

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 конструктора класса Class.....	8
3.2 Алгоритм метода Output класса Class.....	8
3.3 Алгоритм деструктора класса Class.....	9
3.4 Алгоритм функции main.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	11
5 КОД ПРОГРАММЫ.....	14
5.1 Файл Class.cpp.....	14
5.2 Файл Class.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:

- объект а класса Class предназначен для Создания первого массива;
- объект b класса Class предназначен для Создания второго массива;
- cin - объект стандартного потока ввода;
- cout - объект стандартного потока вывода;
- if - условный оператор;
- for - оператор начала цикла.

Класс Class:

- свойства/поля:
 - поле Массив:
 - наименование — mas;
 - тип — Указатель на массив целого типа;
 - модификатор доступа — public;
- функционал:
 - метод Class — Конструктор (Создание массива);
 - метод Output — Вывод значений элементов массива;
 - метод ~Class — Деструктор.

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 конструктора класса Class

Функционал: Конструктор (Создание массива).

Параметры: целочисленный, n , количество элементов в массиве и их значение.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Class

№	Предикат	Действия	№ перехода
1		Присвоение указателю на массив целочисленного типа <code>mas</code> размерности n	2
2		Заполнение массива элементами со значением n	Ø

3.2 Алгоритм метода Output класса Class

Функционал: Вывод значений элементов массива.

Параметры: нет.

Возвращаемое значение: `void` - не возвращает значения.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода Output класса Class

№	Предикат	Действия	№ перехода
1		Вывод с первой строки значений элементов массива через пробел	Ø

3.3 Алгоритм деструктора класса Class

Функционал: Деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса Class

№	Предикат	Действия	№ перехода
1		Освобождение выделенной для массива памяти	Ø

3.4 Алгоритм функции main

Функционал: Основной алгоритм программы.

Параметры: нет.

Возвращаемое значение: Целочисленный - индикатор корректности завершения программы.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление целочисленных n, n1	2
2		объявление указателя на целочисленный массив mas1	3
3		ввод значения n с клавиатуры	4
4	n<5	Вывод с первой строки n"?"	Ø
			5
5		Создание объекта а класса Class с параметром n	6
6		Присвоение указателю на целочисленный массив mas1 сформированный массив mas конструктором для объекта а	7

№	Предикат	Действия	№ перехода
7		ввод значения n1 с клавиатуры	8
8	n1<5	Вывод с первой строки n1"?"	∅
			9
9		Создание объекта b класса Class с параметром n1	10
10		Присвоению объекту a объекта b	11
11		Присвоению указателю mas на целочисленный массив объекта a, mas1	12
12		Вывод с первой строки на экран значений массива первого объекта через два пробела с помощью метода Output класса Class	13
13		Вывод со второй строки на экран значений массива второго объекта через два пробела с помощью метода Output класса Class	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

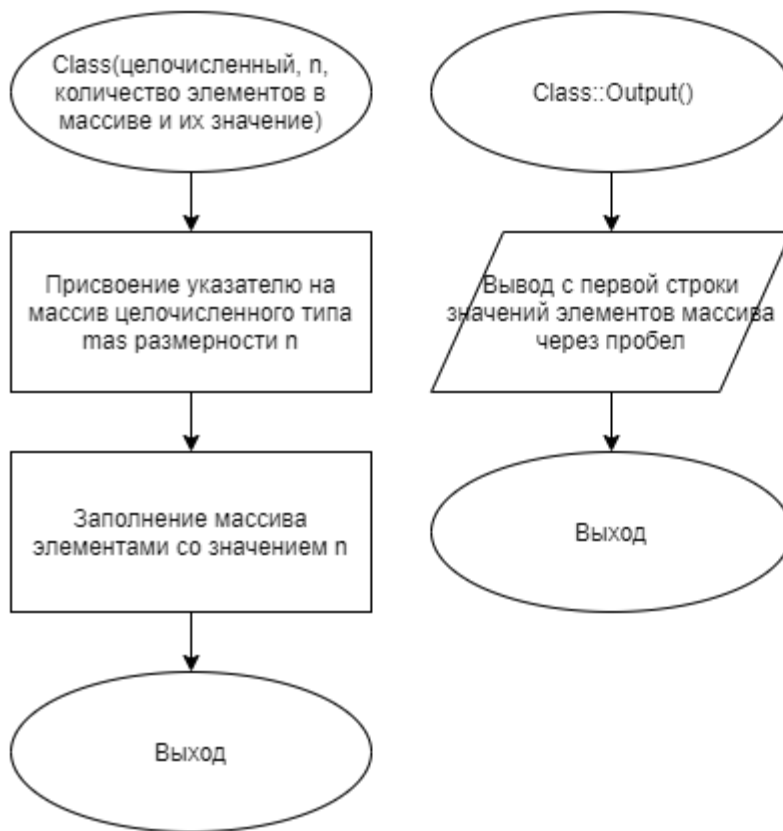


Рисунок 1 – Блок-схема алгоритма

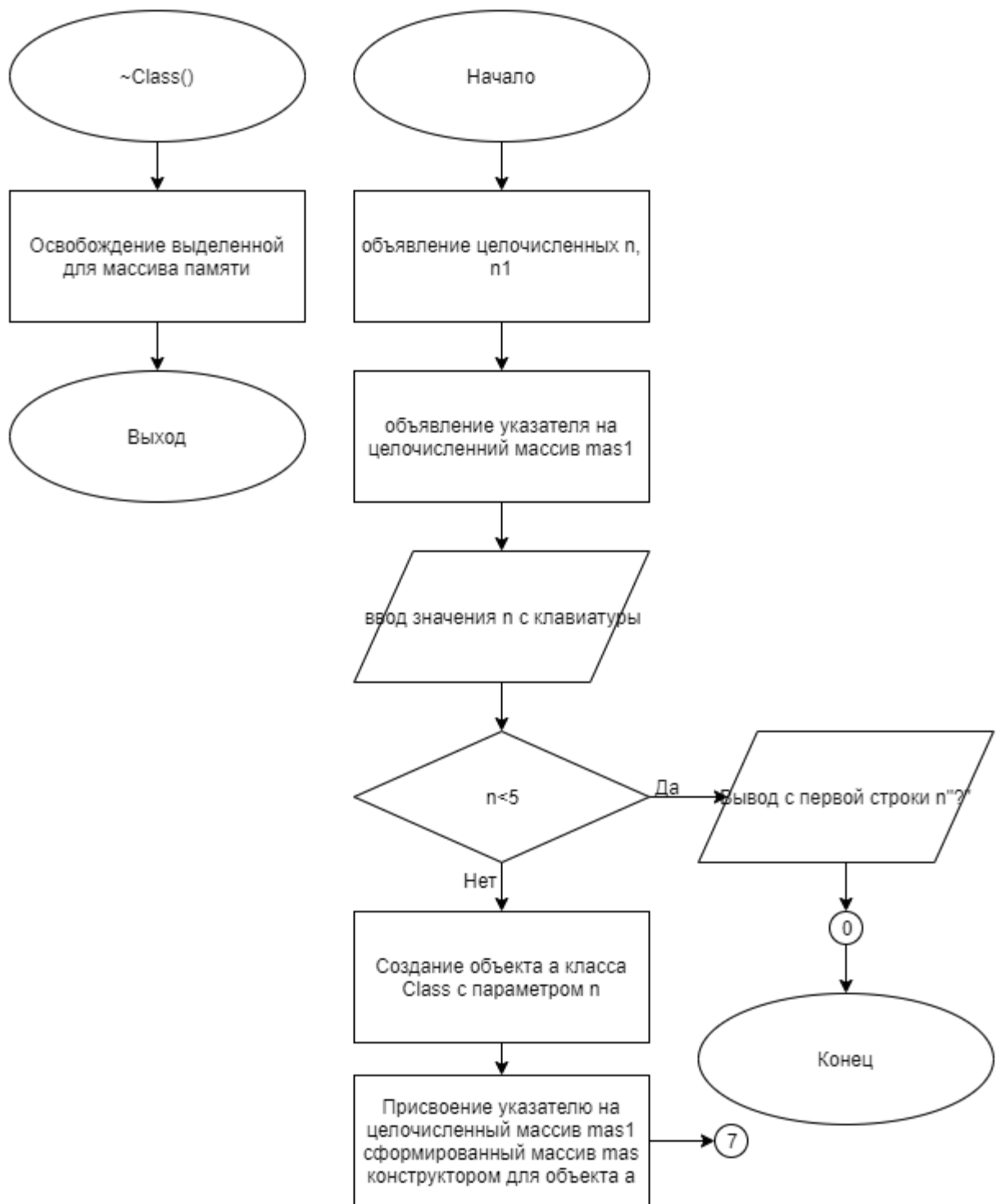


Рисунок 2 – Блок-схема алгоритма

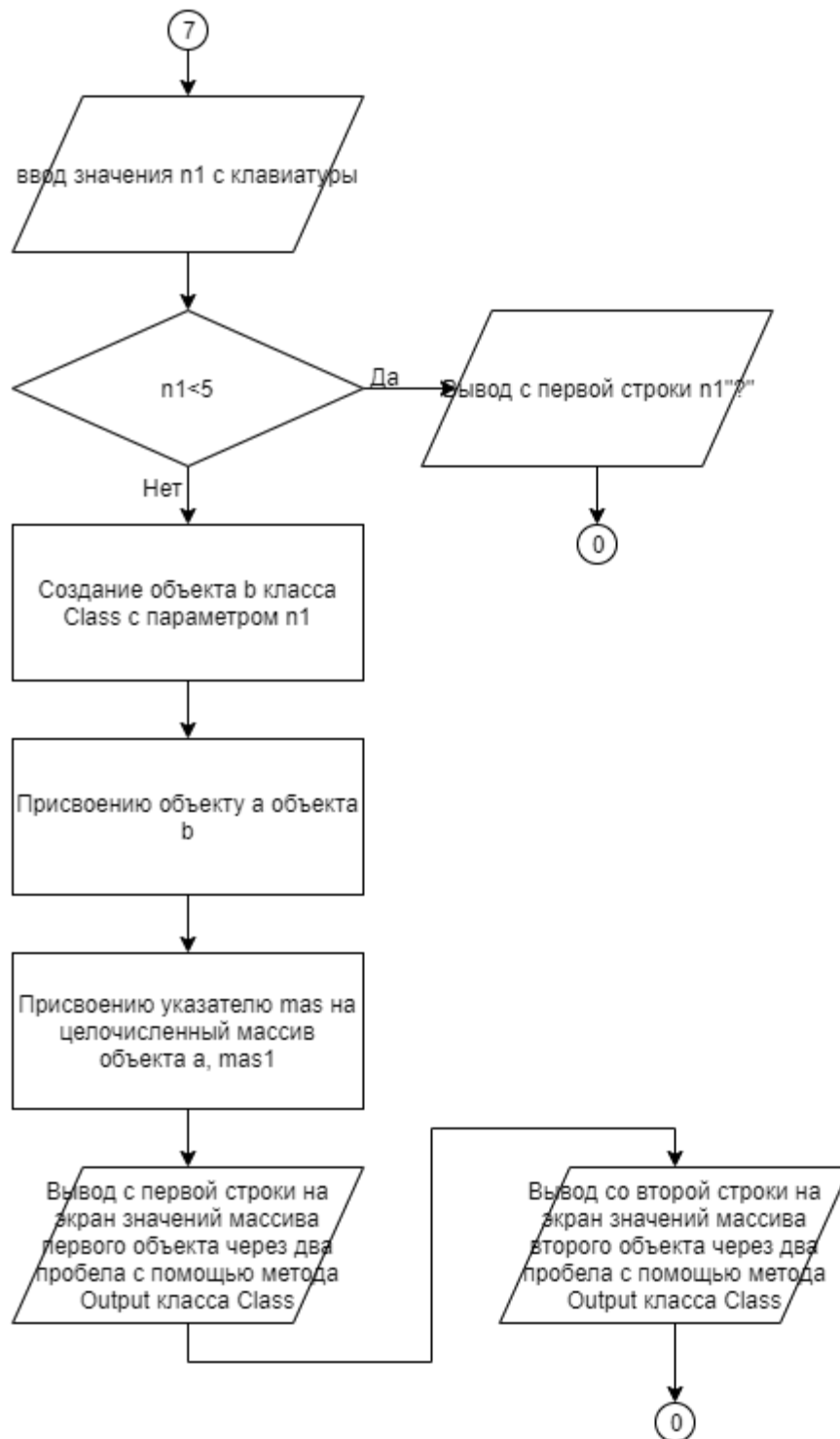


Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Class.cpp

Листинг 1 – Class.cpp

```
#include "Class.h"
#include <iostream>
using namespace std;
Class::Class(int n){
    mas=new int[n];
    for (int i=0; i<n; i++){
        mas[i]=n;
    }
}
void Class::Output(){
    for (int i=0; i<mas[0]-1; i++){
        cout<<mas[i]<<" ";
    }
    cout<<mas[0]<<endl;
}
Class::~~Class(){
    delete[] mas;
}
```

5.2 Файл Class.h

Листинг 2 – Class.h

```
#ifndef __CLASS__H
#define __CLASS__H
class Class{
public:
    Class(int n);
    int *mas;
    ~Class();
    void Output();
};
#endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
using namespace std;
#include "Class.h"
int main()
{
    int n,n1;
    int *mas1;
    cin >> n;
    if (n<5){cout<<n<<"?"; return(0);}
    Class a(n);
    mas1=a.mas;
    cin >> n1;
    if (n1<5){cout<<n1<<"?"; return(0);}
    Class b(n1);
    a=b;
    a.mas=mas1;
    a.Output();
    b.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 10	2?	2?
5 6	5 5 5 5 5 6 6 6 6 6 6	5 5 5 5 5 6 6 6 6 6 6

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).