

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 конструктора класса Cls.....	10
3.2 Алгоритм метода init_arr класса Cls.....	10
3.3 Алгоритм метода output класса Cls.....	11
3.4 Алгоритм функции func.....	11
3.5 Алгоритм функции main.....	12
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	16
5.1 Файл Cls.cpp.....	16
5.2 Файл Cls.h.....	17
5.3 Файл main.cpp.....	18
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, в начале работы выдает сообщение;
- Параметризованный конструктор, передается целочисленный параметр. Параметр должен иметь значение больше 2 и быть четным. В начале работы выдает сообщение;
- Конструктор копии, обеспечивает создание копии объекта в новой области памяти. В начале 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 obj1 класса Cls предназначен для Первый объект для решения задачи;
- объект obj2 класса Cls предназначен для Второй объект для решения задачи;
- функция func для Функция инициализации объекта.

Класс Cls:

- функционал:
  - метод Cls — Параметризованный конструктор;
  - метод init\_arr — Метод инициализации памяти в массиве arr;
  - метод output — Метод вывода содержимого массива arr.

## 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 Cls

Функционал: Параметризированный конструктор.

Параметры: int n - размер массива.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Cls

№	Предикат	Действия	№ перехода
1		Вывод Constructor set	2
2		length = n	Ø

### 3.2 Алгоритм метода init\_arr класса Cls

Функционал: Метод инициализации памяти в массиве arr.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода init\_arr класса Cls

№	Предикат	Действия	№ перехода
1		Выделение памяти по указателю arr размером n	Ø



### 3.3 Алгоритм метода output класса Cls

Функционал: Метод вывода содержимого массива arr.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода output класса Cls

№	Предикат	Действия	№ перехода
1		Инициализация $i = 0$ типа int	2
2	$i < \text{length}$	Вывод значения arr[i]	3
			Ø
3	$i \neq \text{length} - 1$	Вывод трёх пробелов	4
			4
4		$i++$	2

### 3.4 Алгоритм функции func

Функционал: Инициализация объекта класса Cls.

Параметры: int n - длина массива.

Возвращаемое значение: Cls - объект класса Cls.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции func

№	Предикат	Действия	№ перехода
1		Создание объекта obj класса Cls с передачей аргумента n	2
2		Возврат obj	Ø

### 3.5 Алгоритм функции main

Функционал: Основная функция программы.

Параметры: нет.

Возвращаемое значение: int - код ошибки.

Алгоритм функции представлен в таблице 5.

Таблица 5 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление n типа int	2
2		Ввод значения n	3
3	$n \% 2 \parallel n \leq 2$	Вывод n?	16
			4
4		Вывод n	5
5		Создание объекта obj1 класса Cls	6
6		Присвоение объекту obj1 значения вызова функции func с аргументом n	7
7		Вызов метода init_arg у объекта obj1	8
8		Вызов метода input у объекта obj1	9
9		Вызов метода m2 у объекта obj1	10
10		Создание объекта obj2 класса Cls с аргументом obj1	11
11		Вызов метода m1 у объекта obj2	12
12		Вызов метода output у объекта obj1	13
13		Вывод значения вызова метода m3 у объекта obj1	14
14		Вызов метода output у объекта obj2	15
15		Вывод значения вызова метода m3 у объекта obj2	16
16		Возврат значения 0	∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

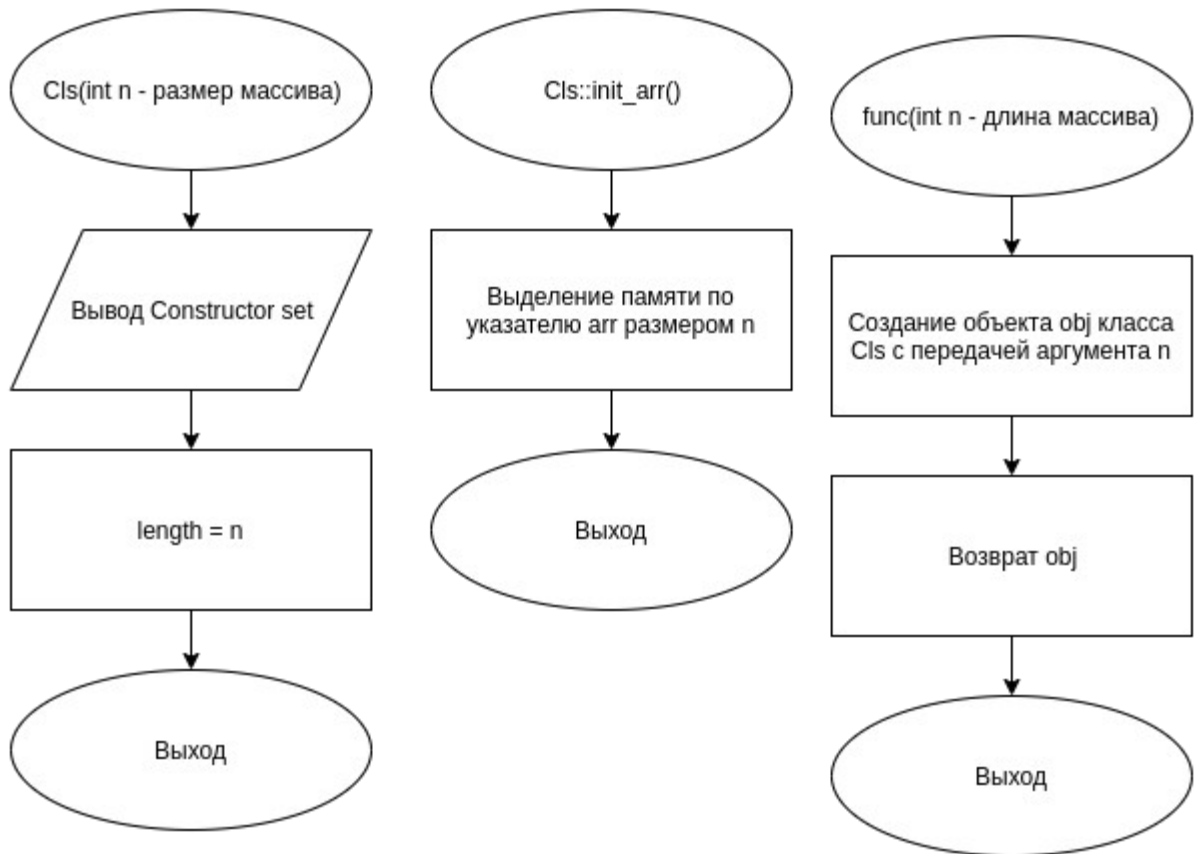
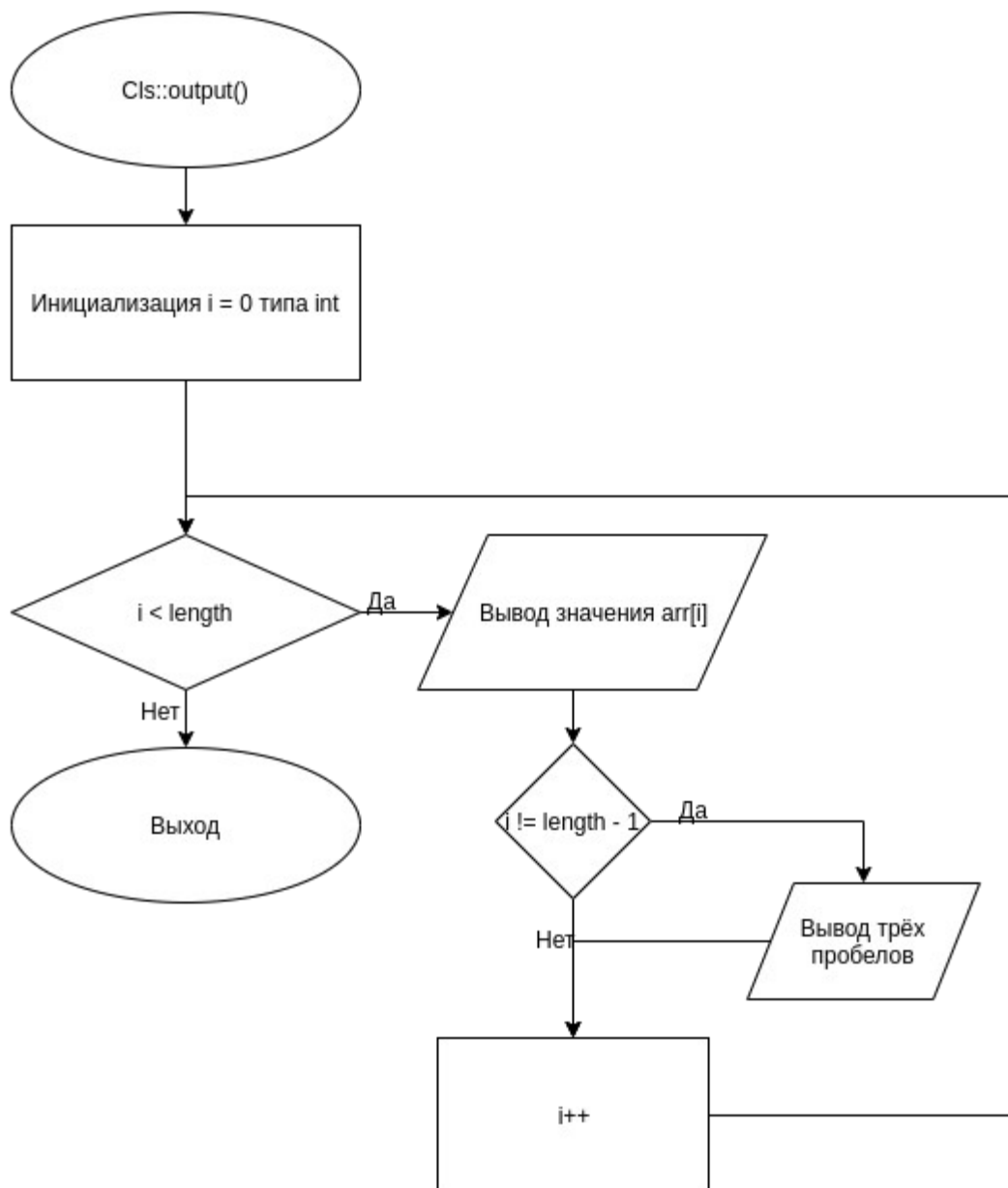


Рисунок 1 – Блок-схема алгоритма



**Рисунок 2 – Блок-схема алгоритма**

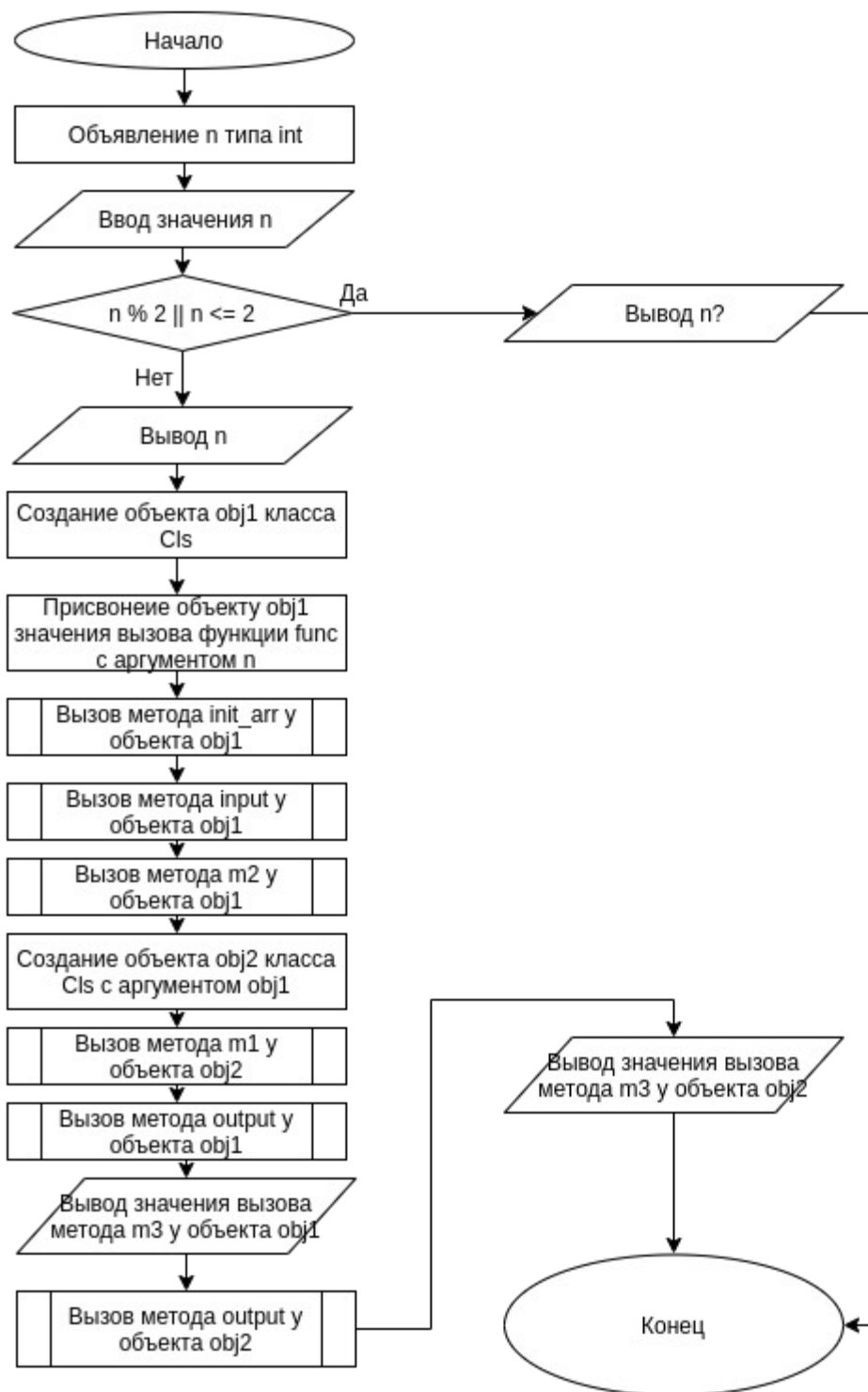


Рисунок 3 – Блок-схема алгоритма

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл Cls.cpp

*Листинг 1 – Cls.cpp*

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

using namespace std;

Cls::Cls()
{
    cout << endl << "Default constructor";
}
Cls::Cls(int n)
{
    cout << endl << "Constructor set";
    length = n;
}
Cls::~Cls()
{
    cout << endl << "Destructor";
    delete arr;
}
Cls::Cls(const Cls & ob)
{
    cout << endl << "Copy constructor";

    length = ob.length;
    arr = new int[ob.length];
    for(int i = 0; i < length; i++)
    {
        arr[i] = ob.arr[i];
    }
}

void Cls::init_arr()
{
    arr = new int[length];
}

void Cls::input()
{
    for(int i = 0; i < length; i++)
```

```

        {
            cin >> arr[i];
        }
    }

    void Cls::output()
    {
        cout << endl;
        for(int i = 0; i < length; i++)
        {
            cout << arr[i];
            if(i != length - 1) cout << "    ";
        }
    }

    int Cls::m1()
    {
        for(int i = 0; i < length; i+=2)
        {
            arr[i] = arr[i] + arr[i + 1];
        }
        return m3();
    }
    int Cls::m2()
    {
        for(int i = 0; i < length; i+=2)
        {
            arr[i] = arr[i] * arr[i + 1];
        }
        return m3();
    }
    int Cls::m3()
    {
        int sum = 0;
        for(int i = 0; i < length; i++)
        {
            sum += arr[i];
        }
        return sum;
    }
}

```

## 5.2 Файл Cls.h

*Листинг 2 – Cls.h*

```

#ifndef __CLS__H
#define __CLS__H

class Cls

```

```

{
private:
    int* arr = nullptr;
    int length;
public:
    Cls();
    Cls(int n);
    ~Cls();
    Cls(const Cls &ob);
    void init_arr();
    void input();
    void output();
    int m1();
    int m2();
    int m3();
};

#endif

```

### 5.3 Файл main.cpp

*Листинг 3 – main.cpp*

```

#include <stdlib.h>
#include <stdio.h>
#include "Cls.h"
#include <iostream>

using namespace std;

Cls func(int n){
    Cls obj_loc(n);
    return obj_loc;
}

int main()
{
    int n;
    cin >> n;
    if(n % 2 != 0 || n <= 2){
        cout << n << "?";
        return(0);
    }

    cout << n;

    Cls obj1;
    obj1 = func(n);
    obj1.init_arr();
    obj1.input();
}

```



```
    obj1.m2();

    Cls obj2(obj1);
    obj2.m1();

    obj1.output();
    cout << endl << obj1.m3();
    obj2.output();
    cout << endl << obj2.m3();

    // program here
    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
3	3?	3?

## 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).