

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 Class.....	10
3.2 Алгоритм конструктора класса Class.....	10
3.3 Алгоритм конструктора класса Class.....	11
3.4 Алгоритм деструктора класса Class.....	11
3.5 Алгоритм метода Input класса Class.....	12
3.6 Алгоритм метода SumPara класса Class.....	12
3.7 Алгоритм метода ProizPara класса Class.....	13
3.8 Алгоритм метода Sum класса Class.....	14
3.9 Алгоритм функции Call.....	14
3.10 Алгоритм функции main.....	15
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	16
5 КОД ПРОГРАММЫ.....	23
5.1 Файл Class.cpp.....	23
5.2 Файл Class.h.....	24
5.3 Файл main.cpp.....	25
6 ТЕСТИРОВАНИЕ.....	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	27

# 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. По значению параметра определяется размерность целочисленного массива из закрытой области. Массив создается. 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. Размер должен иметь значение больше 2 и быть четным.
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 `Class` предназначен для демонстрация передачи объекта функциям в качестве параметров;
- функция `Call` для вывод результата работы метода `ProizPara()` объекта `ob`;
- `cin` - объект стандартного потока ввода;
- `cout` - объект стандартного потока вывода;
- `if...else` - условный оператор (оператор ветвления);
- `for` - оператор цикла со счетчиком.

Класс `Class`:

- свойства/поля:
  - поле `mas` размера массива `mas`:
    - наименование — `n`;
    - тип — `int`;
    - модификатор доступа — `private`;
  - поле `mas` целочисленного массива `mas`:
    - наименование — `mas`;
    - тип — `int`;
    - модификатор доступа — `private`;
- функционал:
  - метод `Class` — конструктор по умолчанию, в начале работы выдает сообщение об отработке;
  - метод `Class(int n)` — параметризованный конструктор, передается целочисленный параметр. По значению параметра определяется размерность целочисленного массива из закрытой области;
  - метод `Class(const Class& obj)` — конструктор копии для исключения

ошибки при передаче объекта в функцию по значению;

- о метод `~Class` — деструктор, выводит сообщение об отработке;
- о метод `Input` — метод для ввода значений элементов массива;
- о метод `SumPara` — метод, который суммирует значения очередной пары элементов и присваивает сумму первому элементу пары, далее суммирует элементы полученного массива и возвращает это значение;
- о метод `ProizPara` — метод, который умножает значения очередной пары элементов и результат присваивает первому элементу пары, далее суммирует элементы полученного массива и возвращает это значение;
- о метод `Sum` — суммирует значения элементов массива.

## 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 Class

Функционал: конструктор по умолчанию, в начале работы выдает сообщение об отработке.

Параметры: none.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Class

№	Предикат	Действия	№ перехода
1		вывод "Default constructor" и перенос на новую строку	Ø

### 3.2 Алгоритм конструктора класса Class

Функционал: параметризованный конструктор, передается целочисленный параметр. По значению параметра определяется размерность целочисленного массива из закрытой области.

Параметры: int n.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса Class

№	Предикат	Действия	№ перехода
1		вывод "Constructor set" и перенос на новую строку	2
2		создание целочисленного массива mas размерности n	3



№	Предикат	Действия	№ перехода
3		присвоение значению переменной класса n класса Class значения параметра n, поданного в конструктор	Ø

### 3.3 Алгоритм конструктора класса Class

Функционал: конструктор копии для исключения ошибки при передаче объекта в функцию по значению.

Параметры: Class& obj.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса Class

№	Предикат	Действия	№ перехода
1		вывод "Copy Constructor" и перенос на новую строку	2
2		копирование значение переменной n	3
3		объявление целочисленного массива mas размерности n	4
4		инициализация целочисленной переменной i со значением 0	5
5	i < n	копирование i-го элемента массива mas	6
			Ø
6		инкремент i	5

### 3.4 Алгоритм деструктора класса Class

Функционал: деструктор, выводит сообщение об отработке.

Параметры: none.

Алгоритм деструктора представлен в таблице 4.

Таблица 4 – Алгоритм деструктора класса Class

№	Предикат	Действия	№ перехода
1		удаление массива по адресу указателя mas	Ø

### 3.5 Алгоритм метода Input класса Class

Функционал: метод для ввода значений элементов массива.

Параметры: none.

Возвращаемое значение: none.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода Input класса Class

№	Предикат	Действия	№ перехода
1		объявление переменной x типа int	2
2		инициализация переменной i типа int со значением 0	3
3	i < n	ввод значения переменной x	4
			Ø
4		присваивание значению i-го элемента массива mas значения x	5
5		инкремент i	3

### 3.6 Алгоритм метода SumPara класса Class

Функционал: метод, который суммирует значения очередной пары элементов и присваивает сумму первому элементу пары, далее суммирует элементы полученного массива и возвращает это значение.

Параметры: none.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 6.

Таблица 6 – Алгоритм метода *SumPara* класса *Class*

№	Предикат	Действия	№ перехода
1		инициализация переменной <i>i</i> типа <i>int</i> со значением 0	2
2	<i>i</i> < <i>n</i>	присваивание значению <i>i</i> -го элемента массива <i>mas</i> значения суммы <i>i</i> -го элемента и следующего	3
		возврат результата работы метода <i>Sum()</i> текущего объекта	Ø
3		увеличивает значение <i>i</i> на 2	2

### 3.7 Алгоритм метода *ProizPara* класса *Class*

Функционал: метод, который умножает значения очередной пары элементов и результат присваивает первому элементу пары, далее суммирует элементы полученного массива и возвращает это значение.

Параметры: none.

Возвращаемое значение: *int*.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода *ProizPara* класса *Class*

№	Предикат	Действия	№ перехода
1		инициализация переменной <i>i</i> типа <i>int</i> со значением 0	2
2	<i>i</i> < <i>n</i>	присваивание значению <i>i</i> -го элемента массива <i>mas</i> значения произведения <i>i</i> -го элемента и следующего	3
		возврат результата работы метода <i>Sum()</i> текущего объекта	Ø

№	Предикат	Действия	№ перехода
3		увеличение значения i на 2	2

### 3.8 Алгоритм метода Sum класса Class

Функционал: суммирует значения элементов массива.

Параметры: none.

Возвращаемое значение: int.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода Sum класса Class

№	Предикат	Действия	№ перехода
1		инициализация переменной s типа int со значением 0	2
2		инициализация переменной i типа int со значением 0	3
3	i < n	значение s увеличивается на значение i-го элемента массива mas	4
		возврат значения переменной s	∅
4		инкремент i	3

### 3.9 Алгоритм функции Call

Функционал: вывод результата работы метода ProizPara объекта ob.

Параметры: Class ob.

Возвращаемое значение: none.

Алгоритм функции представлен в таблице 9.

Таблица 9 – Алгоритм функции Call

№	Предикат	Действия	№ перехода
1		вывод значения, возвращаемого методом ProizPara() для объекта ob	Ø

### 3.10 Алгоритм функции main

Функционал: запуск программы.

Параметры: none.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление переменной size типа int	2
2		ввод значения переменной size	3
3	size<=2  size%2!=0	вывод значения переменной size и "?"	12
			4
4		вывод значения переменной size	5
5		вывод переноса на новую строку	6
6		создание объекта obj класса Class с параметром size	7
7		вызов метода Input() объекта obj	8
8		вызов функции Call(obj)	9
9		вывод переноса на новую строку	10
10		вызов метода SumPara() объекта obj	11
11		вывод значенияБ возвратимого методом Sum() объекта obj	12
12		возврат значения 0	Ø

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

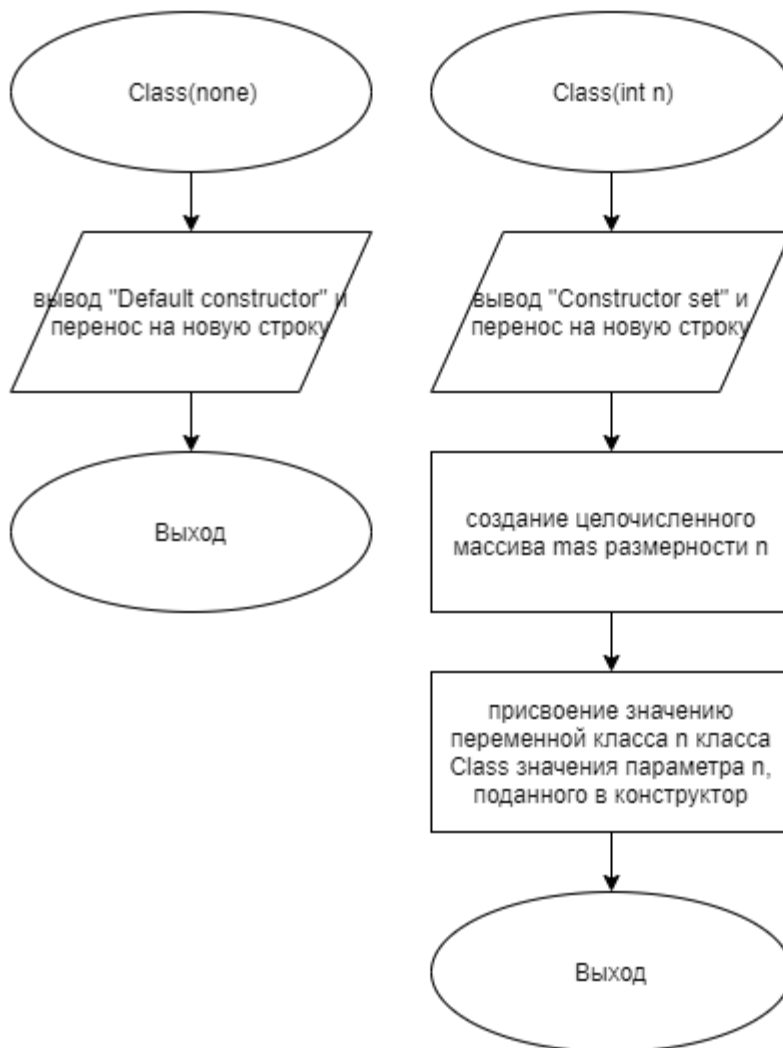


Рисунок 1 – Блок-схема алгоритма

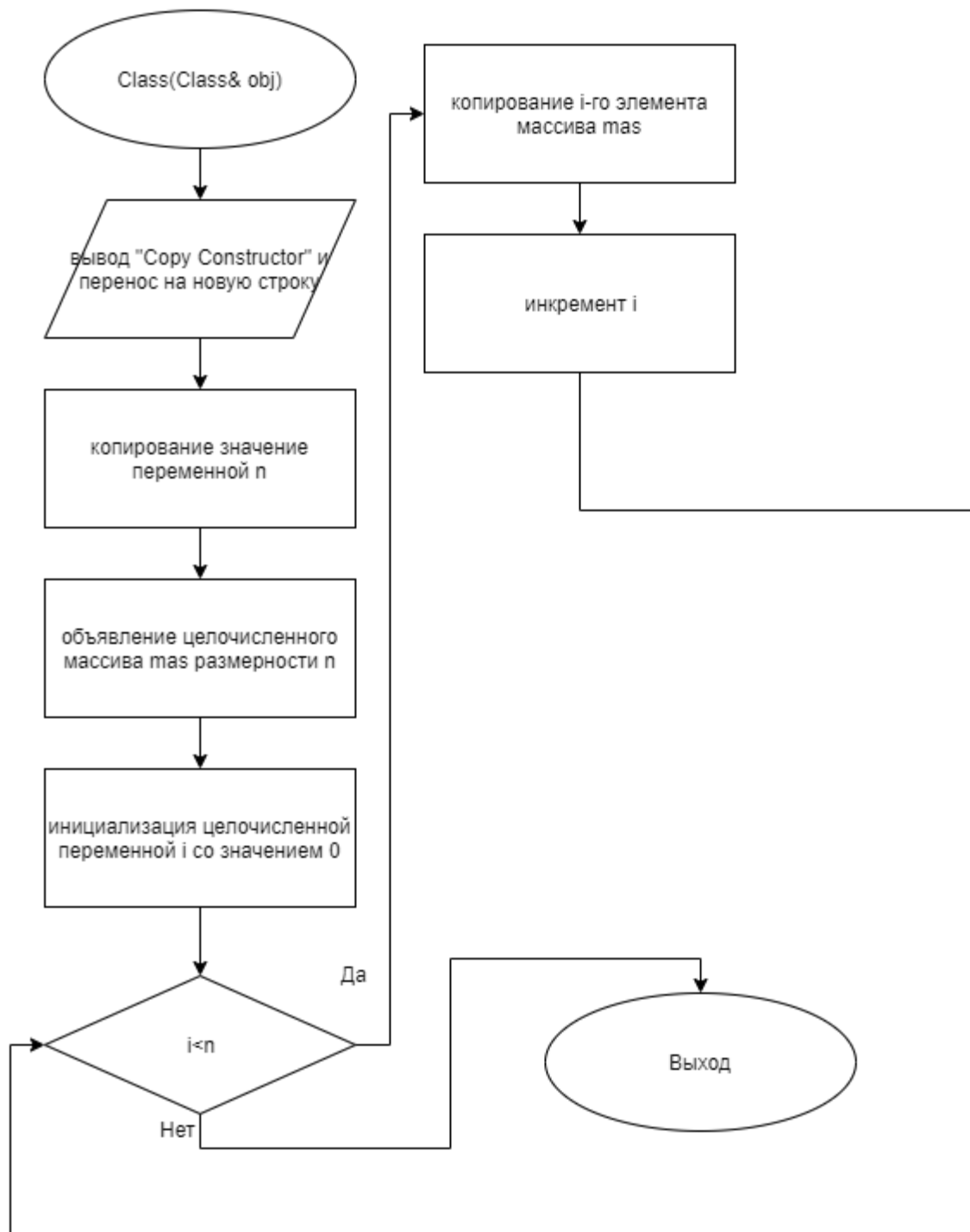


Рисунок 2 – Блок-схема алгоритма

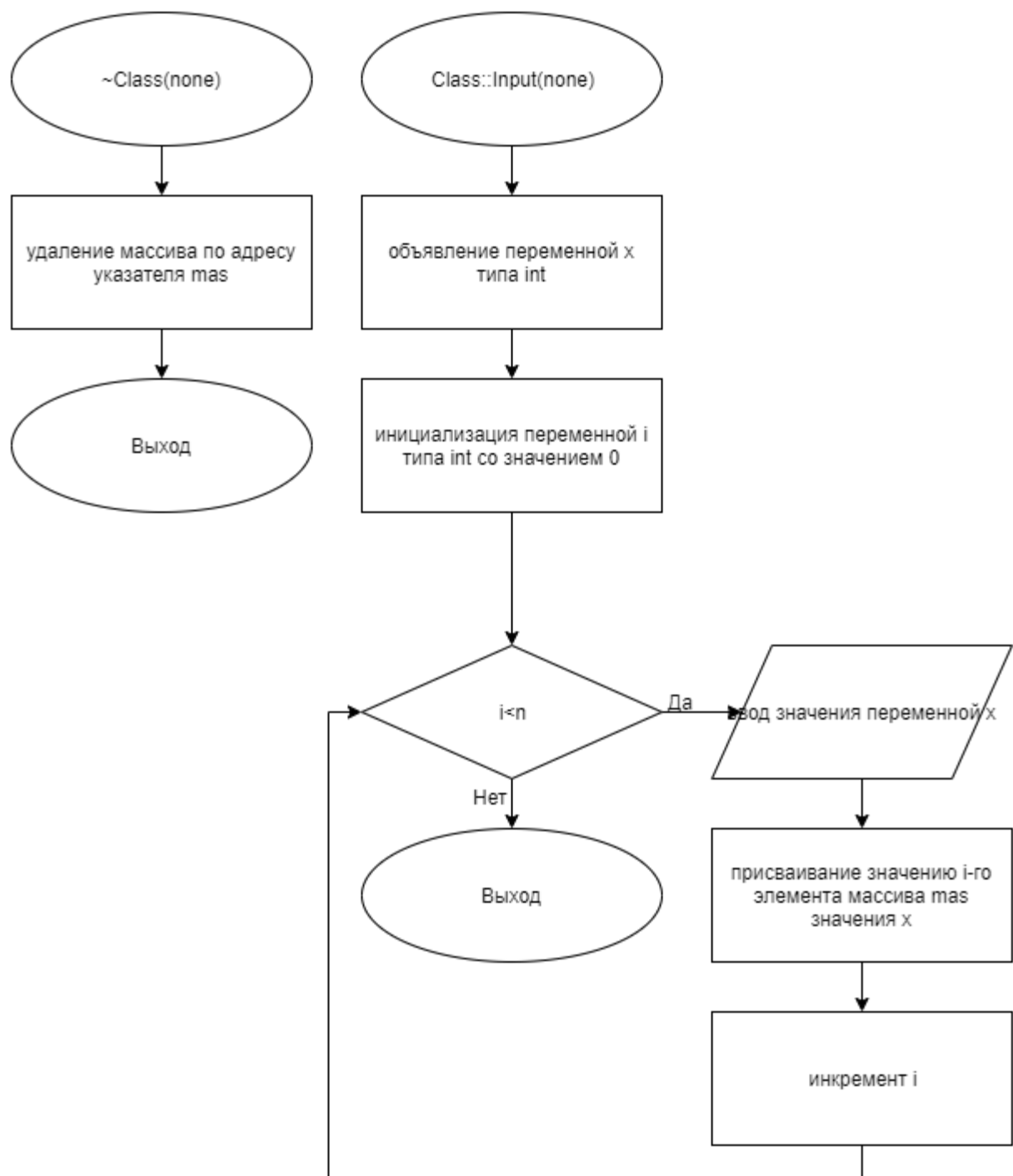
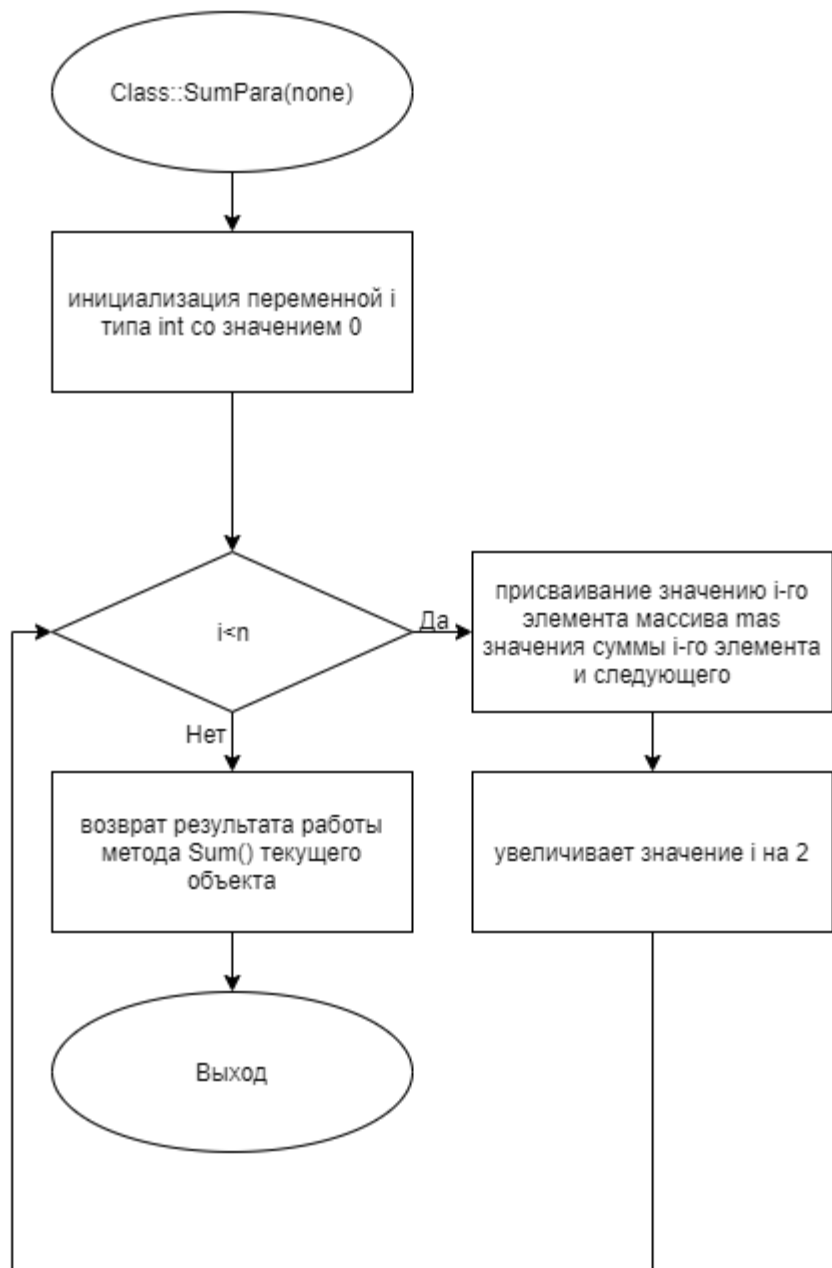


Рисунок 3 – Блок-схема алгоритма





**Рисунок 4 – Блок-схема алгоритма**

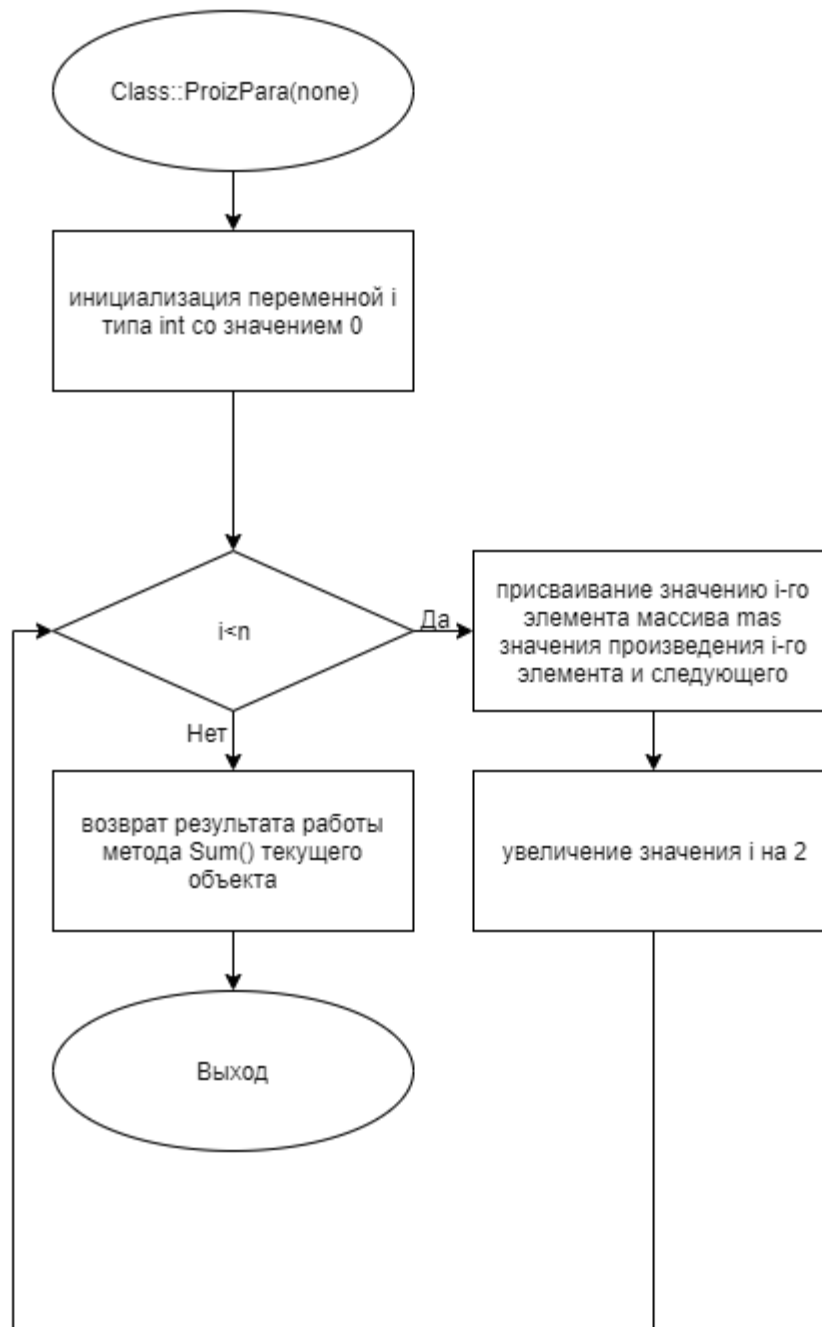


Рисунок 5 – Блок-схема алгоритма

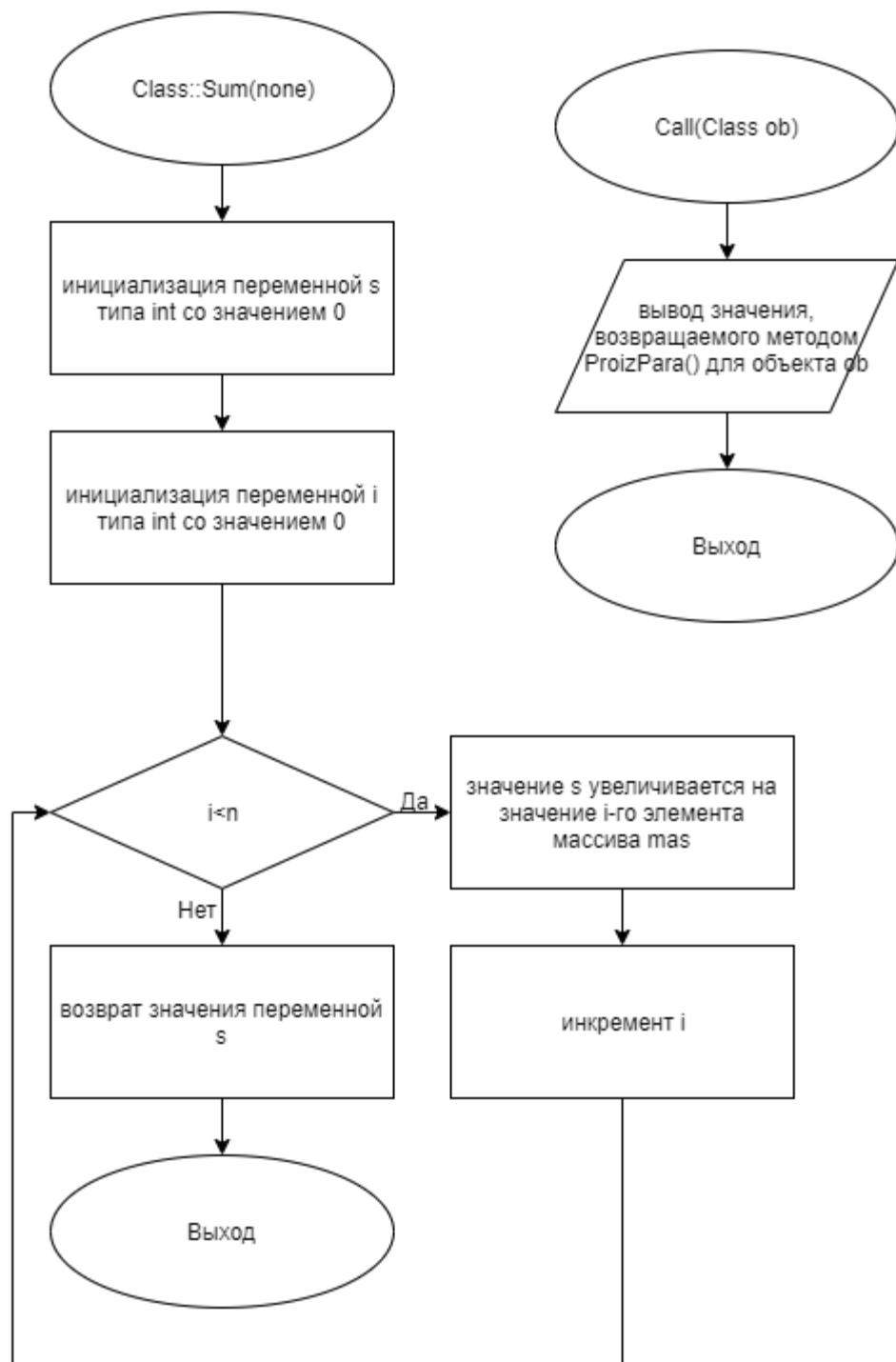


Рисунок 6 – Блок-схема алгоритма

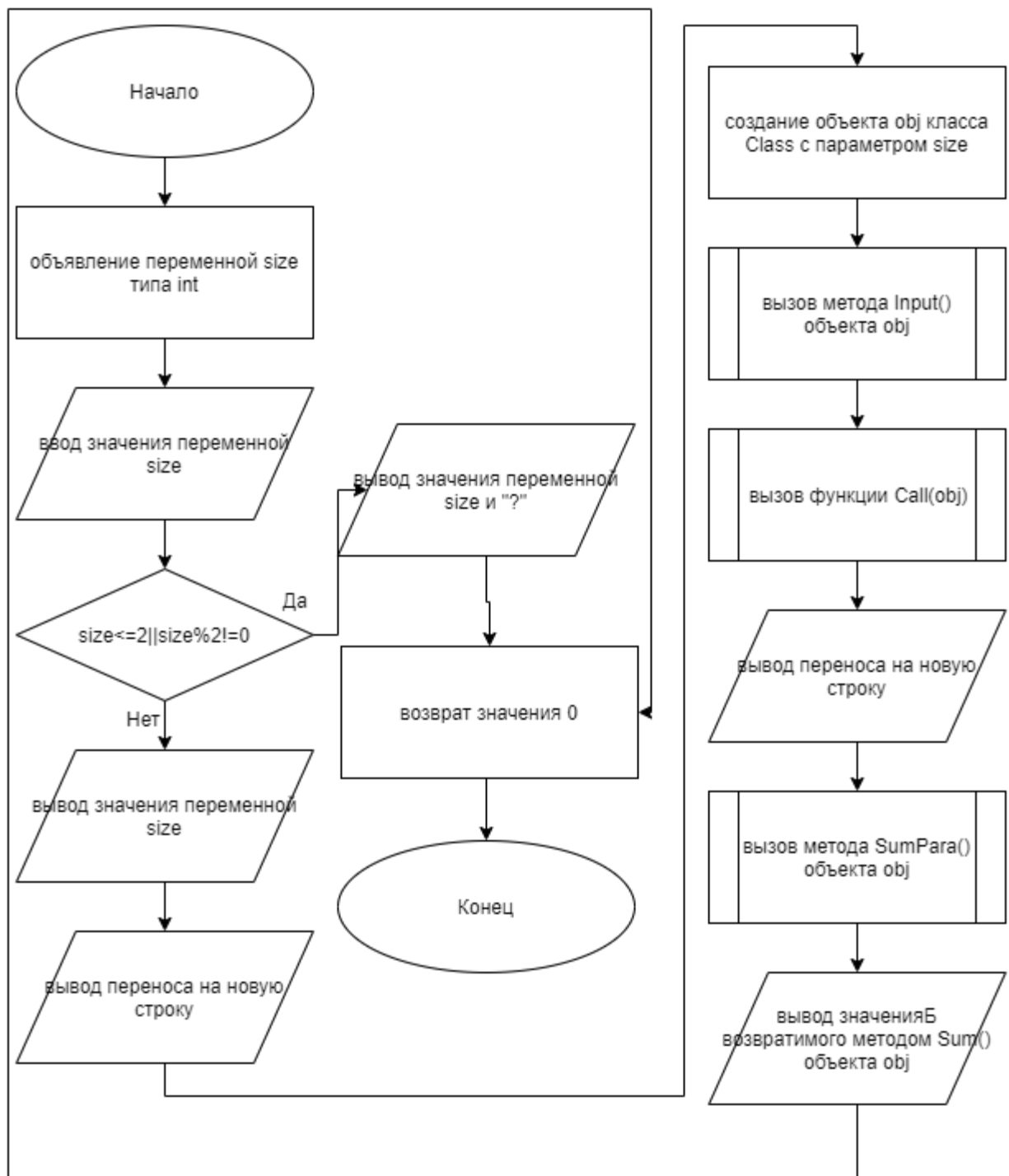


Рисунок 7 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл Class.cpp

*Листинг 1 – Class.cpp*

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

using namespace std;

Class::Class()
{
    cout<<"Default constructor"<<endl;
}
Class::Class(int n)
{
    cout<<"Constructor set"<<endl;
    mas=new int[n];
    this->n=n;
}
Class::Class(const Class& obj)
{
    cout<<"Copy constructor"<<endl;
    n=obj.n;
    mas=new int[n];
    for (int i=0; i<n; i++)
    {
        mas[i]=obj.mas[i];
    }
}
Class::~~Class()
{
    cout<<endl<<"Destructor";
    if (mas!=nullptr)
    {
        delete[] mas;
    }
}
void Class::Input()
{
    int x;
    for (int i=0; i<n; i++)
    {
        cin>>x;
```

```

        mas[i]=x;
    }
}
int Class::SumPara()
{
    for (int i=0; i<n; i+=2)
    {
        mas[i]=mas[i]+mas[i+1];
    }
    return Sum();
}
int Class::ProizPara()
{
    for (int i=0; i<n; i+=2)
    {
        mas[i]=mas[i]*mas[i+1];
    }
    return Sum();
}
int Class::Sum()
{
    int s=0;
    for (int i=0; i<n; i++)
    {
        s+=mas[i];
    }
    return s;
}

```

## 5.2 Файл Class.h

*Листинг 2 – Class.h*

```

#ifndef __CLASS__H
#define __CLASS__H

using namespace std;

class Class
{
private:
    int n;
    int *mas;
public:
    Class();
    Class(int n);
    Class(const Class& obj);
    ~Class();
    void Input();
    int SumPara();
}

```

```
        int ProizPara();  
        int Sum();  
};  
  
#endif
```

## 5.3 Файл main.cpp

*Листинг 3 – main.cpp*

```
#include <stdlib.h>  
#include <stdio.h>  
#include <iostream>  
#include "Class.h"  
  
using namespace std;  
  
void Call(Class ob)  
{  
    cout<<ob.ProizPara();  
}  
  
int main()  
{  
    int size;  
    cin>>size;  
    if (size<=2||size%2!=0)  
    {  
        cout<<size<<"?";  
        return 0;  
    }  
    cout<<size;  
    cout<<endl;  
    Class obj(size);  
    obj.Input();  
    Call(obj);  
    cout<<endl;  
    obj.SumPara();  
    cout<<obj.Sum();  
    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



## 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).