

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 функции fun.....	10
3.2 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	14
5.1 Файл main.cpp.....	14
5.2 Файл obj.cpp.....	15
5.3 Файл obj.h.....	16
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 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 obj предназначен для ;
- объект obj2 класса obj предназначен для ;
- функция fun для Создание и возврат локального объекта;
- функция main для Основная функция;
- Условный оператор;
- Цикл со счётчиком;
- 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 fun

Функционал: Создание и возврат локального объекта.

Параметры: size.

Возвращаемое значение: Объект.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции fun

№	Предикат	Действия	№ перехода
1		Создадим локальный объекта newobj	2
2		Возврат локального объекта newobj	Ø

3.2 Алгоритм функции main

Функционал: Основная функция.

Параметры: отсутствуют.

Возвращаемое значение: Целочисленное.

Алгоритм функции представлен в таблице 2.

Таблица 2 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной size_arr	2
2		Ввод size_arr	3
3	size_arr > 2 или size_arr	Вывод значения size_arr	4

№	Предикат	Действия	№ перехода
	четное		
		Вывод size_arg и "?"	∅
4		Создание объекта obj1 класса obj	5
5		Присвоение obj1 результат работы функции fun	6
6		Вызов метода Array() объекта obj1	7
7		Вызов метода Fill() объекта obj1	8
8		Вызов метода Metod2()	9
9		Инициализация второго объекта первым объектом	10
10		Вызов метода Metod1() объекта obj2	11
11		Вывод содержимого массива первого объекта	12
12		Вызов метода Sum объекта obj1	13
13		Вывод содержимого массива второго объекта	14
14		Вызов метода Sum объекта obj2	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

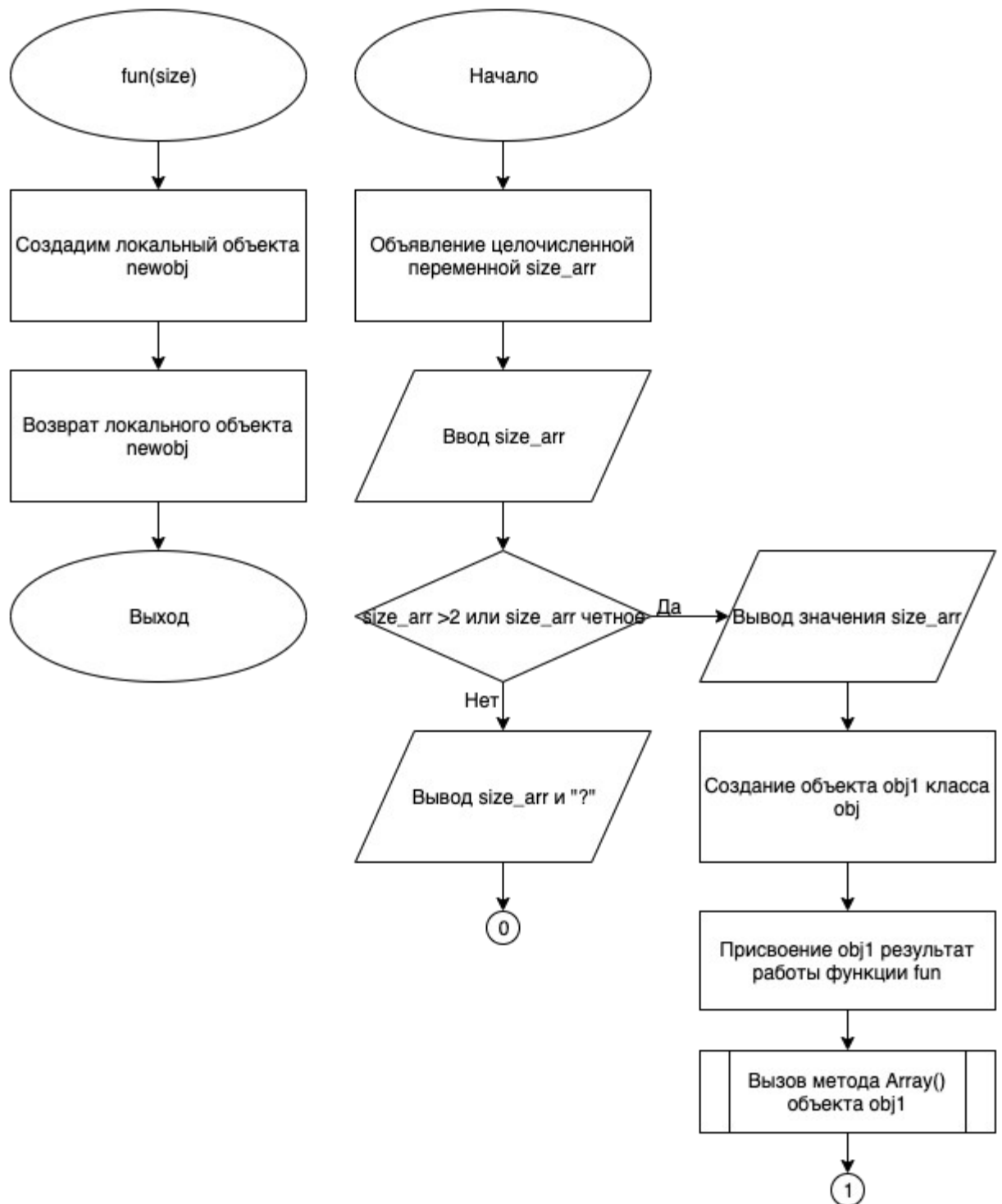


Рисунок 1 – Блок-схема алгоритма

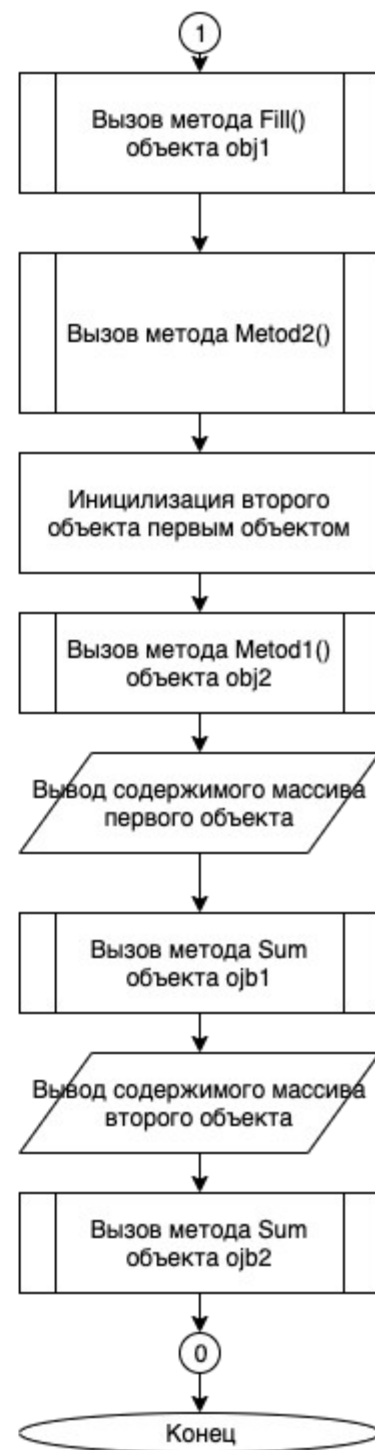


Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл 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 newobj1(size);
    return newobj1;
}
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.Array();
        obj1.Fill();
        obj1.Metod2();
        obj obj2(obj1);
        obj2.Metod1();
        obj1.Vivod();
        cout << endl;
        cout << obj1.Sum() << endl;
        obj2.Vivod();
        cout << endl;
        cout << obj2.Sum();
    }
    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";
    arr = new int[size_arr];
    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";
    if(arr == nullptr)
    {
        delete[] arr;
    }
}
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] << " ";
        }
    }
}

```

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 Sum();  
    void Metod1();  
    void Metod2();  
    void Vivod();  
};  
  
#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 3.

Таблица 3 – 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

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).