

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	7
1.2 Описание выходных данных.....	7
2 МЕТОД РЕШЕНИЯ.....	9
3 ОПИСАНИЕ АЛГОРИТМОВ.....	10
3.1 Алгоритм конструктора класса N.....	10
3.2 Алгоритм конструктора класса N.....	10
3.3 Алгоритм деструктора класса N.....	11
3.4 Алгоритм метода show класса N.....	11
3.5 Алгоритм функции func1.....	12
3.6 Алгоритм функции func2.....	12
3.7 Алгоритм функции main.....	13
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	14
5 КОД ПРОГРАММЫ.....	18
5.1 Файл main.cpp.....	18
5.2 Файл N.cpp.....	19
5.3 Файл N.h.....	19
6 ТЕСТИРОВАНИЕ.....	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	22

1 ПОСТАНОВКА ЗАДАЧИ

Сконструировать систему, которая демонстрирует особенности при возврате объекта из функции. Показывает, какого характера ошибки могут возникнуть при выполнении возврата и как их обойти.

Спроектировать объект (разработать описание класса), который имеет свойства:

- указатель на массив целого типа в открытом доступе;
- переменную целого типа в закрытом доступе, содержит количество элементов массива.

Функционал объекта:

- конструктор по умолчанию. Выдает сообщение, что работает конструктор по умолчанию;
- параметризованный конструктор, с одним целочисленным параметром. С новой строки выводит сообщение, что работает параметризованный конструктор. Конструктор создает массив целочисленного типа размерности согласно значению параметра. Каждому элементу присваивает значение равное квадрату соответствующего индекса;
- метод, в открытом доступе, который выводит значения элементов массива с новой строки, разделенным тремя пробелами;
- деструктор, выдает с новой строки сообщение, что работает деструктор. После этого освобождает память, выделенную для массива.

Разработать функцию номером 1, с параметром целого типа. Посредством данного параметра передается значение размерности массива. В описании функции реализовать алгоритм:

1. Объявить объект, использовать параметризованный конструктор. В качестве аргумента конструктору передать параметр функции.

2. Для объекта вызвать метод вывода содержимого массива.
3. Возвратить объявленный объект.

Разработать функцию номером 2, с параметром целого типа. Посредством данного параметра передается значение размерности массива. В описании функции реализовать алгоритм:

1. Создать объект, использовать параметризованный конструктор. В качестве аргумента конструктору передать параметр функции. Инициировать указатель на объект заданного класса адресом созданного объекта.
2. Для объекта вызвать метод вывода содержимого массива.
3. Вернуть значение указателя на объект заданного класса.

Первоначальный алгоритм сборки и отработки системы:

1. Объявляется целочисленная переменная.
2. Объявляется объект заданного класса.
3. Объявляется указатель на объект заданного класса.
4. Вводится целое, положительное число, его значение размещается в целочисленную переменную.
5. Объявленному объекту присваивается значение, возвращаемое функцией с номером 1. В функцию с номером 1 в качестве аргумента передается целочисленная переменная.
6. Для объявленного объекта вызывается метод вывода содержимого массива.
7. Вводится целочисленное, положительное число, его значение размещается в целочисленную переменную.
8. Объявленному указателю на объект заданного класса присваивается значение, возвращаемое функцией с номером 2. В функцию с номером 2 в качестве аргумента передается целочисленная переменная.

9. Для объекта, адрес которой является значением указателя, вызваться метод вывода содержимого массива.

10. Уничтожается объект, адрес которого является значением указателя.

11. Завершается работа системы со значением кода возврата равного 0.

При такой реализации, работа программы завершиться аварийно или значения третьей и пятой строки вывода не совпадут.

Закомментируйте один оператор так, чтобы программа завершилась нормально и выводимые значения в третьей и пятой строке совпали.

При сдаче задания, надо пояснить, почему первоначально произошла ошибка и почему выполненная коррекция кода исправляет ситуацию.

1.1 Описание входных данных

Первая строка:

«целое число»

вторая строка:

«целое число»

Пример ввода

5
8

1.2 Описание выходных данных

Конструктор по умолчанию выводит сообщение:

Default constructor

Параметризованный конструктор выводит сообщение:

Parameterized constructor

Деструктор выводит сообщение:

Destructor

Метод вывода значений элементов массива выводит последовательность чисел:

«целое число» «целое число» «целое число» . . .

Возможный пример вывода до доработки программы-системы:

```
Default constructor
Parameterized constructor
0  1  4  9  16
Destructor
2060656  2031808  4  9  16
Parameterized constructor
0  1  4  9  16  25  36  49
0  1  4  9  16  25  36  49
Destructor
Destructor
```

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект obj класса N предназначен для ;
- библиотека iostream;
- объекты стандартного потока ввода/вывода данных - cin/cout;
- пространство имен std.

Класс N:

- свойства/поля:
 - поле целочисленная переменная:
 - наименование — size;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод N — конструктор по умолчанию;
 - метод N — параметризованный конструктор;
 - метод ~N — деструктор по умолчанию;
 - метод show — вывод значения элементов массива с новой строки, разделенной тремя пробелами.

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 N

Функционал: конструктор по умолчанию.

Параметры: нет.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса N

№	Предикат	Действия	№ перехода
1		присвоение переменной size значения 0	2
2		вывод "Default constructor"	∅

3.2 Алгоритм конструктора класса N

Функционал: параметризованный конструктор.

Параметры: int P.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса N

№	Предикат	Действия	№ перехода
1		присвоение переменной size значения p	2
2		присвоение arr значением new int[p]	3
3		вывод "Parameterized constructor"	4
4		инициализация i значения 0 типа int	5
5	i < p?	присвоение arr[i] значения i*i	6

№	Предикат	Действия	№ перехода
			∅
6		инкремент i	5

3.3 Алгоритм деструктора класса N

Функционал: деструктор по умолчанию.

Параметры: нет.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса N

№	Предикат	Действия	№ перехода
1		вывод "Destructor"	2
2		delete[] arr	∅

3.4 Алгоритм метода show класса N

Функционал: вывод значения элементов массива с новой строки, разделенной тремя пробелами.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода show класса N

№	Предикат	Действия	№ перехода
1		инициализация i значения 0 типа int	2
2	i < size?		3
			5
3	i == size-1?	вывод i*i	4
		вывод i*i " "	4

№	Предикат	Действия	№ перехода
4		инкремент i	2
5		переход на новую строку	∅

3.5 Алгоритм функции func1

Функционал: вызов метода вывода массива.

Параметры: int N.

Возвращаемое значение: obj.

Алгоритм функции представлен в таблице 5.

Таблица 5 – Алгоритм функции func1

№	Предикат	Действия	№ перехода
1		объявление объекта, используя параметризованный конструктор	2
2		вызов метода вывода содержимого массива	3
3		объявление объекта	∅

3.6 Алгоритм функции func2

Функционал: вызов метода вывода массива.

Параметры: int N.

Возвращаемое значение: нет.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции func2

№	Предикат	Действия	№ перехода
1		создание объекта, используя параметризованный конструктор, инициализация указателя на объект заданного класса адресом заданного объекта	2
2		вызов метода вывода содержимого массива	3

№	Предикат	Действия	№ перехода
3		возвращение значения указателя на объект заданного класса	Ø

3.7 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое число - идентификатор работоспособности программы.

Алгоритм функции представлен в таблице 7.

Таблица 7 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление переменных n1, n2 типа int	2
2		объявление объекта obj заданного класса	3
3		объявление указателя ptr на объекта заданного класса	4
4		ввод значения переменной n1 с клавиатуры	5
5		присвоение obj значения func1(n1)	6
6		вызов метода show() для объекта obj	7
7		ввод значения переменной n2 с клавиатуры	8
8		присвоение ptr значения func2(n2)	9
9		вызов метода show() для указателя ptr	10
10		вывод "Destructor"	Ø

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

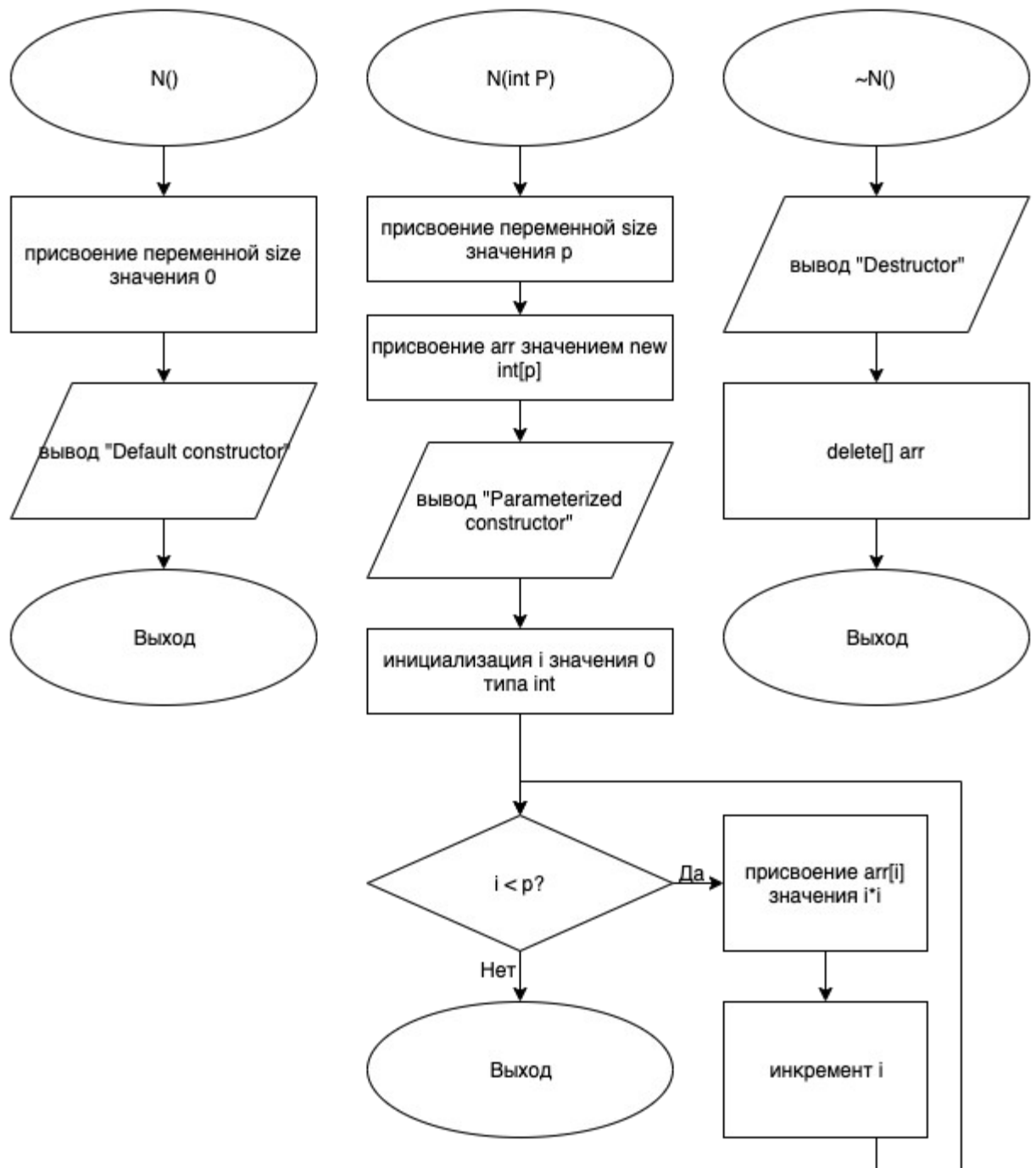


Рисунок 1 – Блок-схема алгоритма

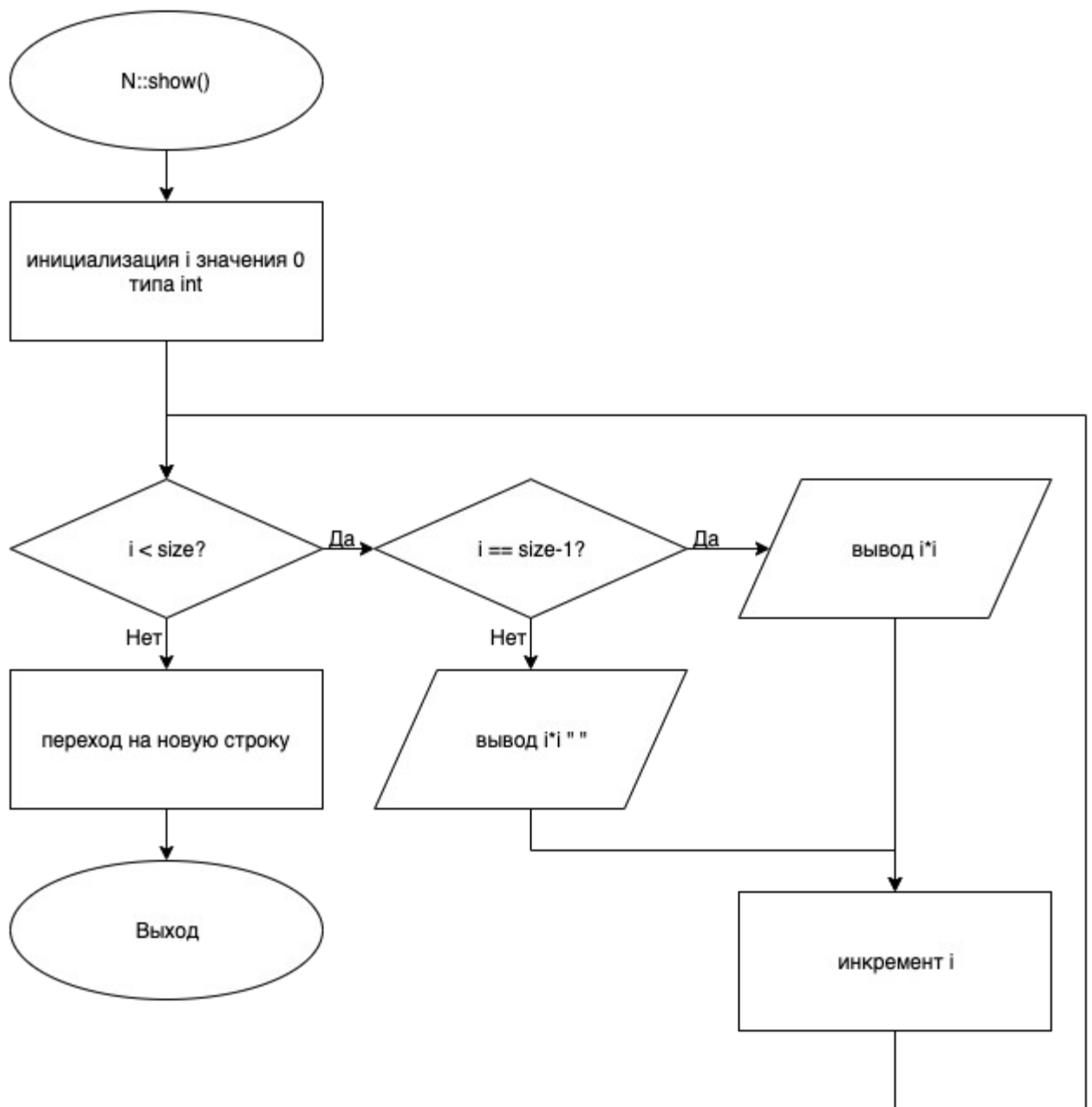


Рисунок 2 – Блок-схема алгоритма

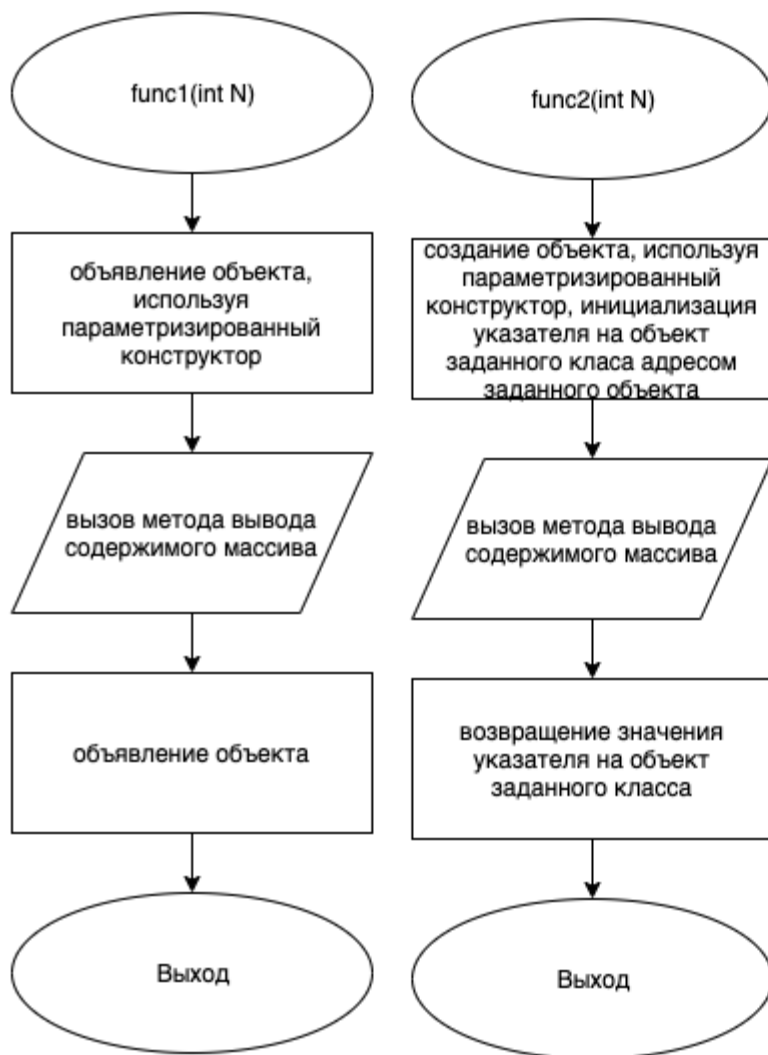


Рисунок 3 – Блок-схема алгоритма

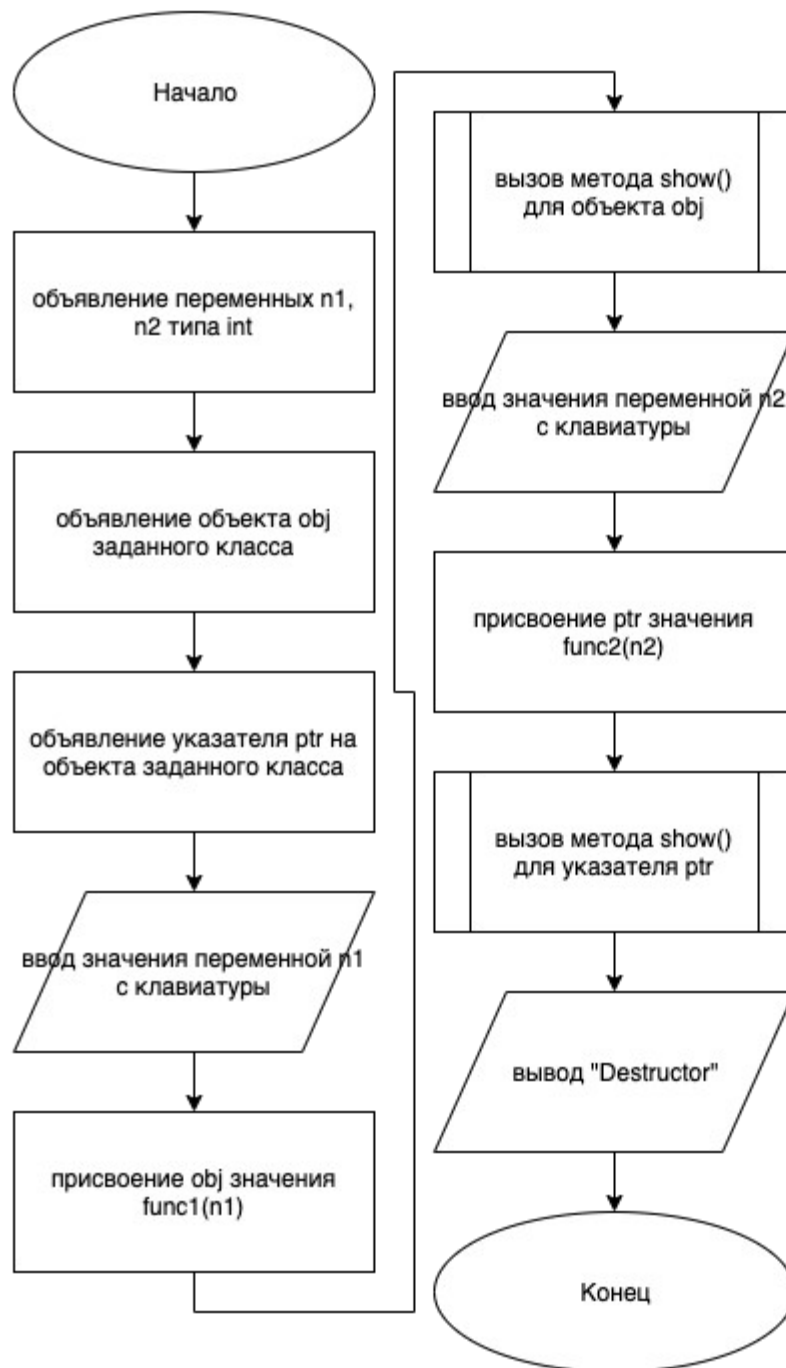


Рисунок 4 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "N.h"
#include <iostream>
using namespace std;

N func1(int A){
    N obj(A);
    obj.show();
    return obj;
}

N*func2(int A){
    N*obj = new N(A);
    obj -> show();
    return obj;
}

int main()
{
    int n1, n2;
    N obj;
    N* ptr;
    cin >> n1;
    obj = func1(n1);
    obj.show();

    cin >> n2;
    ptr = func2(n2);
    ptr -> show();
    cout << "Destructor" << endl;

    return(0);
}
```


5.2 Файл N.cpp

Листинг 2 – N.cpp

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

N::N(){
    size = 0;
    cout << "Default constructor" << endl;
}

N::N(int p){
    size = p;
    arr = new int[p];
    cout << "Parameterized constructor" << endl;
    for (int i = 0; i < p; ++i) {
        arr[i] = i*i;
    }
}

N::~~N(){
    cout << "Destructor" << endl;
}

void N::show(){
    for (int i = 0; i < size; ++i) {
        if (i == size-1) {
            cout << i*i;
        } else {
            cout << i*i << "  ";
        }
    }
    cout << endl;
}
```

5.3 Файл N.h

Листинг 3 – N.h

```
#ifndef __N__H
#define __N__H
#include <iostream>
using namespace std;

class N{
private:
```

```
        int size;  
    public:  
        int* arr;  
        N();  
        ~N();  
        void show();  
        N(int p);  
};  
  
#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 8.

Таблица 8 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
5 8	Default constructor Parameterized constructor 0 1 4 9 16 Destructor 0 1 4 9 16 Parameterized constructor 0 1 4 9 16 25 36 49 0 1 4 9 16 25 36 49 Destructor Destructor	Default constructor Parameterized constructor 0 1 4 9 16 Destructor 0 1 4 9 16 Parameterized constructor 0 1 4 9 16 25 36 49 0 1 4 9 16 25 36 49 Destructor Destructor

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).