

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	7
1.2 Описание выходных данных.....	8
2 МЕТОД РЕШЕНИЯ.....	10
3 ОПИСАНИЕ АЛГОРИТМОВ.....	11
3.1 Алгоритм метода get_ptr класса Cls.....	11
3.2 Алгоритм метода set_ptr класса Cls.....	11
3.3 Алгоритм функции main.....	12
3.4 Алгоритм функции func.....	13
3.5 Алгоритм метода input класса Cls.....	13
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	15
5 КОД ПРОГРАММЫ.....	17
5.1 Файл Cls.cpp.....	17
5.2 Файл Cls.h.....	19
5.3 Файл main.cpp.....	19
6 ТЕСТИРОВАНИЕ.....	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	22

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 значений элементов 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 двумя пробелами;

- метод, который возвращает значение указателя на массив из закрытой области;
- метод, который присваивает значение указателя массива из закрытой области.

Назовём класс описания данного объекта `cl_obj` (для примера, у вас он может называться иначе).

Разработать функцию `func`, которая имеет один целочисленный параметр, содержащий размерность массива. В функции должен быть реализован алгоритм:

1. Инициализация указателя на объект класса `cl_obj` адресом объекта, созданного с использованием параметризованного конструктора.
2. С использованием указателя на объект класса `cl_obj` вызов метода создания массива.
3. С использованием указателя на объект класса `cl_obj` вызов метода ввода значений элементов массива.
4. С использованием указателя на объект класса `cl_obj` вызов метода 2.
5. Возврат указателя на объект класса `cl_obj`.

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. Объявить первый указатель на объект класса `cl_obj`.
5. Присвоение первому указателю результата работы функции `func` с аргументом, содержащим значение размерности массива.
6. С использованием первого указателя вызов метода 1.
7. Инициализация второго указателя на объект класса `cl_obj` адресом

объекта, созданного с использованием конструктора копии с аргументом первого объекта.

8. С использованием второго указателя вызов метода 2.
9. Вывод содержимого массива первого объекта.
10. Вывод суммы элементов массива первого объекта.
11. Вывод содержимого массива второго объекта.
12. Вывод суммы элементов массива второго объекта.
13. Второму объекту присвоить первый объект.
14. С использованием первого указателя вызов метода 1.
15. Вывод содержимого массива второго объекта.
16. Вывод суммы элементов массива второго объекта.
17. Удалит первый объект.
18. Удалить второй объект.

Добавить в этот алгоритм пункты, которые обеспечат корректное завершение работы программы.

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
Constructor set
Copy constructor
20 5 4 2
31
100 5 8 2
```

```
115
100 5 8 2
115
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 для решения задачи.

Класс Cls:

- функционал:
 - метод get_ptr — Получение указателя на arr;
 - метод set_ptr — Установка указателя arr;
 - метод input — Метод ввода данных.

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 `get_ptr` класса `Cls`

Функционал: Получение указателя на арг.

Параметры: нет.

Возвращаемое значение: `int*`.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода `get_ptr` класса `Cls`

№	Предикат	Действия	№ перехода
1		Возврат указателя на арг	Ø

3.2 Алгоритм метода `set_ptr` класса `Cls`

Функционал: Установка указателя арг.

Параметры: `int* parg` - новый указатель.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода `set_ptr` класса `Cls`

№	Предикат	Действия	№ перехода
1		Присваивание указателю арг указателя parg	Ø

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 переменной n типа int	2
2		Ввод значения n	3
3	$n \% 2 \neq 0 \parallel n \leq 2$	Вывод n?	20
			4
4		Вывод n	5
5		Создание указателя obj1 на класс Cls	6
6		Присваивание obj1 значение результата функции func с параметром n	7
7		Вызов метода m1 у объекта по указателю obj1	8
8		Создание указателя obj на класс Cls и передачей объекта указателя obj1 как параметра	9
9		Вызов метода m2 у объекта по указателю obj2	10
10		Вызов метода output у объекта по указателю obj1	11
11		Вывод значения вызова метода m3 у объекта по указателю obj1	12
12		Вызов метода output у объекта по указателю obj2	13
13		Вывод значения вызова метода m3 у объекта по указателю obj2	14
14		Присвоение объекта указателя obj2 объекту указателя obj1	15
15		Вызов метода m2 у объекта по указателю obj1	16

№	Предикат	Действия	№ перехода
16		Вызов метода output у объекта по указателю obj2	17
17		Вывод значения вызова метода m3 у объекта по указателю obj2	18
18		Удаление obj1	19
19		Удаление obj2	20
20		Возврат значения 0	∅

3.4 Алгоритм функции func

Функционал: Инициализация объекта класса Cls.

Параметры: int n - размер массива для объекта класса Cls.

Возвращаемое значение: Cls* - указатель на объекта класса Cls.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции func

№	Предикат	Действия	№ перехода
1		Создание указателя obj на класс Cls и передачей n как параметра	2
2		Вызов метода init_arg у объекта по указателю obj	3
3		Вызов метода input у объекта по указателю obj	4
4		Вызов метода m2 у объекта по указателю obj	5
5		Возврат obj	∅

3.5 Алгоритм метода input класса Cls

Функционал: Метод ввода данных.

Параметры: нет.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода *input* класса *Cls*

№	Предикат	Действия	№ перехода
1		Инициализация $i = 0$ типа <code>int</code>	2
2	$i < \text{length}$	Ввод значения <code>arr[i]</code>	3
			Ø
3		$i++$	1

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

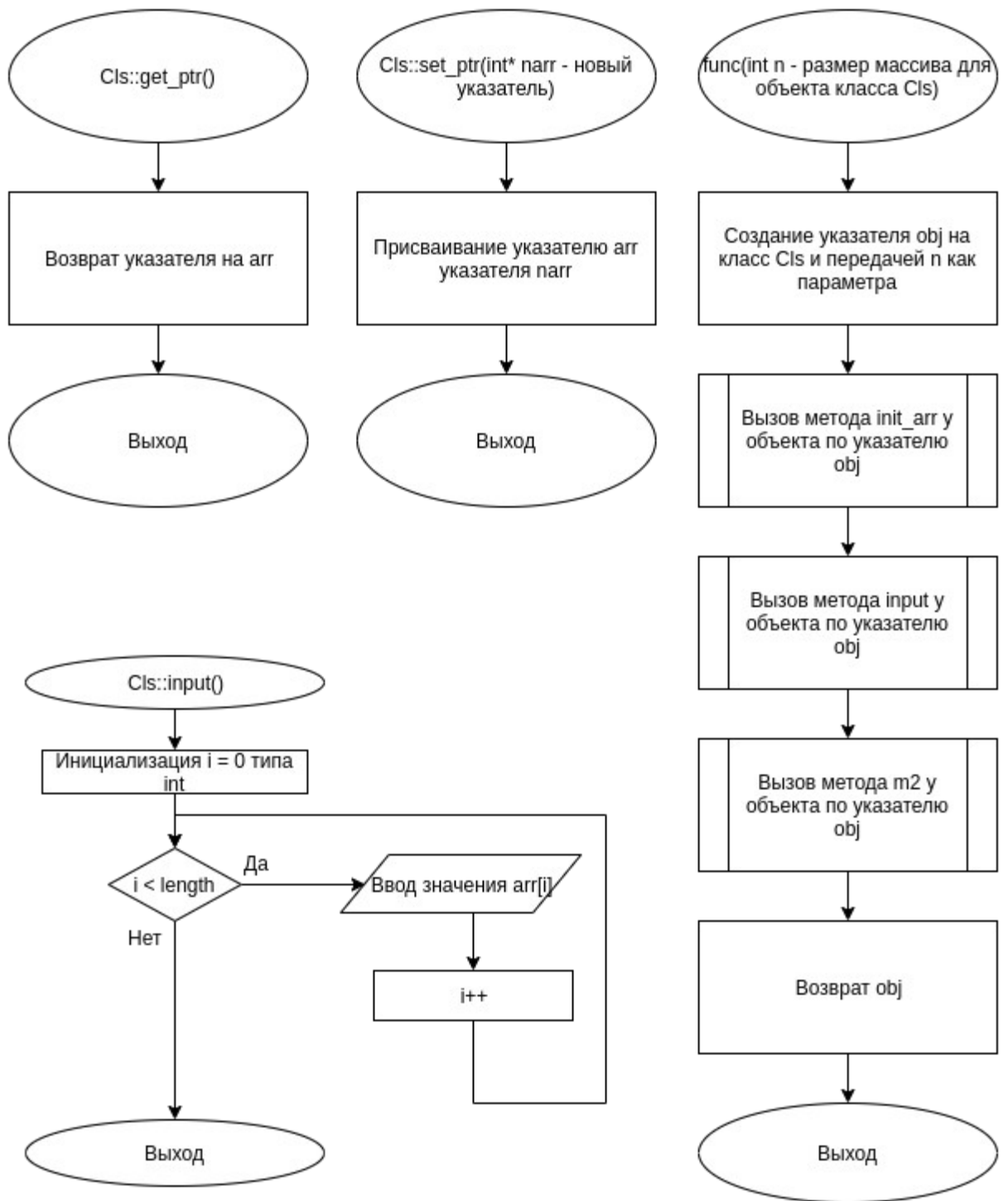


Рисунок 1 – Блок-схема алгоритма

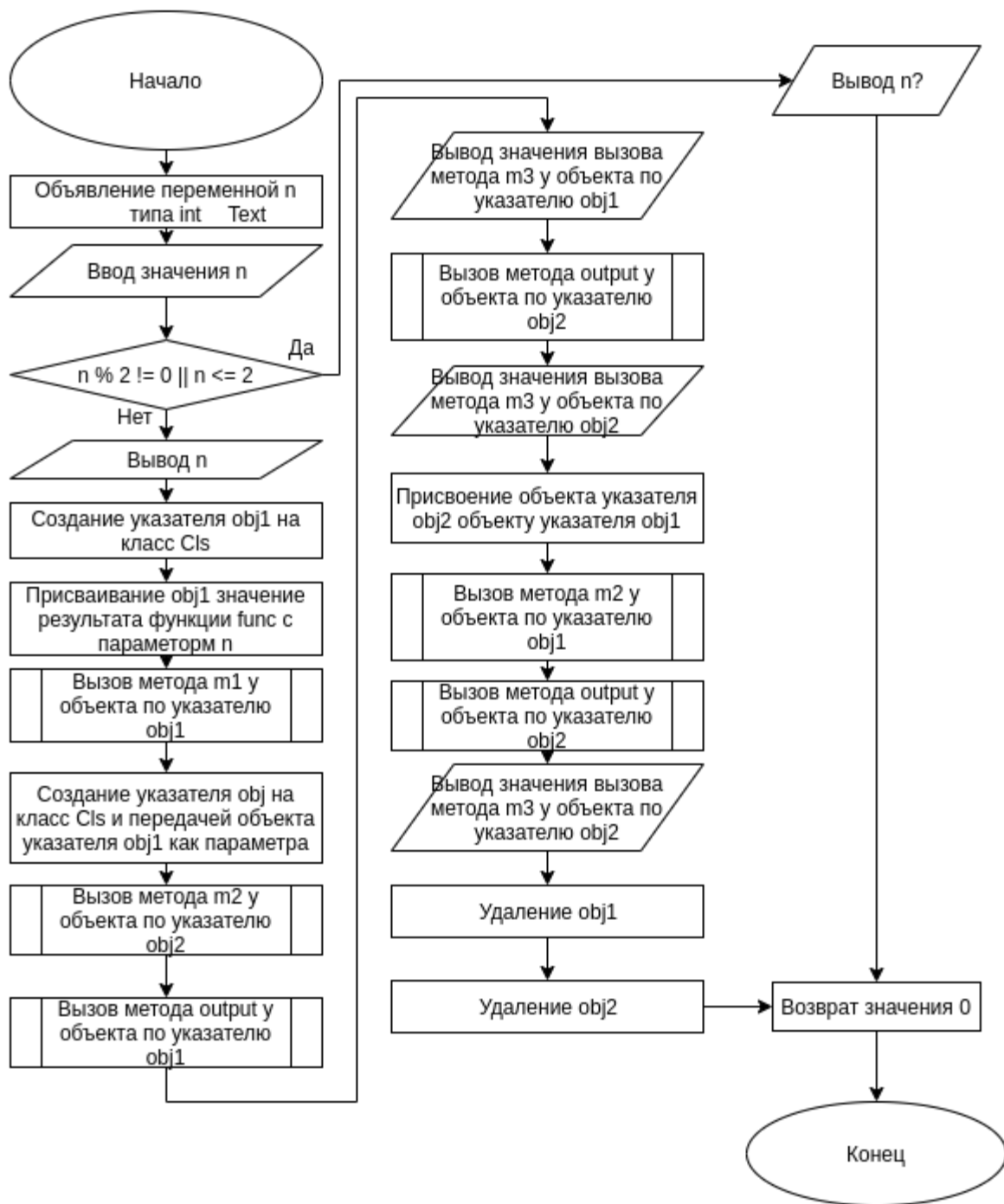


Рисунок 2 – Блок-схема алгоритма

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";
}
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 << " ";
    }
}

void Cls::m1()
{
    for(int i = 0; i < length; i+=2)
    {
        arr[i] = arr[i] + arr[i + 1];
    }
}

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

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

int* Cls::get_ptr(){
    return arr;
}

void Cls::set_ptr(int* narr){
    arr = narr;
}

```


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();
    void m1();
    void m2();
    int m3();
    int* get_ptr();
    void set_ptr(int* arr);
};

#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 = new Cls(n);

    obj->init_arr();
    obj->input();
    obj->m2();

    return obj;
}
```

```

}

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

    cout << n;

    Cls* obj1;
    obj1 = func(n);
    obj1->m1();

    Cls* obj2 = new Cls(*obj1);
    obj2->m2();

    obj1->output();
    cout << endl << obj1->m3();
    obj2->output();
    cout << endl << obj2->m3();

    *obj2 = *obj1;

    obj1->m2();
    obj2->output();
    cout << endl << obj2->m3();

    delete obj1;
    delete obj2;
    // 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 Constructor set Copy constructor 20 5 4 2 31 100 5 8 2 115 100 5 8 2 115 Destructor Destructor	4 Constructor set Copy constructor 20 5 4 2 31 100 5 8 2 115 100 5 8 2 115 Destructor Destructor
5	5?	5?

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 (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 (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).