

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 функции main.....	11
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	14
5.1 Файл main.cpp.....	14
5.2 Файл obj.cpp.....	15
5.3 Файл obj.h.....	17
6 ТЕСТИРОВАНИЕ.....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19

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 obj предназначен для ;
- объект obj2 класса obj предназначен для ;
- функция main для Основная;
- функция fun для ;
- Условный оператор;
- Оператор со счетчиком;
- Объект стандартного потока ввода/вывода cin/cout.

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 функции `main`

Функционал: Основная функция.

Параметры: Отсутствуют.

Возвращаемое значение: Целое.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции `main`

№	Предикат	Действия	№ перехода
1		Ввод размерности массива	2
2		Если размерность массива некорректная, вывод сообщения и завершить работу алгоритма.	3
3		Вывод значения размерности массива.	4
4		Объявить первый указатель на объект класса <code>cl_obj</code> .	5
5		Присвоение первому указателю результата работы функции <code>func</code> с аргументом, содержащим значение размерности массива.	6
6		С использованием первого указателя вызов метода 1.	7
7		Инициализация второго указателя на объект класса <code>cl_obj</code> адресом объекта, созданного с использованием конструктора копии с аргументом первого объекта.	8
8		С использованием второго указателя вызов метода 2.	9
9		Вывод содержимого массива первого объекта.	10
10		Вывод суммы элементов массива первого объекта.	11
11		Вывод содержимого массива второго объекта.	12

№	Предикат	Действия	№ перехода
12		Вывод суммы элементов массива второго объекта.	13
13		Второму объекту присвоить первый объект.	14
14		С использованием первого указателя вызов метода 1.	15
15		Вывод содержимого массива второго объекта.	16
16		Вывод суммы элементов массива второго объекта.	17
17		Удалит первый объект.	18
18		Удалит второй объект.	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

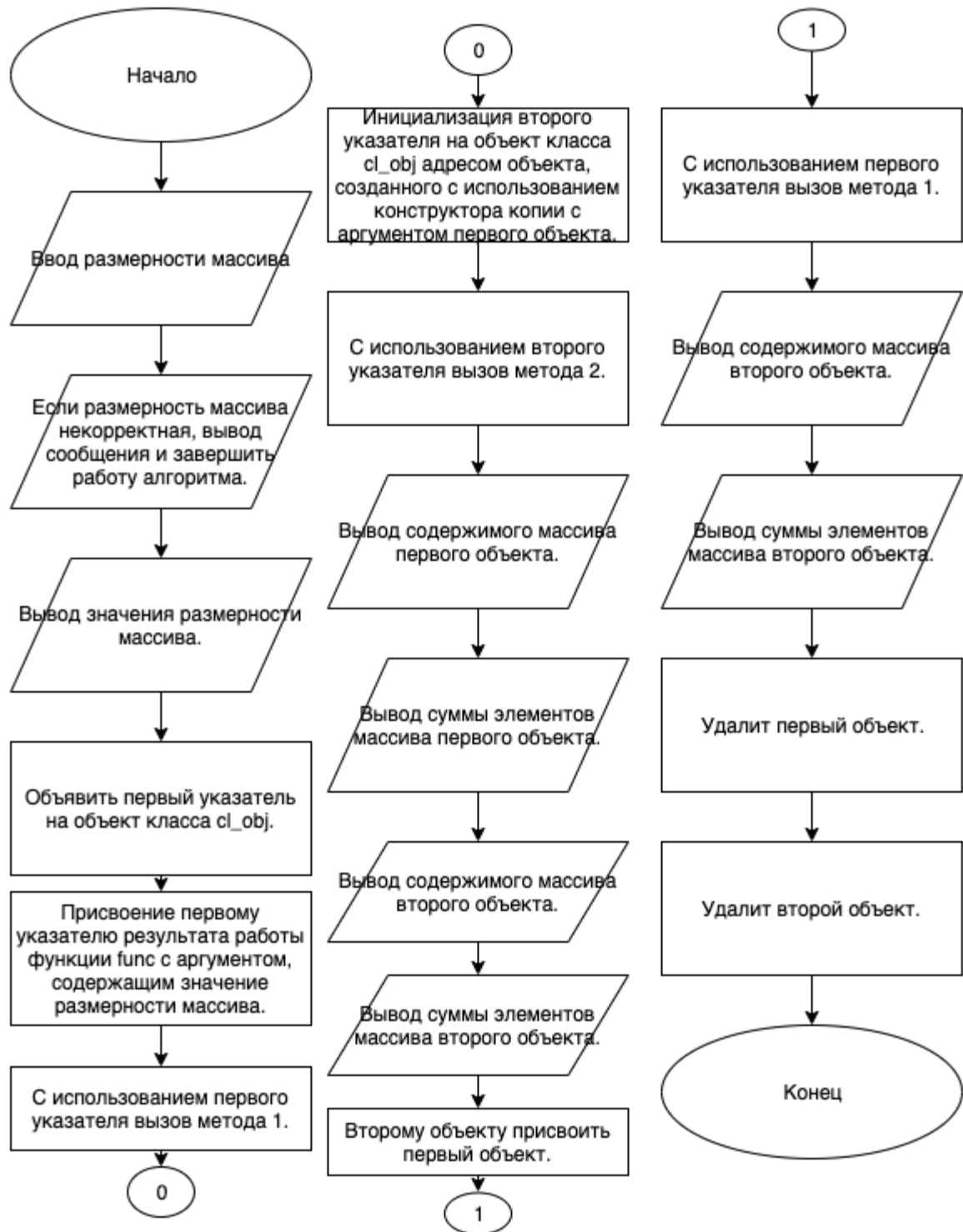


Рисунок 1 – Блок-схема алгоритма

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 "obj.h"

using namespace std;
obj* fun(int size)
{
    obj* cl_obj = new obj(size);
    cl_obj->Array();
    cl_obj->Fill();
    cl_obj->Metod2();
    return cl_obj;
}
int main()
{
    int size_arr;
    cin >> size_arr;
    if(size_arr > 2 && size_arr %2 == 0)
    {
        cout << size_arr << endl;
        obj* obj1;
        obj1 = fun(size_arr);
        obj1->Metod1();
        obj* obj2 = new obj(*obj1);
        obj2->Metod2();
        obj1->Vivod();
        cout << obj1->Sum() << endl;
        obj2->Vivod();
        cout << obj2->Sum() << endl;
        *obj2 = *obj1;
        obj1->Metod2();
        obj2->Vivod();
        cout << obj2->Sum();
        delete obj1;
        delete obj2;
    }
    else
    {
```

```

        cout << size_arr << "?";
    }
    return(0);
}

```

5.2 Файл obj.cpp

Листинг 2 – obj.cpp

```

#include "obj.h"
#include <iostream>
using namespace std;
obj::obj()
{
    cout << "Default constructor" << endl;
}

obj::obj(int size_arr)
{
    cout << "Constructor set";
    this->size_arr = size_arr;
}
obj::obj(const obj & obj)
{
    cout << "\nCopy constructor" << endl;
    size_arr = obj.size_arr;
    arr = new int[size_arr];
    for(int i = 0; i < size_arr; i++)
    {
        arr[i] = obj.arr[i];
    }
}
obj::~~obj()
{
    cout << "\nDestructor";
}
void obj::Array()
{
    arr = new int[size_arr];
}
void obj::Fill()
{
    for(int i = 0; i < size_arr; i++)
    {
        cin >> arr[i];
    }
}

int obj::Sum()
{

```

```

        int sum = 0;
        for(int i = 0; i < size_arr; i++)
        {
            sum += arr[i];
        }
        return sum;
    }

    void obj::Metod1()
    {
        for(int i = 0; i < size_arr; i+=2)
        {
            arr[i] += arr[i+1];
        }
    }

    void obj::Metod2()
    {
        for(int i = 0; i < size_arr; i+=2)
        {
            arr[i] *= arr[i+1];
        }
    }

    void obj::Vivod()
    {
        for(int i = 0; i < size_arr; i++)
        {
            if(i == size_arr -1)
            {
                cout << arr[i];
            }
            else
            {
                cout << arr[i] << " ";
            }
        }
        cout << endl;
    }

    int* obj::Get()
    {
        return arr;
    }

    void obj::Set(int* arr)
    {
        this->arr = arr;
    }

```

5.3 Файл obj.h

Листинг 3 – obj.h

```
#ifndef __IKB051__H
#define __IKB051__H

class obj
{
private:
    int* arr;
    int size_arr;
public:
    obj();
    obj(int size_arr);
    obj(const obj & obj_);
    ~obj();
    void Fill();
    void Array();
    int* Get();
    void Set(int*);
    int Sum();
    void Metod1();
    void Metod2();
    void Vivod();
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 2.

Таблица 2 – 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

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).