

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.....	9
3 ОПИСАНИЕ АЛГОРИТМОВ.....	10
3.1 Алгоритм метода create_а класса classname.....	10
3.2 Алгоритм метода print класса classname.....	10
3.3 Алгоритм функции main.....	11
3.4 Алгоритм функции function.....	12
3.5 Алгоритм метода fill_а класса classname.....	12
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	14
5 КОД ПРОГРАММЫ.....	17
5.1 Файл classname.cpp.....	17
5.2 Файл classname.h.....	18
5.3 Файл main.cpp.....	19
6 ТЕСТИРОВАНИЕ.....	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	21

# 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;
- Параметризованный 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 параметризованного 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

Параметризованный 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 `obj` класса `classname` предназначен для демонстрация применения конструктора копирования;
- объект `obj_2` класса `classname` предназначен для демонстрация применения конструктора копирования;
- функция `function` для возврата локально созданного объекта;
- функция `assert` для сообщения об ошибке;
- `cin/cout` - объекты стандартного потока ввода/вывода;
- `if..else` - условный оператор;
- `for` - оператор цикла со счётчиком.

Класс `classname`:

- функционал:
  - метод `create_a` — создание массива заданного размера;
  - метод `print` — вывод массива;
  - метод `fill_a` — ввод 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 метода `create_a` класса `classname`

Функционал: создание массива заданного размера.

Параметры: нет.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода `create_a` класса `classname`

№	Предикат	Действия	№ перехода
1		полю <code>a</code> текущего объекта присвоить массив размера поля <code>len_a</code> текущего объекта	Ø

### 3.2 Алгоритм метода `print` класса `classname`

Функционал: вывод массива.

Параметры: нет.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `print` класса `classname`

№	Предикат	Действия	№ перехода
1		инициализация переменной <code>n</code> типа <code>int</code> полем <code>len_a</code> текущего объекта	2



№	Предикат	Действия	№ перехода
2		вывод элемента a[0] текущего объекта	3
3		инициализация переменной i типа int значением 1	4
4	i < n	вывод элемента a[i] текущего объекта	5
			∅
5		++i	4

### 3.3 Алгоритм функции main

Функционал: основная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление n типа int	2
2		ввод значения n	3
3	!(n > 2 && n % 2 == 0)	вывод "{n}?"	15
		вывод значения n	4
4		создание объекта obj класса classname	5
5		присвоение объекту obj результата вызова функции function с параметром n	6
6		вызов метода create_a объекта obj	7
7		вызов метода fill_a объекта obj	8
8		вызов метода half_product объекта obj	9
9		инициализация объекта obj_2 класса classname объектом obj	10
10		вызов метода half_sum объекта obj_2	11
11		вызов метода print объекта obj	12

№	Предикат	Действия	№ перехода
12		вывод результата вызова метода sum_a объекта obj	13
13		вызов метода print объекта obj_2	14
14		вывод результата вызова метода sum_a объекта obj_2	15
15		возврат значения 0	∅

### 3.4 Алгоритм функции function

Функционал: возврат локально созданного объекта.

Параметры: int n.

Возвращаемое значение: classname - локально созданный объект.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции function

№	Предикат	Действия	№ перехода
1		создание объекта obj_local класса classname с параметром конструктора n	2
2		возврат объекта obj_local	∅

### 3.5 Алгоритм метода fill\_a класса classname

Функционал: ввод массива.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода fill\_a класса classname

№	Предикат	Действия	№ перехода
1		инициализация переменных типа int: n полем len_a	2

№	Предикат	Действия	№ перехода
		текущего объекта, $i$ значением 0; объявление переменной $x$ типа <code>int</code>	
2	$i < n$	ввод значения $x$	3
			∅
3		элементу $a[i]$ текущего объекта присовить значение $x$	4
4		$++i$	2

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

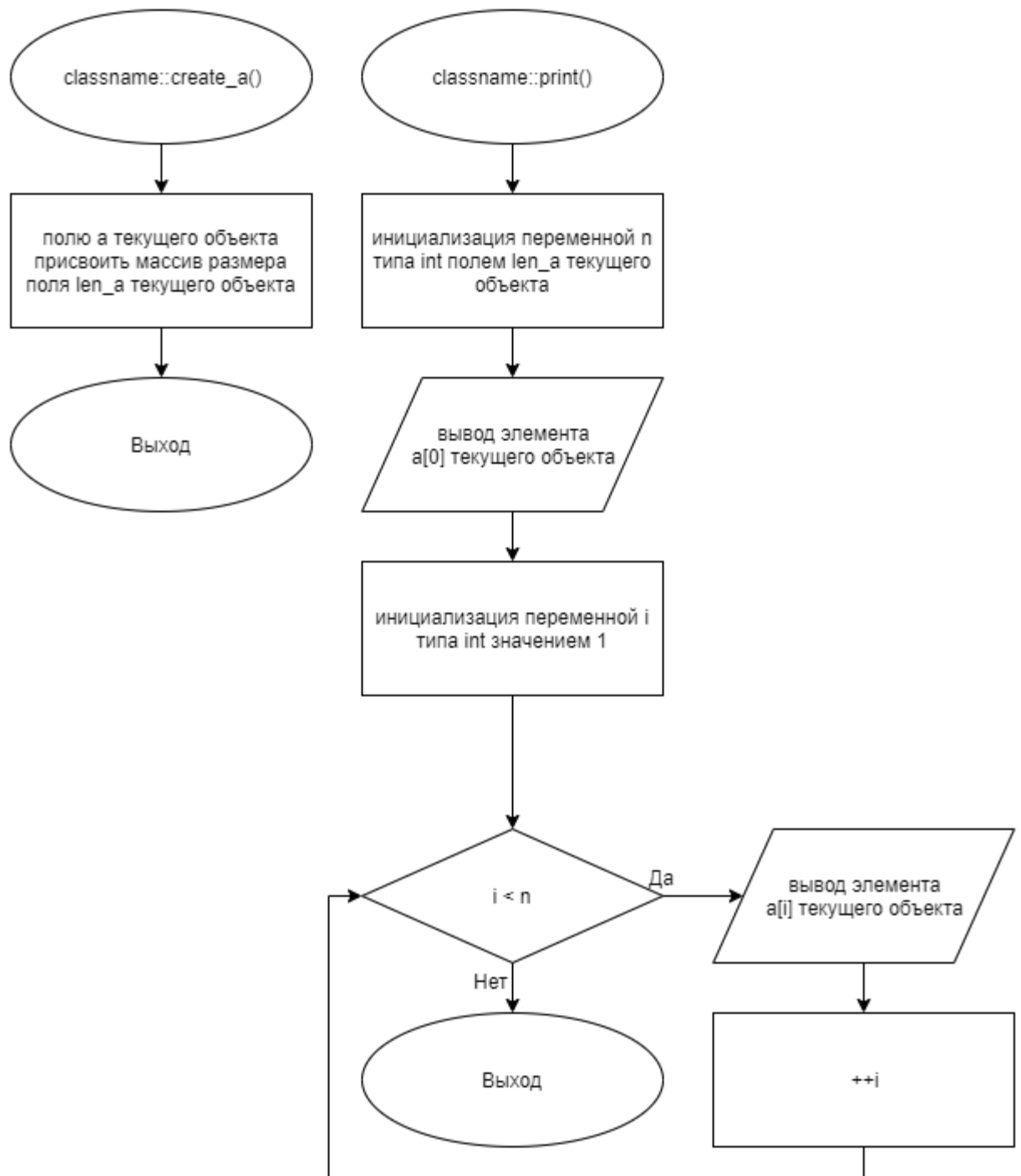


Рисунок 1 – Блок-схема алгоритма

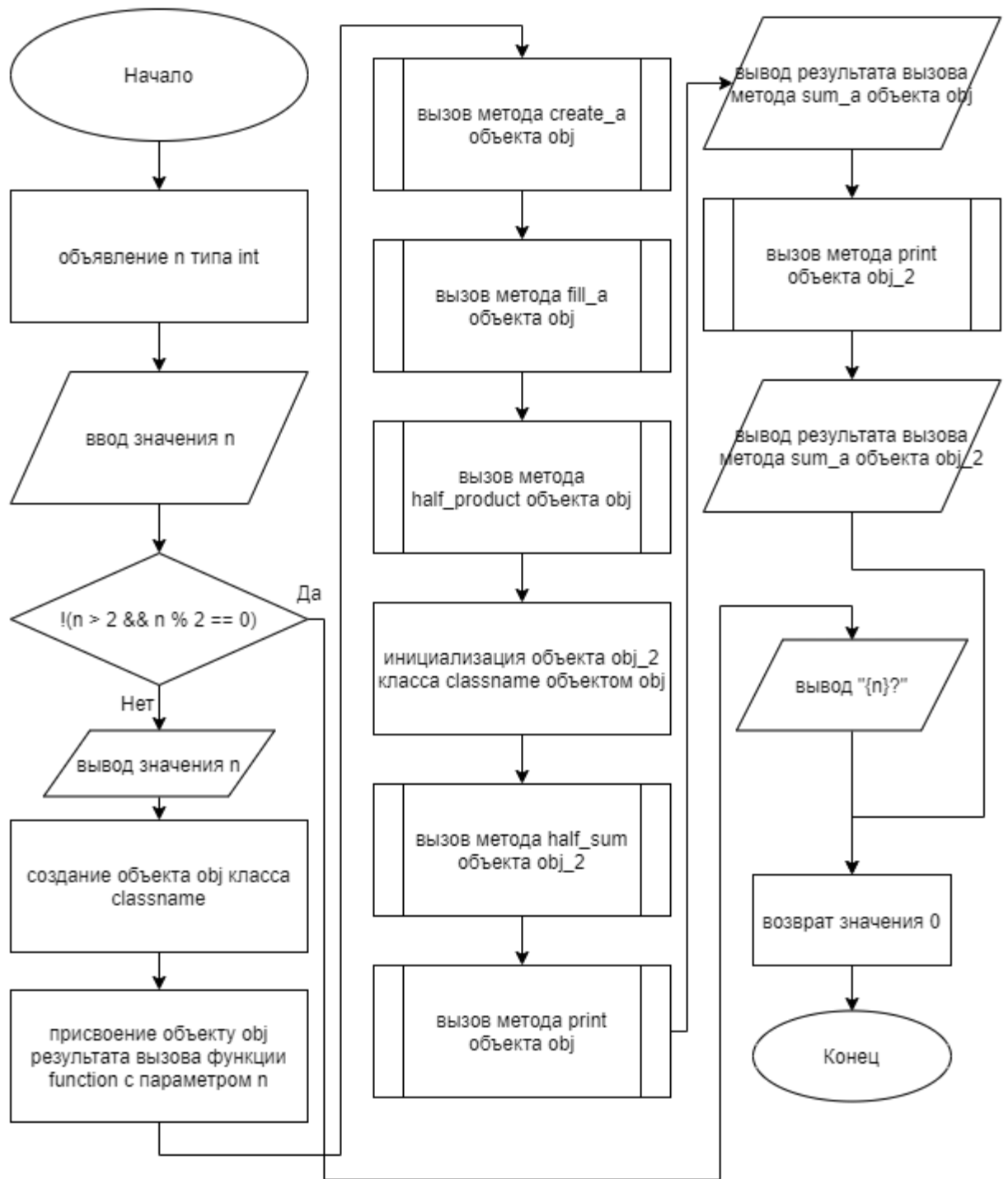


Рисунок 2 – Блок-схема алгоритма

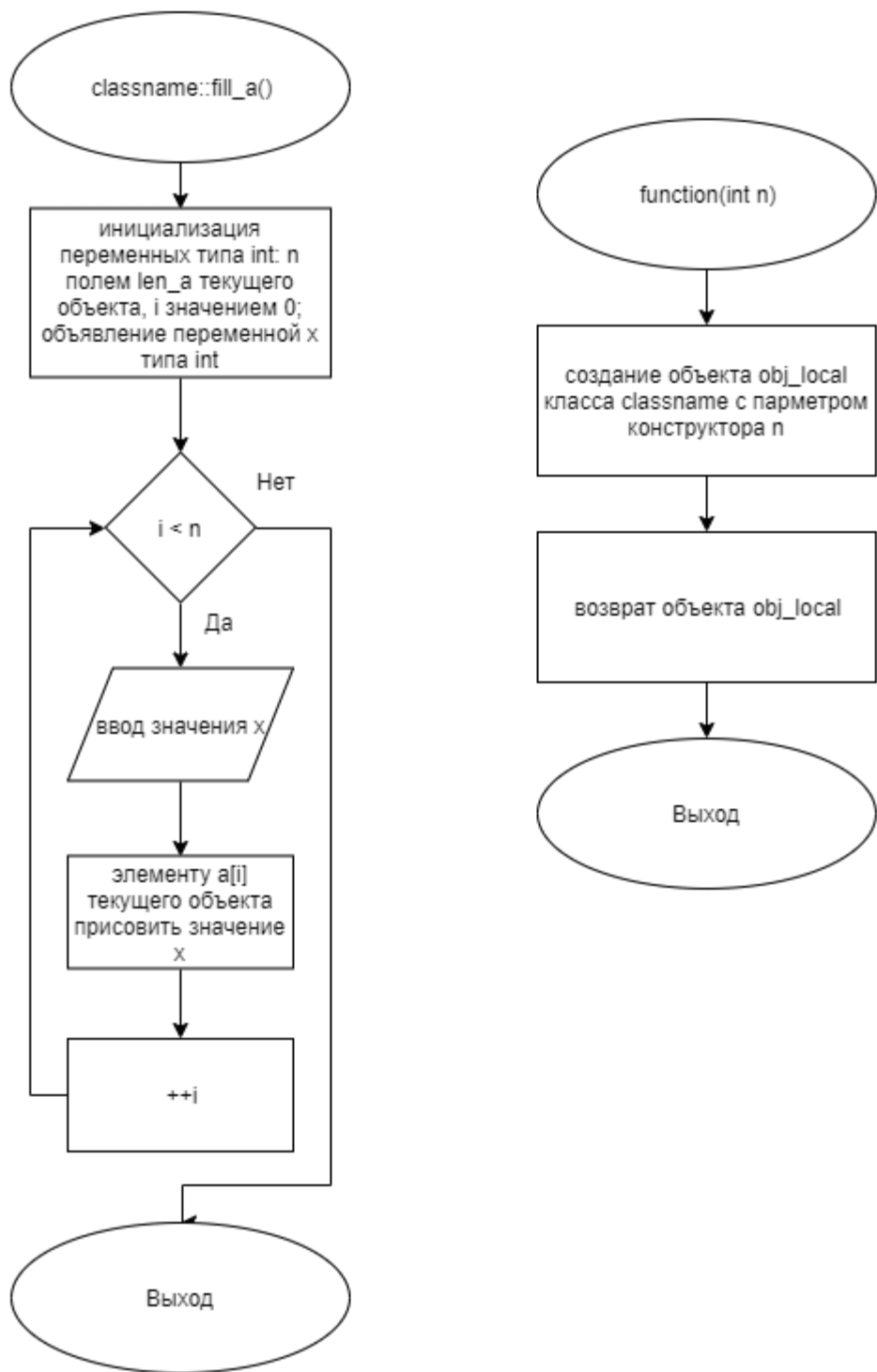


Рисунок 3 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл `classname.cpp`

*Листинг 1 – `classname.cpp`*

```
#include "classname.h"
#include <iostream>
#include <cassert>

classname::classname(){
    std::cout << "\nDefault constructor";
};

classname::classname(int n){
    assert(n>2 && n%2==0);
    this->len_a = n;
    std::cout << "\nConstructor set";
};

classname::classname(const classname & obj){
    len_a = obj.len_a;
    a = obj.a;
    std::cout<<"\nCopy constructor";
};

classname::~classname(){
    std::cout << "\nDestructor";
};

void classname::create_a(){
    this->a.assign(this->len_a, 0);
};

void classname::fill_a(){
    int n = this->len_a, x;
    for(int i=0;i<n;++i){
        std::cin>>x;
        this->a[i] = x;
    }
};

int classname::sum_a(){
```

```

        int n = this->len_a, summa = 0;
        for(int i=0;i<n;++i)
            summa+=this->a[i];
        return summa;
};

int classname::half_sum(){
    int n = this->len_a;
    for(int i=0;i < n - 1;i+=2)
        this->a[i] = this->a[i] + this->a[i+1];
    return this->sum_a();
};

int classname::half_product(){
    int n = this->len_a;
    for(int i=0;i < n - 1;i+=2)
        this->a[i] = this->a[i] * this->a[i+1];
    return this->sum_a();
};

void classname::print(){
    int n = this->len_a;
    std::cout<<'\\n';
    std::cout<<this->a[0];
    for (int i = 1; i < n; ++i)
        std::cout <<"    " << this->a[i];
};

```

## 5.2 Файл classname.h

*Листинг 2 – classname.h*

```

#ifndef __CLASSNAME__H
#define __CLASSNAME__H
#include <vector>

class classname{
    std::vector<int>a;
    int len_a;
public:
    classname();
    classname(int n);
    classname(const classname & obj);
    ~classname();
    void create_a();
    void fill_a();
    int sum_a();
    int half_sum();
    int half_product();
    void print();
};

```



```
};  
  
#endif
```

## 5.3 Файл main.cpp

*Листинг 3 – main.cpp*

```
//#include <stdlib.h>  
//#include <stdio.h>  
#include <iostream>  
#include "classname.h"  
  
classname function(int n){  
    classname obj_local(n);  
    return obj_local;  
}  
  
int32_t main()  
{  
    int n; std::cin>>n;  
    if(!(n>2 && n%2==0)){  
        std::cout<<n<<'?';  
        return 0;  
    }  
    std::cout << n;  
    classname obj;  
    obj = function(n);  
    obj.create_a();  
    obj.fill_a();  
    obj.half_product();  
  
    classname obj_2 = obj;  
    obj_2.half_sum();  
  
    obj.print();  
    std::cout<<'\\n'<<obj.sum_a();  
  
    obj_2.print();  
    std::cout<<'\\n'<<obj_2.sum_a();  
  
    return(0);  
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

Таблица 6 – 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
2	2?	2?
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](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).