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. 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 параметризированного 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
```

Параметризированный 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 obj1 класса Test предназначен для ;
- объект obj2 класса Test предназначен для ;
- объект local класса Test предназначен для ;
- функция fun для работы с локальным объектом;
- 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 параметризированный;
    - метод new_mass — Создание целочисленного массива в закрытой области;
    - метод output — Вывод содержимого элементов массива;
    - метод metod1 — Суммирует значения очередной пары элементов и

сумму присваивает первому элементу пары.;

- метод metod2 — Умножает значения очередной пары элементов и результат присваивает 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 метода new_mass класса Test

Функционал: Создание целочисленного массива в закрытой области.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода new_mass класса Test

№	Предикат	Действия	№ перехода
1		Указателю mass присваивается адрес массива размерностью n	∅

3.2 Алгоритм метода metod1 класса Test

Функционал: Суммирует значения очередной пары элементов и сумму присваивает первому элементу пары..

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода metod1 класса Test

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной i инициализация 0	2
2	i<n	Значение ячейки с адресом mass[i] увеличивается	2

№	Предикат	Действия	№ перехода
		на значение ячейки mass[i+1] Увеличение i на 1	
			∅

3.3 Алгоритм метода metod2 класса Test

Функционал: Умножает значения очередной пары элементов и результат присваивает первому элементу пары..

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода metod2 класса Test

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

3.4 Алгоритм метода output класса Test

Функционал: Вывод содержимого элементов массива.

Параметры: нет.

Возвращаемое значение: Ничего.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода output класса Test

№	Предикат	Действия	№ перехода
1		Объявление целочисленой переменной i и инициализация 0	2
2	i<n-1	Вывод на экран значения ячейки с адресом mass[i] и трех пробелов	3
		Вывод на экран значения ячейки с адресом mass[n-1] и переноса строки	∅
3		Увеличение i на 1	2

3.5 Алгоритм конструктора класса Test

Функционал: Параметризированный конструктор.

Параметры: Целочисленная переменная x для ввода размера массива.

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса Test

№	Предикат	Действия	№ перехода
1		Вывод на экран "Constructor set"	2
2		Закрытой переменной n присваивается значение параметра x	∅

3.6 Алгоритм функции fun

Функционал: Создание локального объекта с использованием параметризированного конструктора.

Параметры: Целочисленный параметр x содержащий размерность массива.

Возвращаемое значение: Объект класса Test.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции fun

№	Предикат	Действия	№ перехода
1		Создание объекта local класса Test с параметром x	2
2		Возврат local	∅

3.7 Алгоритм функции main

Функционал: Выполнение действий, определенных в задаче.

Параметры: нет.

Возвращаемое значение: Целочисленное значение.

Алгоритм функции представлен в таблице 7.

Таблица 7 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленнной переменной x	2
2		Ввод с клавиатуры значения x	3
3	x>2 и x четно	Вывод на экран "x"	4
		Вывод на экран "(x)?"	∅
4		Создание объекта obj1 класса Test	5
5		Вызов функции fun с параметром x и инициализация объекта obj1 результатом	6
6		Вызов метода new_mass объекта obj1	7
7		Вызов метода input объекта obj1	8
8		Вызов метода metod2 объекта obj1	9
9		Инициализация obj2 класса Test объектом obj1	10
10		Вызов метода metod1 объекта obj2	11
11		Вызов метода output объекта obj1	12
12		Вызов метода sum объекта obj1 и вывод результата на экран	13
13		Вызов метода output объекта obj2	14

№	Предикат	Действия	№ перехода
14		Вызов метода sum объекта obj2 и вывод результата на экран	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

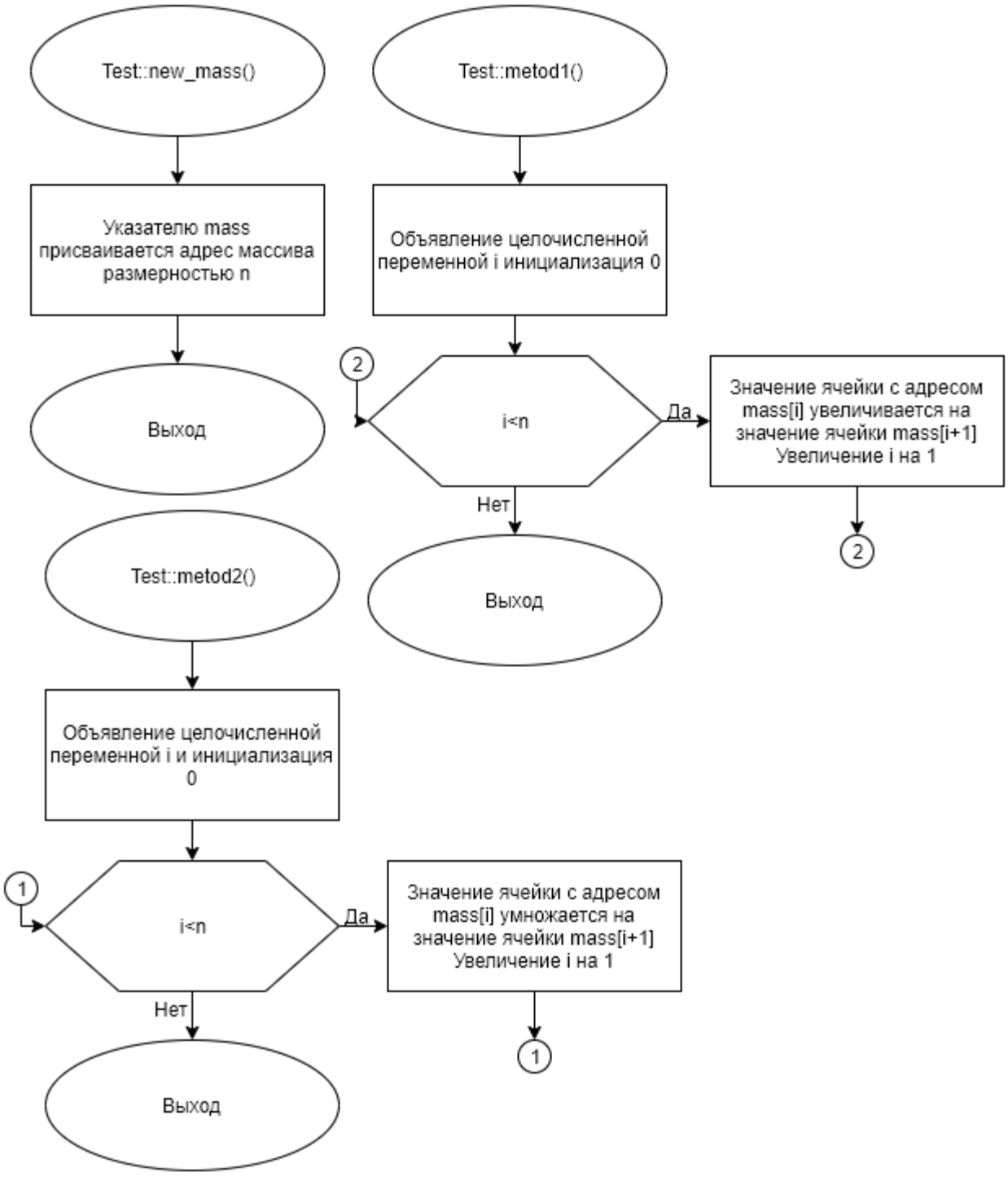

Рисунок 1 – Блок-схема алгоритма

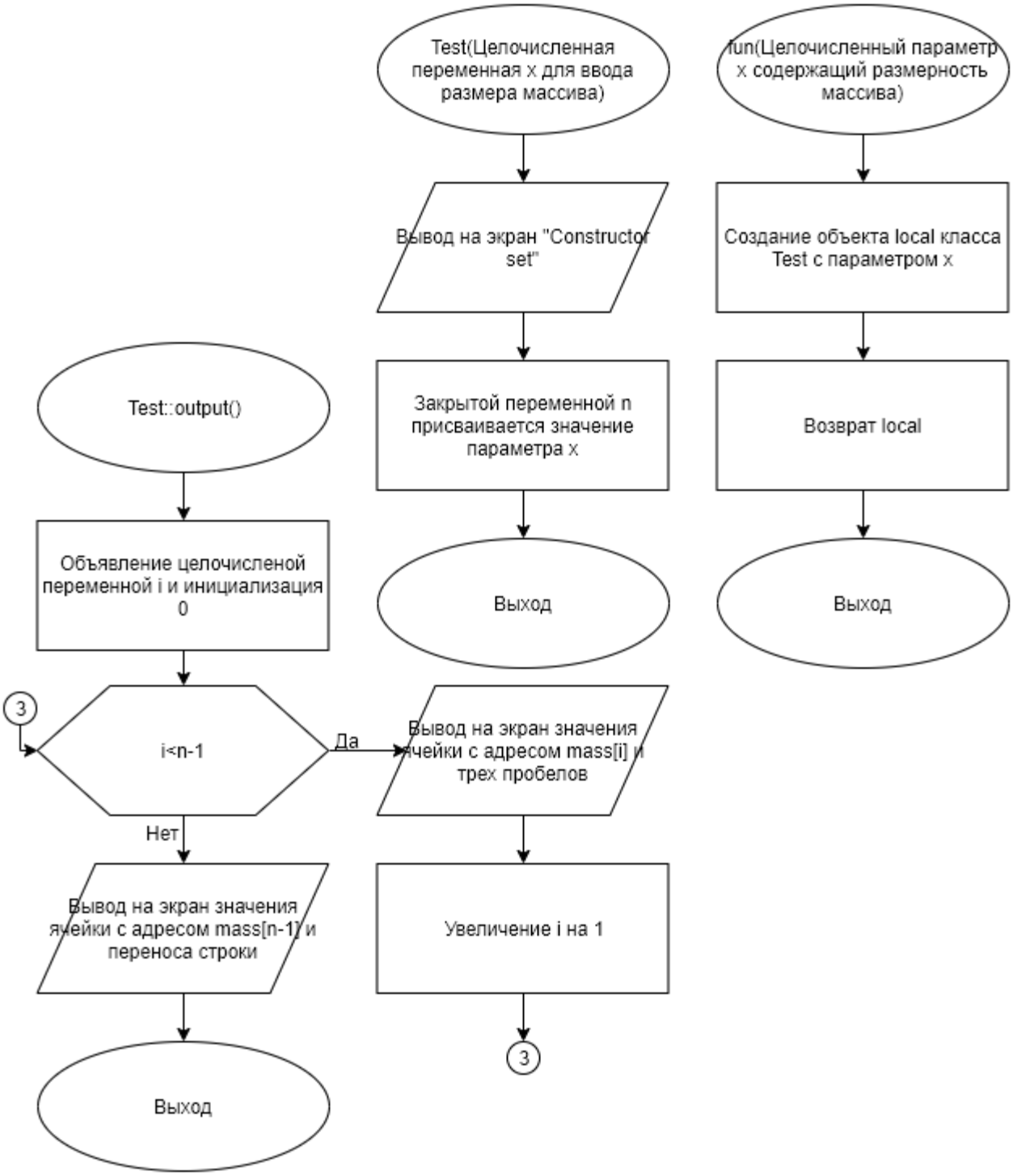

Рисунок 2 – Блок-схема алгоритма

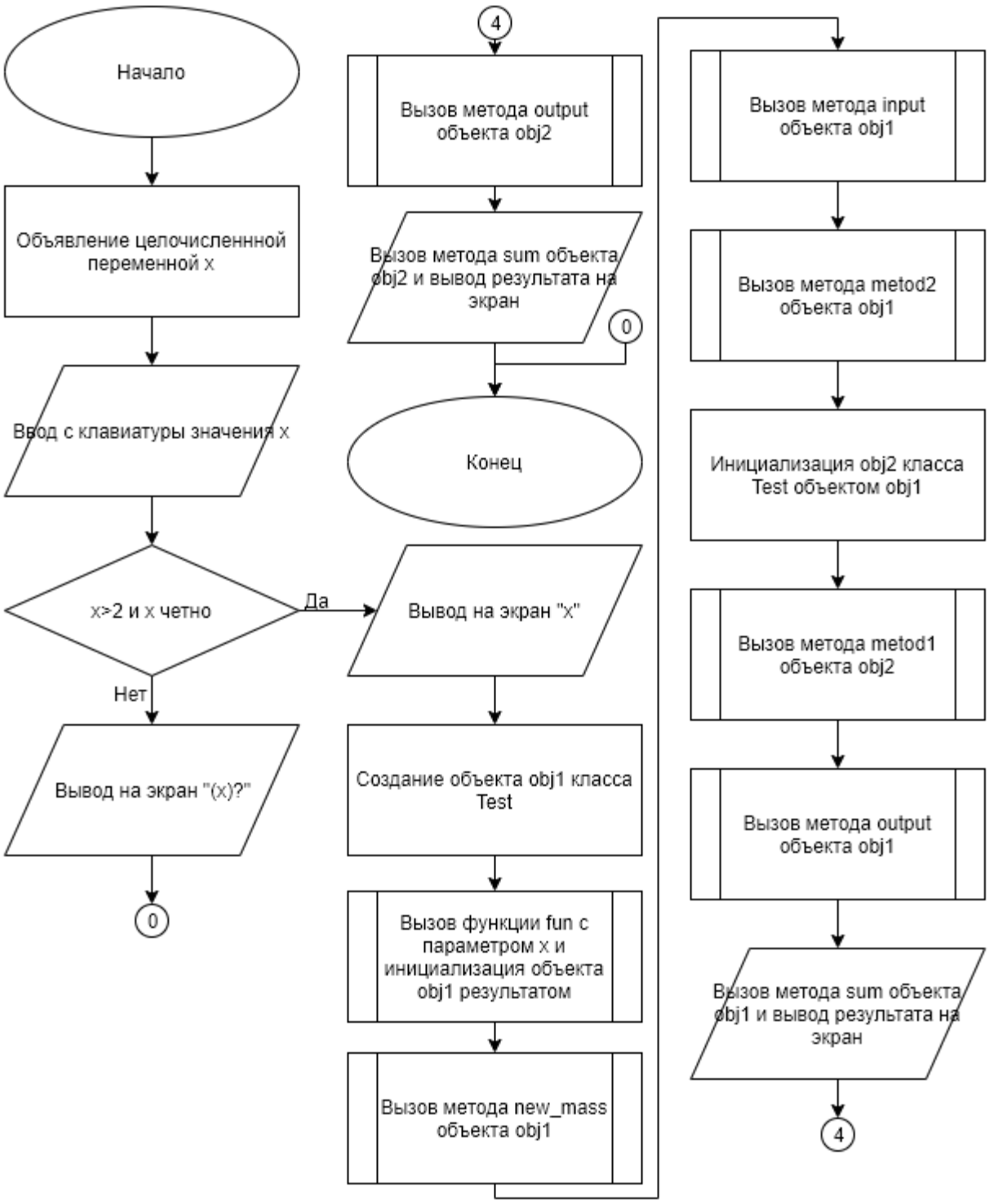

Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

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

using namespace std;

Test func(int x)
{
   Test local(x);
   return local;
}

int main()
{
   int x;
   cin >> x;
   if (x > 2 && x % 2 == 0) {
      cout << x << endl;
      Test obj1;
      obj1 = func(x);
      obj1.new_mass();
      obj1.input();
      obj1.metod2();
      Test obj2 = obj1;
      obj2.metod1();
      obj1.output();
      cout << obj1.sum() << endl;
      obj2.output();
      cout << obj2.sum() << 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;
}

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::new_mass()
{
   mass = new int[n];
}

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

void Test::output()
{
   for(int i = 0; i < n - 1; i++) {
      cout << mass[i] << "   ";
   }
   cout << mass[n - 1] << endl;
}

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

```
   }
   return(summ);
}

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

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

Test::~Test()
{
   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 new_mass();
void input();
void output();
int sum();
void metod1();
void metod2();
~Test();
```

```
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 8.

Таблица 8 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
4 3 5 1 2	4 Default constructor Constructor set Destructor Copy constructor 15 5 2 2 24 20 5 4 2 31 Destructor Destructor	4 Default constructor Constructor set Destructor Copy constructor 15 5 2 2 24 20 5 4 2 31 Destructor Destructor
8 1 2 3 4 5 6 7 8	8 Default constructor Constructor set Destructor Copy constructor 2 2 12 4 30 6 56 8 120 4 2 16 4 36 6 64 8 140 Destructor Destructor	8 Default constructor Constructor set Destructor Copy constructor 2 2 12 4 30 6 56 8 120 4 2 16 4 36 6 64 8 140 Destructor Destructor
3 1 6 3	3?	3?
2 1 2	2?	2?

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-avrora.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-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).

4. Шилдт Г. C++: 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).