Здесь будет титульник, листай ниже

**СОДЕРЖАНИЕ**

[1 ПОСТАНОВКА ЗАДАЧИ 5](#__RefHeading___Toc2571_1148708417)

[1.1 Описание входных данных 6](#__RefHeading___Toc2573_1148708417)

[1.2 Описание выходных данных 6](#__RefHeading___Toc2575_1148708417)

[2 МЕТОД РЕШЕНИЯ 7](#__RefHeading___Toc2577_1148708417)

[3 ОПИСАНИЕ АЛГОРИТМОВ 8](#__RefHeading___Toc2579_1148708417)

[3.1 Алгоритм конструктора класса Arr 8](#__RefHeading___Toc2581_1148708417)

[3.2 Алгоритм метода coutArr класса Arr 8](#__RefHeading___Toc2583_1148708417)

[3.3 Алгоритм функции main 9](#__RefHeading___Toc2585_1148708417)

[3.4 Алгоритм деструктора класса Arr 10](#__RefHeading___Toc2587_1148708417)

[4 БЛОК-СХЕМЫ АЛГОРИТМОВ 11](#__RefHeading___Toc2589_1148708417)

[5 КОД ПРОГРАММЫ 15](#__RefHeading___Toc2591_1148708417)

[5.1 Файл Arr.cpp 15](#__RefHeading___Toc2593_1148708417)

[5.2 Файл Arr.h 15](#__RefHeading___Toc2595_1148708417)

[5.3 Файл main.cpp 16](#__RefHeading___Toc2597_1148708417)

[6 ТЕСТИРОВАНИЕ 17](#__RefHeading___Toc2599_1148708417)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#__RefHeading___Toc2601_1148708417)

# 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:

Объекты стандартного потокового ввода(cin) и вывода (cout);

Условный оператор if;

Оператор цикла for;

Класс Arr:

Поля/свойства:

Доступный указатель arr на объект целого типа;

Методы:

Конструктор Arr:

Функционал: 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.

Доступный метод coutArr:

Функционал: Выводит содержимое целочисленного массива. Вывод производит последовательно, разделяя значения двумя пробелами.

Деструктор ~Arr:

Функционал: Освобождение выделенной памяти под массив.

# 3 ОПИСАНИЕ АЛГОРИТМОВ

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

## Алгоритм конструктора класса Arr

Функционал: 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..

Параметры: Целочисленный i.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Arr

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 | Параметр i > 4 | Присвоение arr и выделение памяти на  целочисленный массив размера i | 2 |
|  |  | ∅ |
| 2 |  | Объявление и инициализация целой переменной j  нулем | 3 |
| 3 | Цикл j < i | arr[j] = i, увеличение j на 1 | 3 |
|  |  | ∅ |

## Алгоритм метода coutArr класса Arr

Функционал: Вывод элементов массива..

Параметры: Отсутствуют.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода coutArr класса Arr

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Объявление и инициализация целой переменной j  нулем | 2 |
| 2 | Цикл j < arr[0] | Вывод arr[j] | 3 |
|  |  | ∅ |
| 3 | j != arr[0] - 1 | Вывод двух пробелов | 2 |
|  |  | 2 |

## Алгоритм функции main

Функционал: Основной алгоритм программы.

Параметры: Отсутствуют.

Возвращаемое значение: Целое - индикатор корректности завершения программы.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции main

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Объявление целых переменных i, j и ввод их  значений с клавиатуры | 2 |
| 2 | i <= 4 | Вывод i с "?" | ∅ |
|  |  | 3 |
| 3 | j <= 4 | Вывод j с "?" | ∅ |
|  |  | 4 |
| 4 |  | Объявление и инициализация объекта класса Arr  ar1 и вызов конструктора с выделением памяти параметром i с | 5 |
| 5 |  | Объявление и инициализация объекта класса Arr  ar2 и вызов конструктора с параметром j с  выделением памяти | 6 |
| 6 |  | Создание указателя на объект целого типа tmp и  присваивание ему значение ar1.arr | 7 |
| 7 |  | Присваивание переменной ar1 значение  переменной ar2 | 8 |
| 8 |  | Присваивание полю arr объекта ar2 значение tmp | 9 |
| 9 |  | Вызов метода coutArr() объекта ar2 и вывод  перехода на новую строку | 10 |
| 10 |  | Вызов метода coutArr() объекта ar1 | ∅ |

## Алгоритм деструктора класса Arr

Функционал: Очищение выделенной памяти под динамический массив.

Параметры: Отсутствуют.

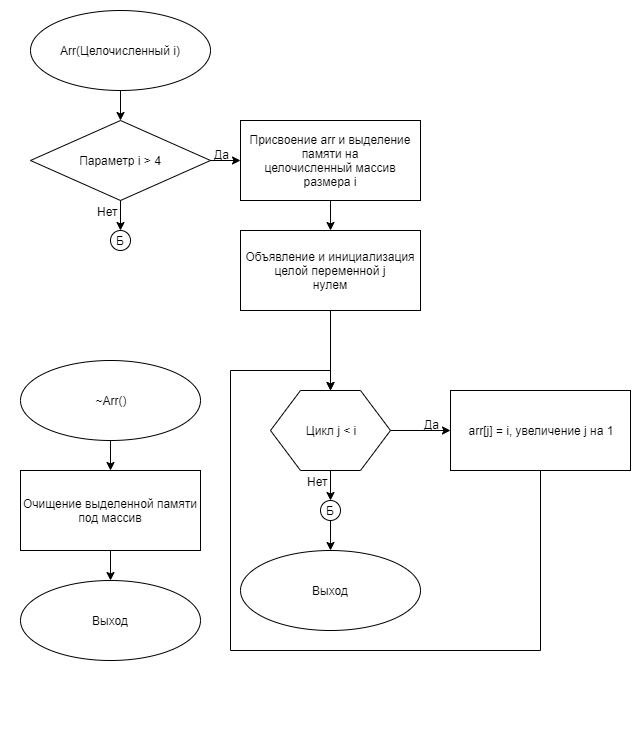
Алгоритм деструктора представлен в таблице 4.

Таблица 4 – Алгоритм деструктора класса Arr

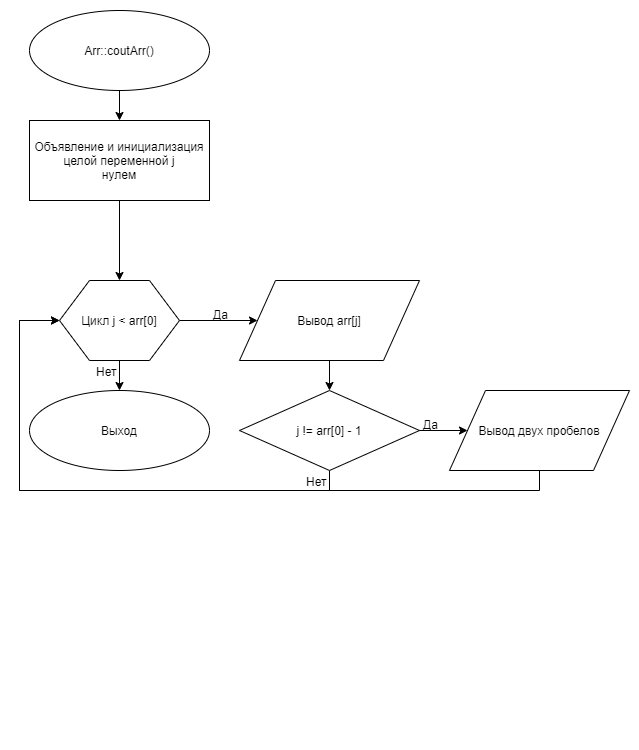
| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Очищение выделенной памяти под массив | ∅ |

# 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

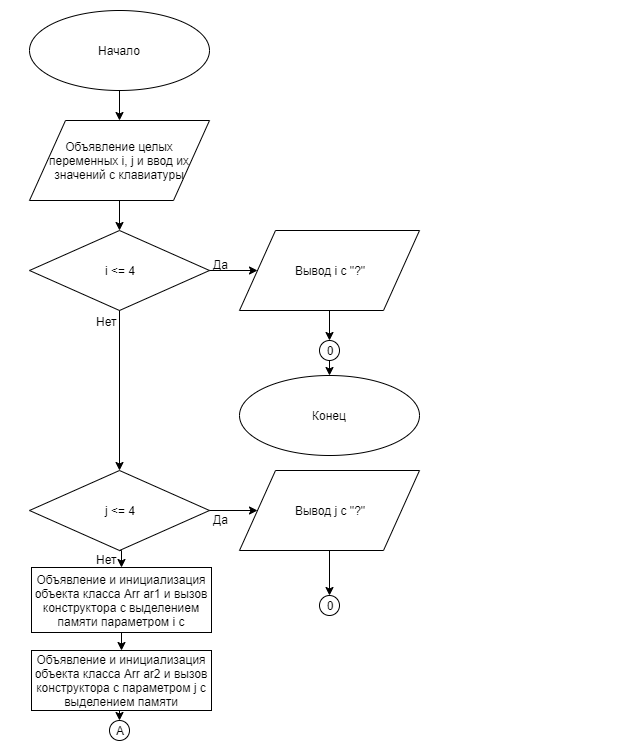
Представим описание алгоритмов в графическом виде на рисунках 1-4.



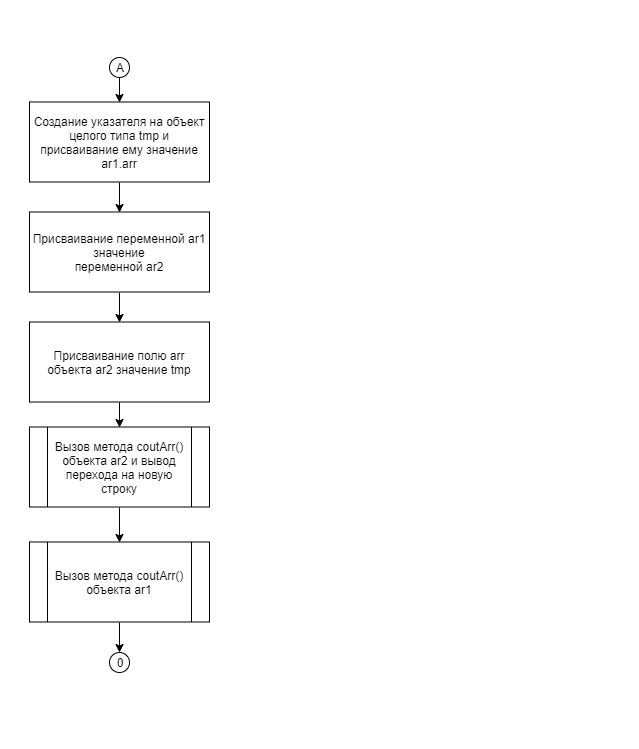
**Рисунок 1 – Блок-схема алгоритма**



**Рисунок 2 – Блок-схема алгоритма**



**Рисунок 3 – Блок-схема алгоритма**



**Рисунок 4 – Блок-схема алгоритма**

# 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

## **Файл** **Arr.cpp**

Листинг 1 – Arr.cpp

|  |
| --- |
| #include "Arr.h"  Arr::Arr(int i)  {  if(i > 4){  arr = new int[i];  for(int j = 0; j < i; j++)  {  arr[j] = i;  }  }  }  void Arr::coutArr()  {  for(int j = 0; j < arr[0]; j++)  {  std::cout << arr[j];  if(j != arr[0] - 1)  {  std::cout << " ";  }  }  }  Arr::~Arr(){  delete[] arr;  } |

## **Файл** **Arr.h**

Листинг 2 – Arr.h

|  |
| --- |
| #ifndef ARR\_H  #define ARR\_H  #include <iostream>  class Arr  {  public:  int\*arr;  Arr(int i);  ~Arr();  void coutArr();  };  #endif |

## **Файл** **main.cpp**

Листинг 3 – main.cpp

|  |
| --- |
| #include <stdlib.h>  #include <stdio.h>  #include "Arr.h"  int main()  {  int i, j;  std::cin >> i >> j;  if(i <= 4){  std::cout << i << "?";  return(0);  }  if(j <= 4){  std::cout << j << "?";  return(0);  }  Arr ar1 = Arr(i);  Arr ar2 = Arr(j);  int\*tmp = ar1.arr;  ar1 = ar2;  ar2.arr = tmp;  ar2.coutArr();  std::cout << "\n";  ar1.coutArr();  return(0);  } |

# 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

| Входные данные | Ожидаемые выходные данные | Фактические выходные данные |
| --- | --- | --- |
| 5  6 | 5 5 5 5 5  6 6 6 6 6 6 | 5 5 5 5 5  6 6 6 6 6 6 |
| 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 |
| 5  8 | 5 5 5 5 5  8 8 8 8 8 8 8 8 | 5 5 5 5 5  8 8 8 8 8 8 8 8 |
| 5  9 | 5 5 5 5 5  9 9 9 9 9 9 9 9 9 | 5 5 5 5 5  9 9 9 9 9 9 9 9 9 |
| 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 |
| 1  2 | 1? | 1? |
| 1  3 | 1? | 1? |
| 2  3 | 2? | 2? |
| 3  4 | 3? | 3? |
| 4  3 | 4? | 4? |

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на С++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.

2. Шилдт Г. С++: 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/methodichescoe\_posobie\_dlya\_laboratorny h\_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).