

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
3.1 Алгоритм функции main.....	8
3.2 Алгоритм конструктора класса Cocs1.....	9
3.3 Алгоритм деструктора класса Cocs1.....	9
3.4 Алгоритм метода Output класса Cocs1.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	14
5.1 Файл Cocs1.cpp.....	14
5.2 Файл Cocs1.h.....	14
5.3 Файл main.cpp.....	15
6 ТЕСТИРОВАНИЕ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17

1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В открытом 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 и каждому элементу присваивается это же значение.

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

Спроектировать систему, которая содержит два объекта. Для построения системы последовательно, с новых строк вводятся целочисленные значения. Если очередное введенное значение меньше или равно 4, то создание системы прекращается, выводится сообщение и программа завершает работу. Если очередное введенное значение больше 4, то посредством параметризованного конструктора создается первый объект, а затем, второй.

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

Для решения задачи используется:

- функция `main` для Главная функция программы;
- библиотека `iostream`;
- пространство имён `std`;
- оператор цикла со счётчиком `for`;
- условный оператор `if`;
- объект `cout` стандартного потока вывода данных;
- объект `cout` стандартного потока ввода данных.

Класс `Cocs1`:

- свойства/поля:
 - поле `Отвечает` за объявление указателя на динамический массив:
 - наименование — `a`;
 - тип — `int`;
 - модификатор доступа — `public`;
- функционал:
 - метод `Cocs1` — параметизированный конструктор для создания объекта класса `Cocs`;
 - метод `~Cocs1` — деструктор для удаления объекта класса `Cocs`;
 - метод `Output` — вывод всех элементов массива.

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: Главная функция программы.

Параметры: нет.

Возвращаемое значение: Целое, индикатор корректности завершения программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной data	2
2		Ввод значения переменной data	3
3		Создание объекта ob1 класса Cocs1	4
4		Ввод значения переменной data	5
5		Создание объекта ob2 класса Cocs1	6
6		Инициализация динамического массива b со значением поля a объекта ob1	7
7		Объекту ob1 присвоить значение объекта ob2	8
8		Полю a объекта ob1 присвоить значение переменной b	9
9		Вызов метода Output объекта ob1	10
10		Переход на новую строку	11
11		Вызов метода Output объекта ob2	Ø

3.2 Алгоритм конструктора класса Cocs1

Функционал: Создание объекта.

Параметры: int s - задание размера поля а класса Cocs1.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса Cocs1

№	Предикат	Действия	№ перехода
1	значение переменной s меньше или равно 4	Полю а присваивается значение нелевого указателя	2
		Выделение памяти под динамический массив а размера s с помощью оператора new	4
2		Вывод значения переменной s и "?"	3
3		Вызов метода ~Cocs1	∅
4	Целочисленная переменная i от 0 до s, шаг 1	i ячейке массива а присваивается значение переменной s	5
			∅
5		Инкремент i	4

3.3 Алгоритм деструктора класса Cocs1

Функционал: Удаления объекта класса Cocs.

Параметры: нет.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса Cocs1

№	Предикат	Действия	№ перехода
1	Значение поля а класса Cocs1 не равно значению нулевого указателя	Удаление динамического массива а и освобождение выделенной памяти с помощью оператора delete	∅

№	Предикат	Действия	№ перехода
			∅

3.4 Алгоритм метода Output класса Cocs1

Функционал: вывод всех элементов массива.

Параметры: нет.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода Output класса Cocs1

№	Предикат	Действия	№ перехода
1		Вывод первого элемента массива a	2
2	Целочисленная переменная i от 1 до первого элемента массива a, шаг 1	Вывод двух пробелов и i элемент массива a	3
			∅
3		Инкремент i	2

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-3.



Рисунок 1 – Блок-схема алгоритма

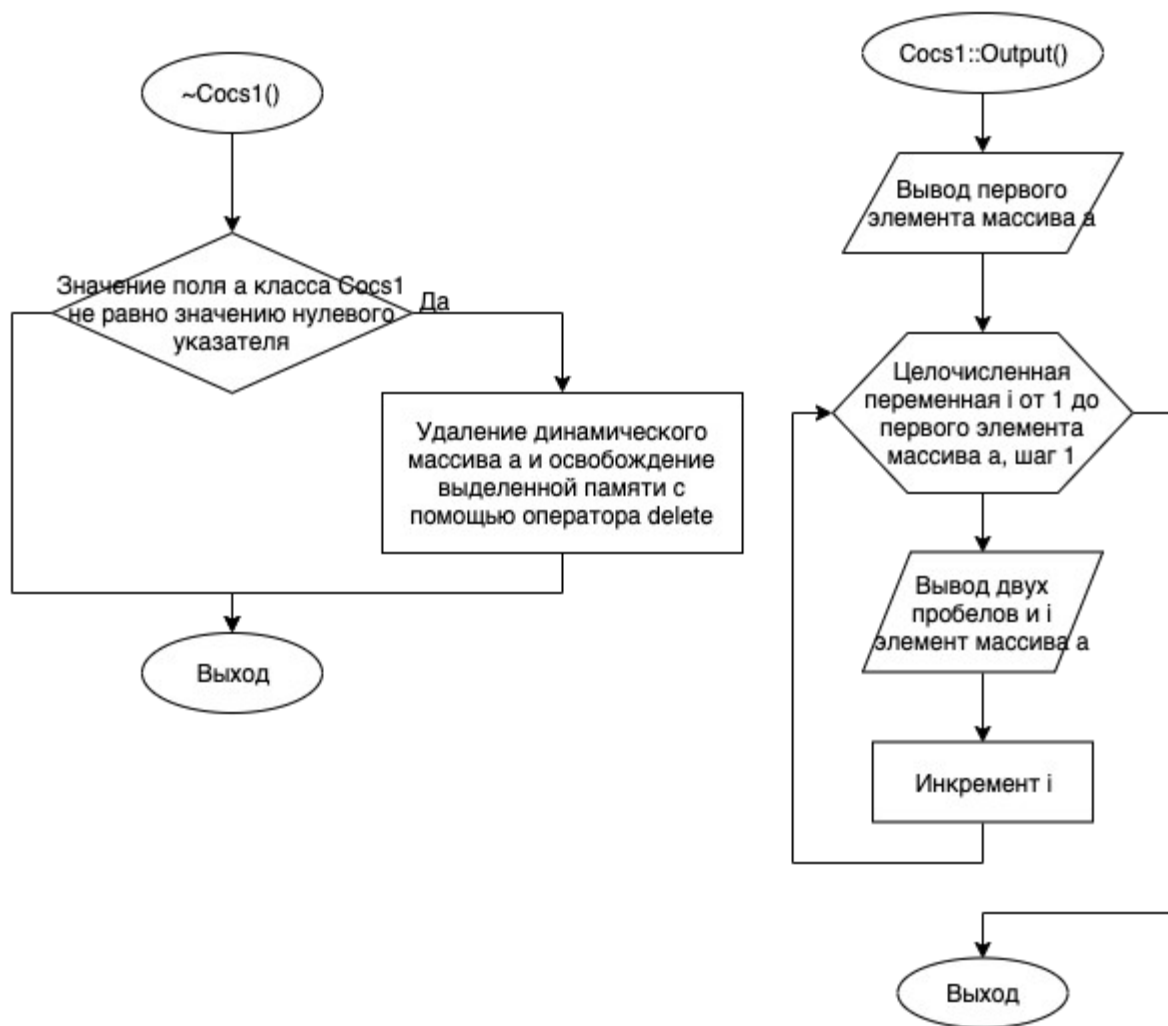


Рисунок 2 – Блок-схема алгоритма

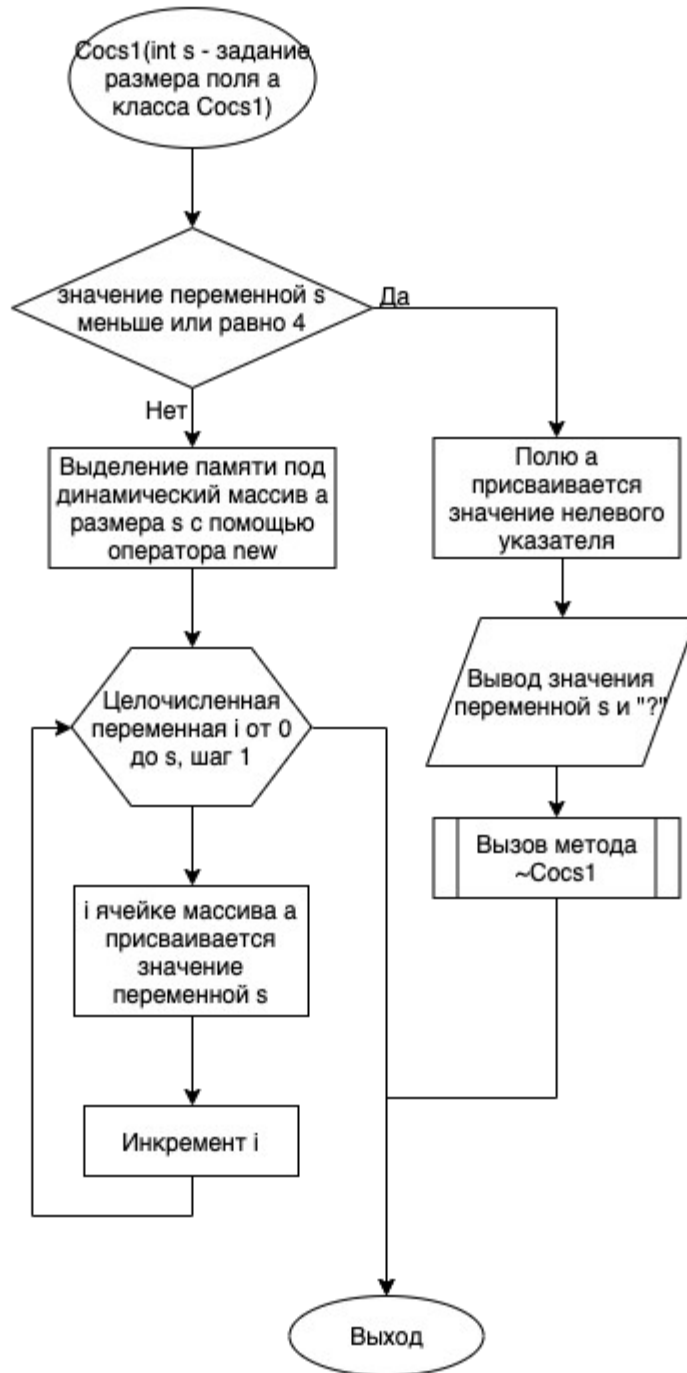


Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Cocs1.cpp

Листинг 1 – Cocs1.cpp

```
#include <iostream>
#include "Cocs1.h"
using namespace std;
Cocs1::Cocs1(int s)
{
    if (s<=4){
        a=nullptr;
        cout<<s<<"?";
        exit(0);
    }
    a=new int[s];
    for (int i=0;i<s;i++)
        a[i]=s;
}
void Cocs1::Output()
{
    cout<<a[0];
    for (int i=1;i<a[0];i++)
        cout<<" "<<a[i];
}
Cocs1::~Cocs1()
{
    if(a!=nullptr){
        delete[] a;
    }
}
```

5.2 Файл Cocs1.h

Листинг 2 – Cocs1.h

```
#ifndef __COCS__H
#define __COCS__H
class Cocs1
```

```
{
    public:
        int* a;
        Cocs1(int s);
        void Output();
        ~Cocs1();
};
#endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <iostream>
#include "Cocs1.h"
using namespace std;

int main()
{
    int data;
    cin>>data;
    Cocs1 ob1(data);
    cin>>data;
    Cocs1 ob2(data);
    int* b=ob1.a;
    ob1=ob2;
    ob1.a=b;
    ob1.Output();
    cout<<endl;
    ob2.Output();
    return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
2 9	2?	2?
0 5	0?	0?
6 7	6 6 6 6 6 6 7 7 7 7 7 7 7	6 6 6 6 6 6 7 7 7 7 7 7 7
5 7	5 5 5 5 5 7 7 7 7 7 7 7	5 5 5 5 5 7 7 7 7 7 7 7

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).