

Здесь будет титульник, листай ниже

# СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	6
1.2 Описание выходных данных.....	6
2 МЕТОД РЕШЕНИЯ.....	7
3 ОПИСАНИЕ АЛГОРИТМОВ.....	8
3.0 Алгоритм функции main.....	8
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	9
5 КОД ПРОГРАММЫ.....	10
5.0 Файл main.cpp.....	10
5.1 Файл MyClass.cpp.....	11
5.2 Файл MyClass.h.....	12
6 ТЕСТИРОВАНИЕ.....	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	14

# 1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В закрытом доступе имеется указатель на массив целого типа.

Конструктору объекта передается целочисленный параметр. Параметр должен иметь значение больше 4. По значению параметра определяется размерность целочисленного массива из закрытой области и каждому элементу присваивается это же значение.

Объект имеет функциональность, по которой выводит содержимое целочисленного массива. Вывод производит последовательно, разделяя значения двумя пробелами.

Функциональность объекта можно расширить по усмотрению разработчика не более чем на два метода.

Спроектировать систему, которая содержит два объекта. Для построения системы последовательно, с новых строк вводятся целочисленные значения. Если значение меньше или равно 4, то создание системы прекращается и выводится сообщение. Если система построена, то посредством параметризованного конструктора создаются объекты.

Далее система функционирует по алгоритму:

1. . . . .
2. Первому объекту присвоить второй объект.
3. . . . .
4. С первой строки вывести содержимое массива первого объекта.
5. . . . .
6. Со второй строки вывести содержимое массива второго объекта.

## 1.1 Описание входных данных

Первая строка:

«Целое число»

Вторая строка:

«Целое число»

Пример.

5

8

## 1.2 Описание выходных данных

Если система была построена, то в первой строке:

«Целое число» «Целое число» . . .

Во второй строке:

«Целое число» «Целое число» . . .

Если система не была построена, то в первой строке выводится некорректное значение и вопросительный знак:

«Целое число»?

Пример вывода.

5 5 5 5 5

8 8 8 8 8 8 8 8

## **2 МЕТОД РЕШЕНИЯ**

Метод решения схож с предыдущей задачей

## 3 ОПИСАНИЕ АЛГОРИТМОВ

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

### 3.0 Алгоритм функции main

Функционал: основная функция.

Параметры: отсутствуют.

Возвращаемое значение: целое признак успеха.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Алгоритм схож с предыдущей задачей 3_1_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.

### 5.0 Файл main.cpp

Листинг 1 – main.cpp

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

int main()
{
    int size;
    cin>>size;
    MyClass *obj1=new MyClass(size);
    if (size<=4)
    {
        cout<<size<<"?";
        return 0;
    }
    cin>>size;
    MyClass *obj2=new MyClass(size);
    if (size<=4)
    {
        cout<<size<<"?";
        return 0;
    }
    int *temp=obj1->arr;
    int a=obj1->n;
    obj1=obj2;
    for (int i=1;i<=a;i++)
    {
        if (i!=a)
            cout<<temp[i-1]<<" ";
        else
            cout<<temp[i-1];
    }
    cout<<endl;
    obj1->print();
    //obj1.operator=(obj2);
    return(0);
}
```



## 5.1 Файл MyClass.cpp

Листинг 2 – MyClass.cpp

```
#include "MyClass.h"
#include <iostream>
using namespace std;
MyClass::MyClass(int size)
{
    n=size;
    //if (size>4)
    //{
        arr=new int[size];
        for (int i=0;i<size;i++)
        {
            arr[i]=size;
        }
    //}
    //else
        //cout<<size;
}
/*MyClass::~MyClass()
{
    if(arr!=nullptr)
        delete[]arr;
}*/
void MyClass::print()
{
    for(int i=1;i<=n;i++)
    {
        if (i!=n)
            cout<<arr[i-1]<<" ";
        else
            cout<<arr[i-1];
    }
}
/*MyClass& MyClass::operator=(const MyClass& obj)
{
    if(arr!=nullptr)
        delete[] arr;
    int size=obj.arr[0];
    arr=new int[size];
    for(int i=0;i<size;i++)
        arr[i]=obj.arr[i];
    return(*this);
}*/
```

## 5.2 Файл MyClass.h

*Листинг 3 – MyClass.h*

```
#ifndef __MYCLASS_H__
#define __MYCLASS_H__
class MyClass
{
public:
    int *arr;
    int n;
    MyClass(int size);
    void print();
    //~MyClass();
    //MyClass &operator=(const MyClass& obj);
};
#endif
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 2.

Таблица 2 – Результат тестирования программы

<b>Входные данные</b>	<b>Ожидаемые выходные данные</b>	<b>Фактические выходные данные</b>
2 1	2?	2?
3 1	3?	3?

## 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](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](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).