Алгоритм функции func

Функционал: Работа с копией объекта класса Test.

Параметры: Объект ob\_local класса Test для создания копии.

Возвращаемое значение: Ничего.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции func

№	Предикат	Действия	№ перехода
1		Вызов metod2 объекта ob_local класса Test	Ø

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

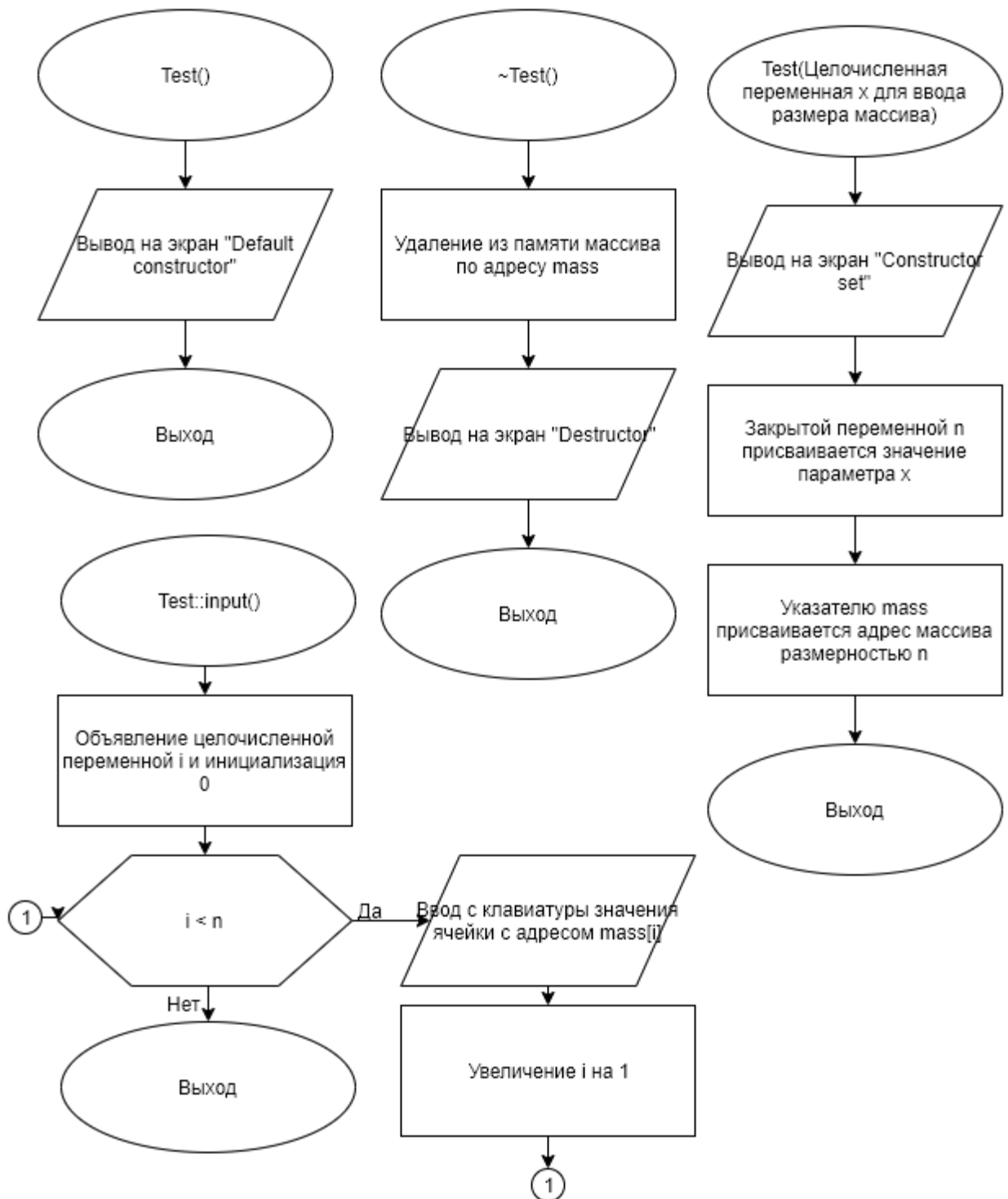


Рисунок 1 – Блок-схема алгоритма

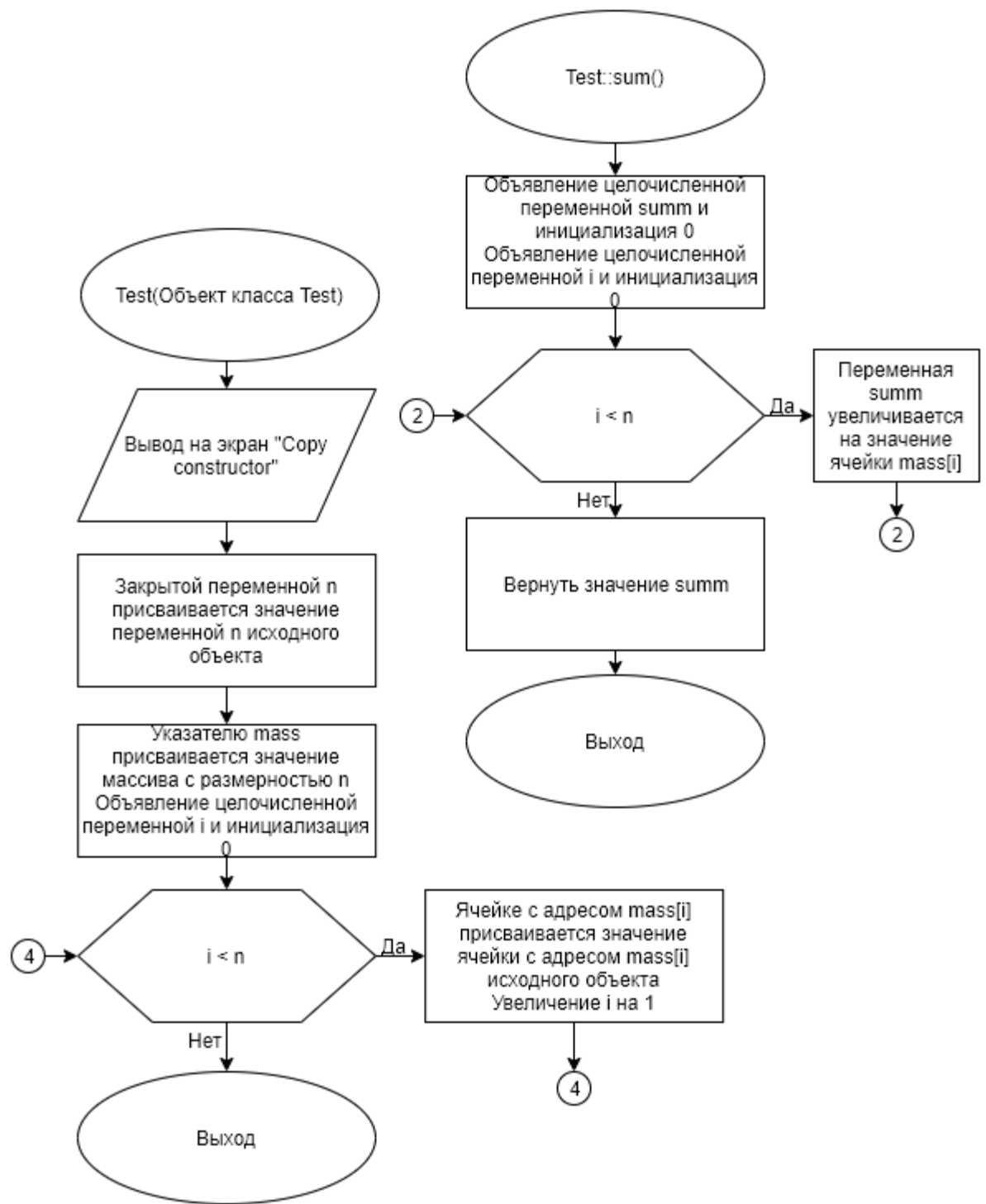
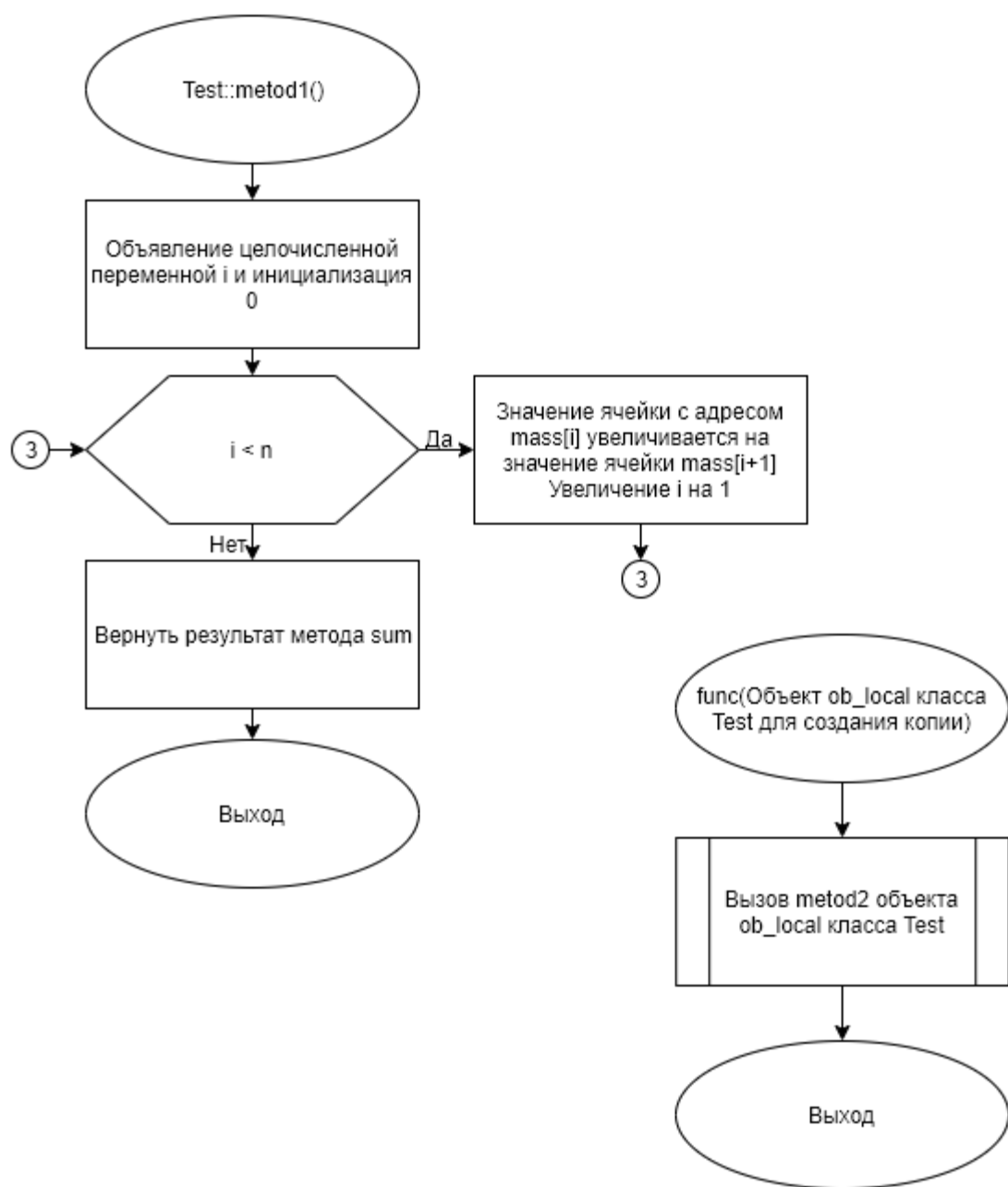


Рисунок 2 – Блок-схема алгоритма



**Рисунок 3 – Блок-схема алгоритма**



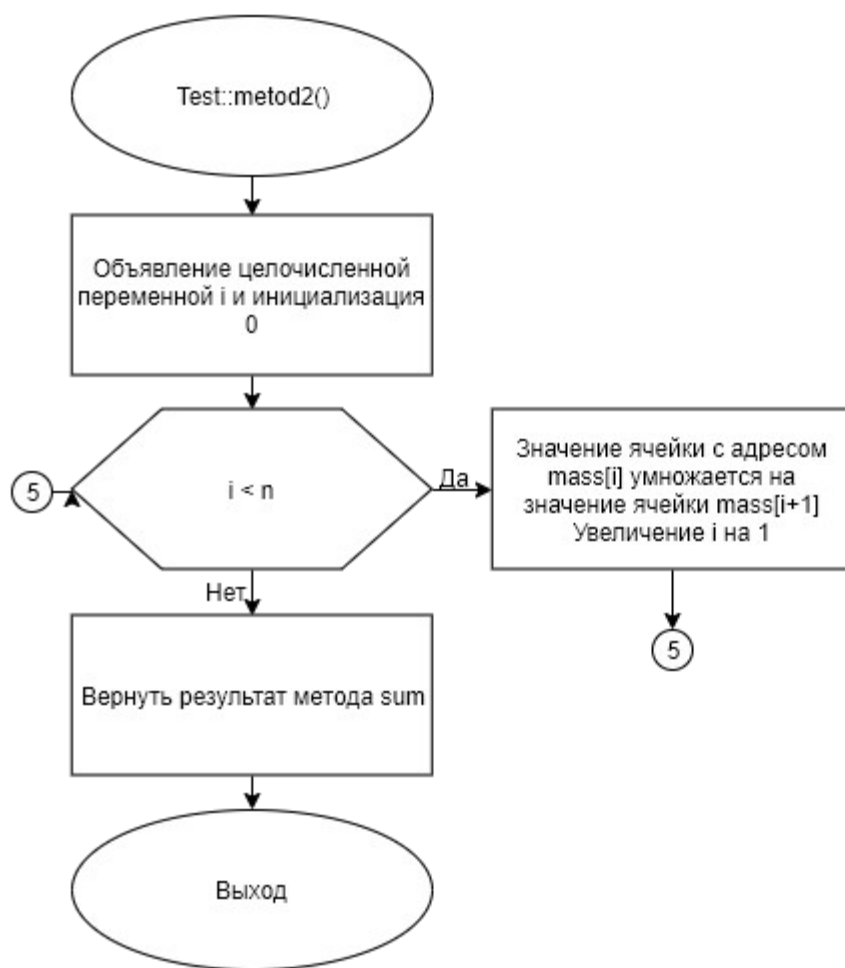


Рисунок 4 – Блок-схема алгоритма

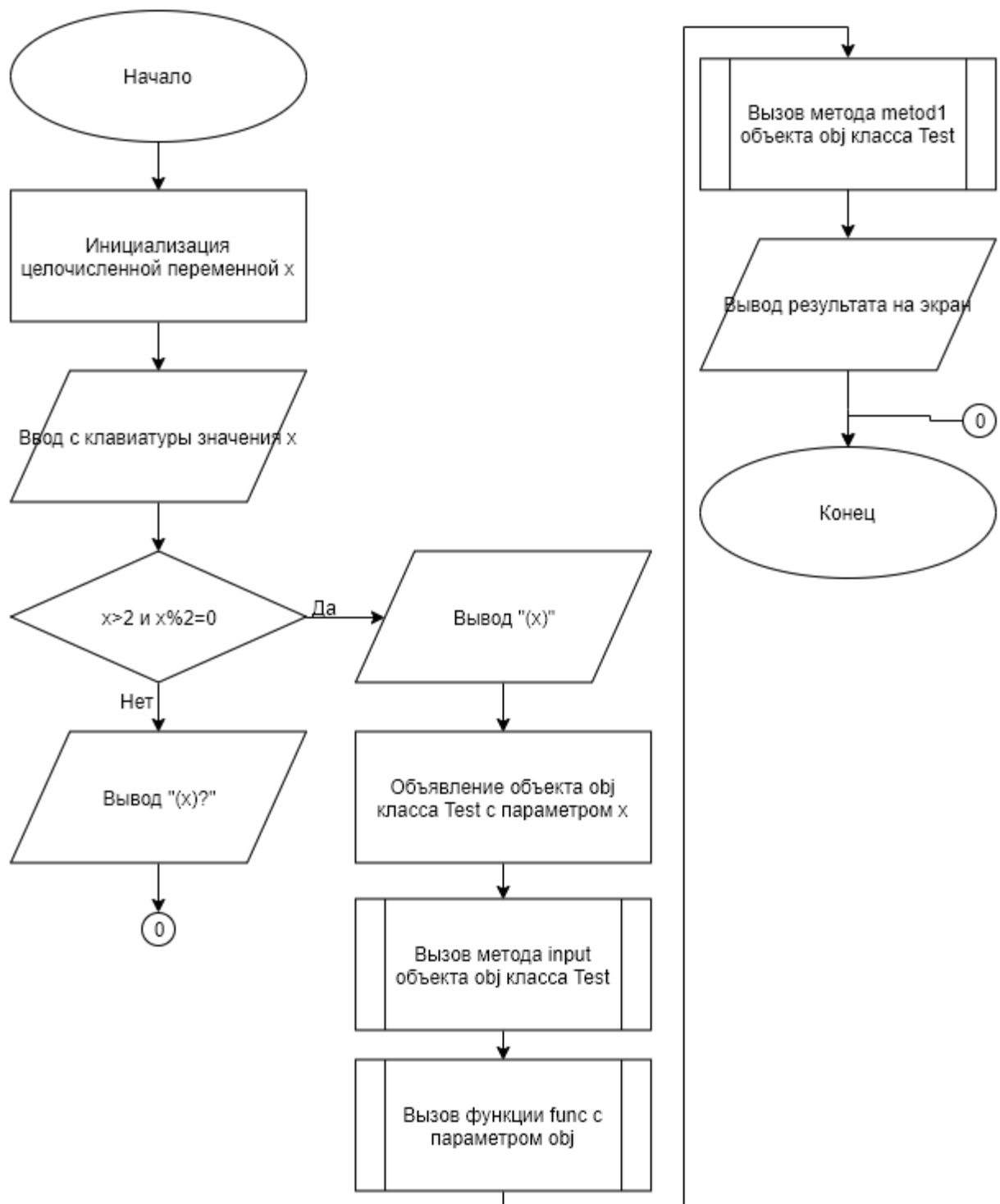


Рисунок 5 – Блок-схема алгоритма

## 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 "Test.h"

using namespace std;

void func(Test ob_local)
{
    cout << ob_local.metod2() << endl;
}

int main()
{
    int x;
    cin >> x;
    if (x > 2 && x % 2 == 0) {
        cout << x << endl;
        Test obj(x);
        obj.input();
        func(obj);
        cout << obj.metod1() << endl;
    }
    else {
        cout << x << "?";
        exit(0);
    }
}
```

### 5.2 Файл Test.cpp

*Листинг 2 – Test.cpp*

```
#include "Test.h"
```

```

using namespace std;

Test::Test()
{
    cout << "Default constructor" << endl;
}

Test::Test(int x)
{
    cout << "Constructor set" << endl;
    n = x;
    mass = new int[n];
}

Test::Test(const Test & ob)
{
    cout << "Copy constructor" << endl;
    n = ob.n;
    mass = new int[n];
    for(int i = 0; i < n; i++) {
        mass[i] = ob.mass[i];
    }
}

void Test::input()
{
    for(int i = 0; i < n; i++) {
        cin >> mass[i];
    }
}

int Test::sum()
{
    int summ = 0;
    for (int i = 0; i < n; i++) {
        summ += mass[i];
    }
    return(summ);
}

int Test::metod1()
{
    for (int i = 0; i < n; i += 2) {
        mass[i] += mass[i + 1];
    }
    return sum();
}

int Test::metod2()
{
    for (int i = 0; i < n; i += 2) {
        mass[i] *= mass[i + 1];
    }
    return sum();
}

```

```
}

Test::~~Test()
{
    delete[] mass;
    cout << "Destructor" << endl;
}
```

## 5.3 Файл Test.h

*Листинг 3 – Test.h*

```
#ifndef __TEST__H
#define __TEST__H
#include <iostream>

using namespace std;

class Test
{
private:
    int n;
    int* mass;

public:
    Test();
    Test(int x);
    Test(const Test & ob);
    void input();
    int sum();
    int metod1();
    int metod2();
    ~Test();

};

#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
5 5 6 7 8 9	5?	5?
2 1 2	2?	2?
6 4 5 6 7 8 9	6 Constructor set Copy constructor 155 Destructor 60 Destructor	6 Constructor set Copy constructor 155 Destructor 60 Destructor
8 8 8 8 8 8 8 8 8	8 Constructor set Copy constructor 288 Destructor 96 Destructor	8 Constructor set Copy constructor 288 Destructor 96 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](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](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).