

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 <code>vivod</code> класса <code>MyClass</code>	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	12
5.1 Файл <code>main.cpp</code>	12
5.2 Файл <code>MyClass.cpp</code>	13
5.3 Файл <code>MyClass.h</code>	14
6 ТЕСТИРОВАНИЕ.....	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	16

1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В закрытом доступе имеется массив целого типа. 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

Метод решения как в задаче 3_1_3

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 `vivod` класса `MyClass`

Функционал: Вывод массива.

Параметры: Отсутствуют.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода `vivod` класса `MyClass`

№	Предикат	Действия	№ перехода
1	<code>i < size</code>		2
			∅
2		Вывод <code>arr[i]</code>	3
3		<code>i++</code>	1

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

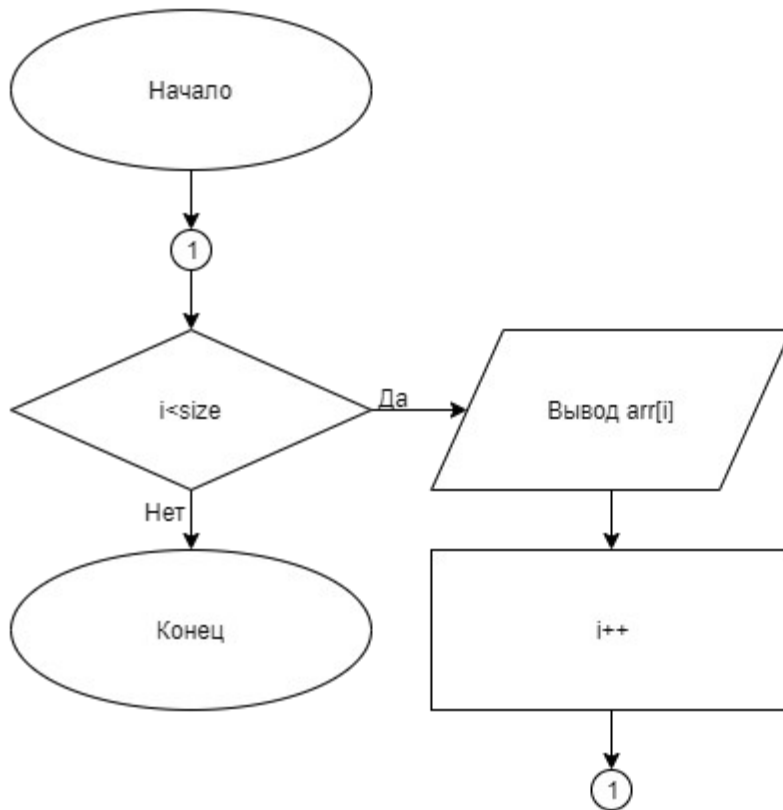


Рисунок 1 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include "MyClass.h"
#include<iostream>
using namespace std;

void func(MyClass obj)
{
    obj.method1();
    obj.vivod();
    obj.sum();
    obj.method2();
    obj.vivod();
    obj.sum();
}

int main()
{
    int size;
    cin>>size;
    if ((size<=2) ||(size%2!=0))
    {
        cout<<size<<"?";
        return 0;
    }
    cout<<size<<endl;
    MyClass *obj_=new MyClass;
    MyClass obj(size);
    delete (obj_);
    cout<<endl;
    obj.input();
    func(obj);
    cout<<endl;
    return(0);
}
```

5.2 Файл MyClass.cpp

Листинг 2 – MyClass.cpp

```
#include "MyClass.h"
#include<iostream>
#include<string>
using namespace std;

MyClass::MyClass()
{
    cout<<"Default constructor"<<endl;
}
MyClass::MyClass(int size)
{
    arr=new int [size];
    cout<<"Constructor set"<<endl;
    this->size=size;
}
MyClass::MyClass(const MyClass & obj)
{
    cout<<"Copy constructor"<<endl;
    size=obj.size;
    arr=new int[size];
    for(int i=0;i<size;i++)
        arr[i]=obj.arr[i];
}
void MyClass::input()
{
    int n;
    for(int i=0;i<size;i++)
    {
        cin>>n;
        arr[i]=n;
    }
}
void MyClass::vivot()
{
    for (int i=0;i<size-1;i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<arr[size-1];
    cout<<endl;
}
MyClass::~MyClass()
{
    cout<<"Destructor";
    if (arr!=nullptr)
        delete[]arr;
}
void MyClass::method1()
{
    for (int i=0;i<size;i=i+2)
    {
```

```

        arr[i]=arr[i]*arr[i+1];
    }
}
void MyClass::method2()
{
    for (int i=0;i<size;i=i+2)
    {
        arr[i]=arr[i]+arr[i+1];
    }
}
void MyClass::sum()
{
    int s=0;
    for (int i=0;i<size;i++)
    {
        s=s+arr[i];
    }
    cout<<s<<endl;
}

```

5.3 Файл MyClass.h

Листинг 3 – MyClass.h

```

#ifndef __MYCLASS_H__
#define __MYCLASS_H__
#include<iostream>
using namespace std;

class MyClass
{
private:
    int* arr;
    int size;
public:
    ~MyClass();
    MyClass();
    MyClass(int size);
    MyClass(const MyClass& obj);
    void input();
    void vivod();
    void sum();
    void method1();
    void method2();
};
#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 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
1	1?	1?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.
3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodicheskoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).
4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).
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).