Здесь будет титульник, листай ниже

**СОДЕРЖАНИЕ**

[1 ПОСТАНОВКА ЗАДАЧИ 5](#__RefHeading___Toc2579_1441051770)

[1.1 Описание входных данных 6](#__RefHeading___Toc2581_1441051770)

[1.2 Описание выходных данных 6](#__RefHeading___Toc2583_1441051770)

[2 МЕТОД РЕШЕНИЯ 7](#__RefHeading___Toc2585_1441051770)

[3 ОПИСАНИЕ АЛГОРИТМОВ 8](#__RefHeading___Toc2587_1441051770)

[3.1 Алгоритм функции main 8](#__RefHeading___Toc2589_1441051770)

[3.2 Алгоритм конструктора класса Class1 9](#__RefHeading___Toc2591_1441051770)

[3.3 Алгоритм деструктора класса Class1 9](#__RefHeading___Toc2593_1441051770)

[3.4 Алгоритм метода ArrayOutput класса Class1 10](#__RefHeading___Toc2595_1441051770)

[4 БЛОК-СХЕМЫ АЛГОРИТМОВ 11](#__RefHeading___Toc2597_1441051770)

[5 КОД ПРОГРАММЫ 15](#__RefHeading___Toc2599_1441051770)

[5.1 Файл Class1.cpp 15](#__RefHeading___Toc2601_1441051770)

[5.2 Файл Class1.h 15](#__RefHeading___Toc2603_1441051770)

[5.3 Файл main.cpp 16](#__RefHeading___Toc2605_1441051770)

[6 ТЕСТИРОВАНИЕ 17](#__RefHeading___Toc2607_1441051770)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#__RefHeading___Toc2609_1441051770)

# 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:

* . . .
* Первому объекту присвоить второй объект.
* . . .
* С первой строки вывести содержимое массива первого объекта.
* . . .
* 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 библиотека iostream, пространство имен std, оператор цикла со счётчиком for, условная конструкция if, объекты стандартных потоков ввода/вывода cin/cout.

Класс Class1:

* Свойства/поля: Поле, отвечающее за объявление указателя на динамический массив:
  + Наименование - arr;
  + Тип - int\*;
  + Модификатор доступа - public;
* Функционал:
  + Метод Class1 - параметризированный конструктор для создания объекта класса Class1.
  + Метод ArrayOutput - используется для вывода всех элементов массива на экран.
  + Метод ~Class1 - деструктор для удаления объекта класса Class1.

# 3 ОПИСАНИЕ АЛГОРИТМОВ

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

## Алгоритм функции main

Функционал: Основной алгоритм работы программы.

Параметры: Отсутствуют.

Возвращаемое значение: int - индикатор корректности завершения работы 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 с клавиатуры | 3 |
| 3 |  | Создание объекта obj1 класса Class1 путём вызова конструктора с параметром data | 4 |
| 4 |  | Ввод значения переменной data с клавиатуры | 5 |
| 5 |  | Создание объекта obj2 класса Class1 путём вызова конструктора с параметром data | 6 |
| 6 |  | Инициализация динамического массива temp значением поля arr объекта obj1 | 7 |
| 7 |  | Присваивание объекту obj1 значение объекта obj2 | 8 |
| 8 |  | Присваивание полю arr объекта obj1 значение переменной temp | 9 |
| 9 |  | Вызов метода ArrayOutput объекта obj1 | 10 |
| 10 |  | Переход на новую строку | 11 |
| 11 |  | Вызов метода ArrayOutput объекта obj2 | ∅ |

## Алгоритм конструктора класса Class1

Функционал: Создание объекта на основе класса Class1.

Параметры: int size - используется для задания размера поля arr класса Class1.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса Class1

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 | Значение переменной size меньше или равно 4 | Присвоение полю arr значение нулевого указателя | 2 |
|  | Выделение памяти под динамический массив arr размера size через оператор new | 4 |
| 2 |  | Вывод значения переменной size и "?" на экран | 3 |
| 3 |  | Вызов деструктора класса Class1 | ∅ |
| 4 |  | Инициализация переменной i значением 0 | 5 |
| 5 | Значение переменной i меньше size | Присваивание i-той ячейке массива arr значение переменной size | 6 |
|  |  | ∅ |
| 6 |  | Инкремент значения переменной i | 5 |

## Алгоритм деструктора класса Class1

Функционал: Удаление объекта на основе класса Class1.

Параметры: отсутствуют.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса Class1

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 | Значение поля arr класса Class1 не равно значению нулевого указателя | Удаление динамического массива arr и  овобождение выделенной под него памяти через  оператор delete | ∅ |
|  |  | ∅ |

## Алгоритм метода ArrayOutput класса Class1

Функционал: Вывод всех элементов массива arr класса Class1 на экран.

Параметры: отсутствуют.

Возвращаемое значение: отсутствует.

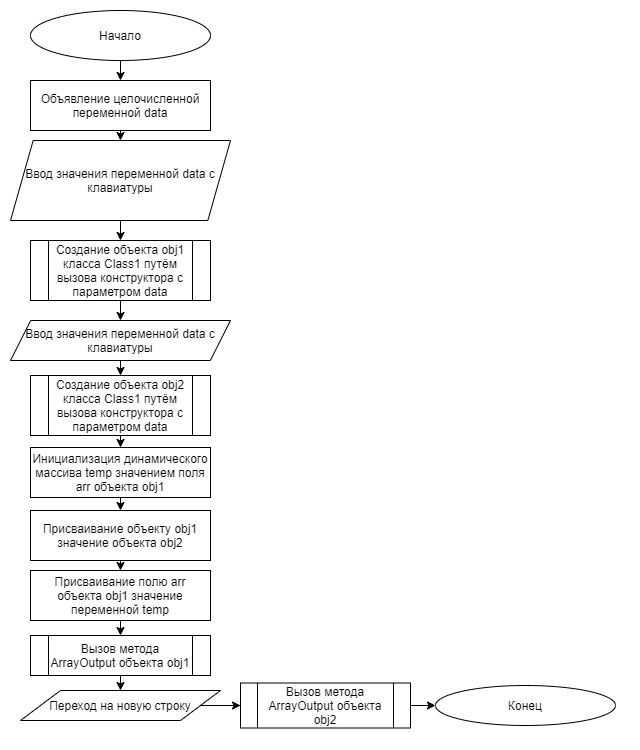
Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода ArrayOutput класса Class1

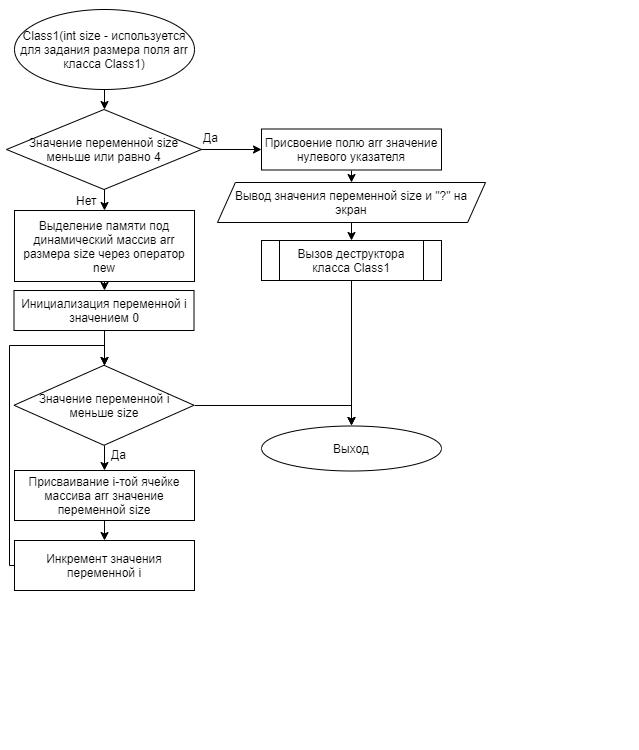
| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вывод первого элемента массива arr на экран | 2 |
| 2 |  | Инициализация переменной i значением 1 | 3 |
| 3 | Значение переменной i меньше первого элемента массива arr | Вывод двух пробелов и i-того элемента массива arr на экран | 4 |
|  |  | ∅ |
| 4 |  | Инкремент значения переменной i | 3 |

# 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

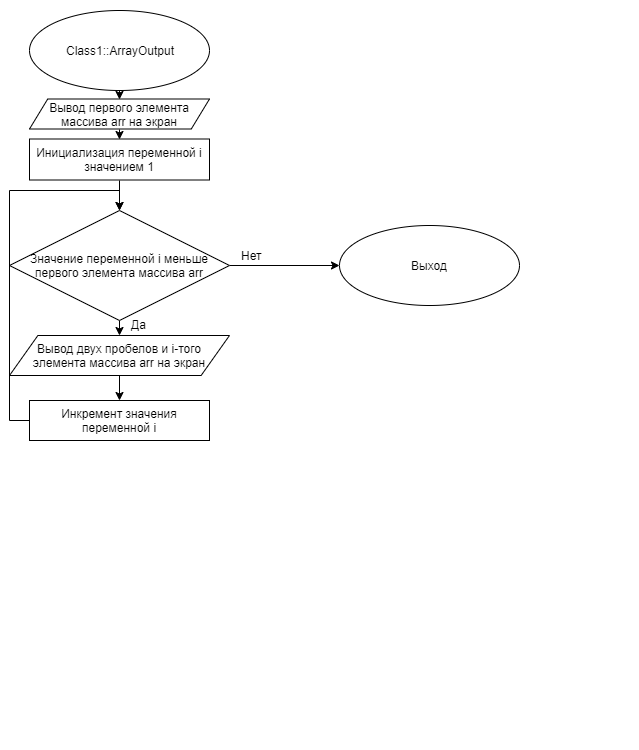
Представим описание алгоритмов в графическом виде на рисунках 1-4.



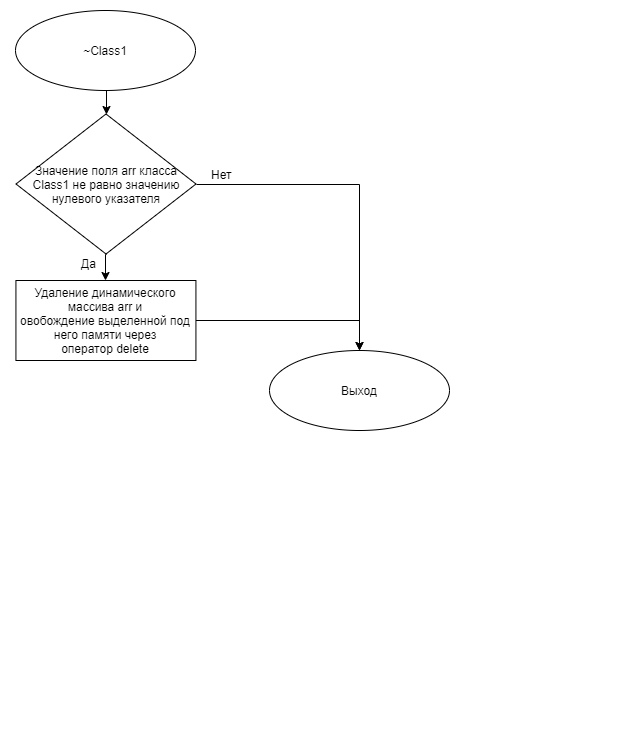
**Рисунок 1 – Блок-схема алгоритма**



**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** **Class1.cpp**

Листинг 1 – Class1.cpp

|  |
| --- |
| #include <iostream>  #include "Class1.h"  using namespace std;  Class1::Class1(int size)  {  if(size <= 4)  {  arr = nullptr;  cout << size << "?";  exit(0);  }  arr = new int[size];  for (int i=0; i < size; i++)  arr[i] = size;  }  void Class1::ArrayOutput()  {  cout << arr[0];  for (int i=1; i < arr[0]; i++)  cout << " " << arr[i];  }  Class1::~Class1()  {  if(arr != nullptr)  {  delete[] arr;  }  } |

## **Файл** **Class1.h**

Листинг 2 – Class1.h

|  |
| --- |
| #ifndef CLASS1\_H  #define CLASS1\_H  class Class1  {  public:  int\* arr;  Class1(int size);  void ArrayOutput();  ~Class1();  };  #endif |

## **Файл** **main.cpp**

Листинг 3 – main.cpp

|  |
| --- |
| #include <iostream>  #include "Class1.h"  using namespace std;  int main()  {  int data;  cin >> data;  Class1 obj1(data);  cin >> data;  Class1 obj2(data);  int\* temp = obj1.arr;  obj1 = obj2;  obj1.arr = temp;  obj1.ArrayOutput();  cout << endl;  obj2.ArrayOutput();  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).