

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	5
1.2 Описание выходных данных.....	5
2 МЕТОД РЕШЕНИЯ.....	6
3 ОПИСАНИЕ АЛГОРИТМОВ.....	7
3.0 Алгоритм функции main.....	7
3.1 Алгоритм конструктора класса cl.....	7
3.2 Алгоритм метода f класса cl.....	8
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	9
5 КОД ПРОГРАММЫ.....	11
5.0 Файл cl.cpp.....	11
5.1 Файл cl.h.....	11
5.2 Файл main.cpp.....	12
6 ТЕСТИРОВАНИЕ.....	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	14

1 ПОСТАНОВКА ЗАДАЧИ

Разработать объект, который обладает следующей функциональностью:

- В конструкторе вводит количество элементов целочисленного массива, создает целочисленный массив заданным количеством элементов, определяет значения элементов массива. Значение каждого элемента равно квадрату индекса элемента.

- Выводит значения элементов массива в обратном порядке.

Необходимые свойства объекта определяется разработчиком. Количество элементов массива принадлежит интервалу от 5 до 20.

Написать программу:

1. Создает объект посредством оператора new.
2. Выводит значения элементов массива в обратном порядке.
3. Удаляет объект посредством оператора delete.

1.1 Описание входных данных

Первая строка (количество элементов массива):

«целое число в десятичном формате»

1.2 Описание выходных данных

2 МЕТОД РЕШЕНИЯ

Для решения задачи понадобится:

оператор for

используются операторы функции new, delete

массив array

библиотека iomanip (функция setw)

Класс cl

Поля:

скрытые элементы:

int n = 0

int* array указатель массива

Методы:

Открытые:

cl() - конструктор класса

f() - вывод элементов массива в обратном порядке

3 ОПИСАНИЕ АЛГОРИТМОВ

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

3.0 Алгоритм функции main

Функционал: главный метод программы.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		создание объекта класса посредством оператора new	2
2		вывод элементов массива в обратном порядке	3
3		удаление объекта класса посредством оператора delete	∅

3.1 Алгоритм конструктора класса cl

Функционал: ввод количество, создание массива, определение элементов массива.

Параметры: нет.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса cl

№	Предикат	Действия	№ перехода
1		ввод количество элементов	2

№	Предикат	Действия	№ перехода
2		создание массива	3
3		приравнивание счетчика с единицей	4
4	счетчик меньше или равно количество элементов	каждый элемент равно квадрату индекса элемента	5
			Ø
5		увеличение счетчика на 1	4

3.2 Алгоритм метода f класса cl

Функционал: вывод элементов массива в обратном порядке.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода f класса cl

№	Предикат	Действия	№ перехода
1		приравнивание счетчика с количеством элементов массива	2
2	счетчик больше нуля	вывод значения массива	3
			Ø
3		уменьшения счетчика на 1	2

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

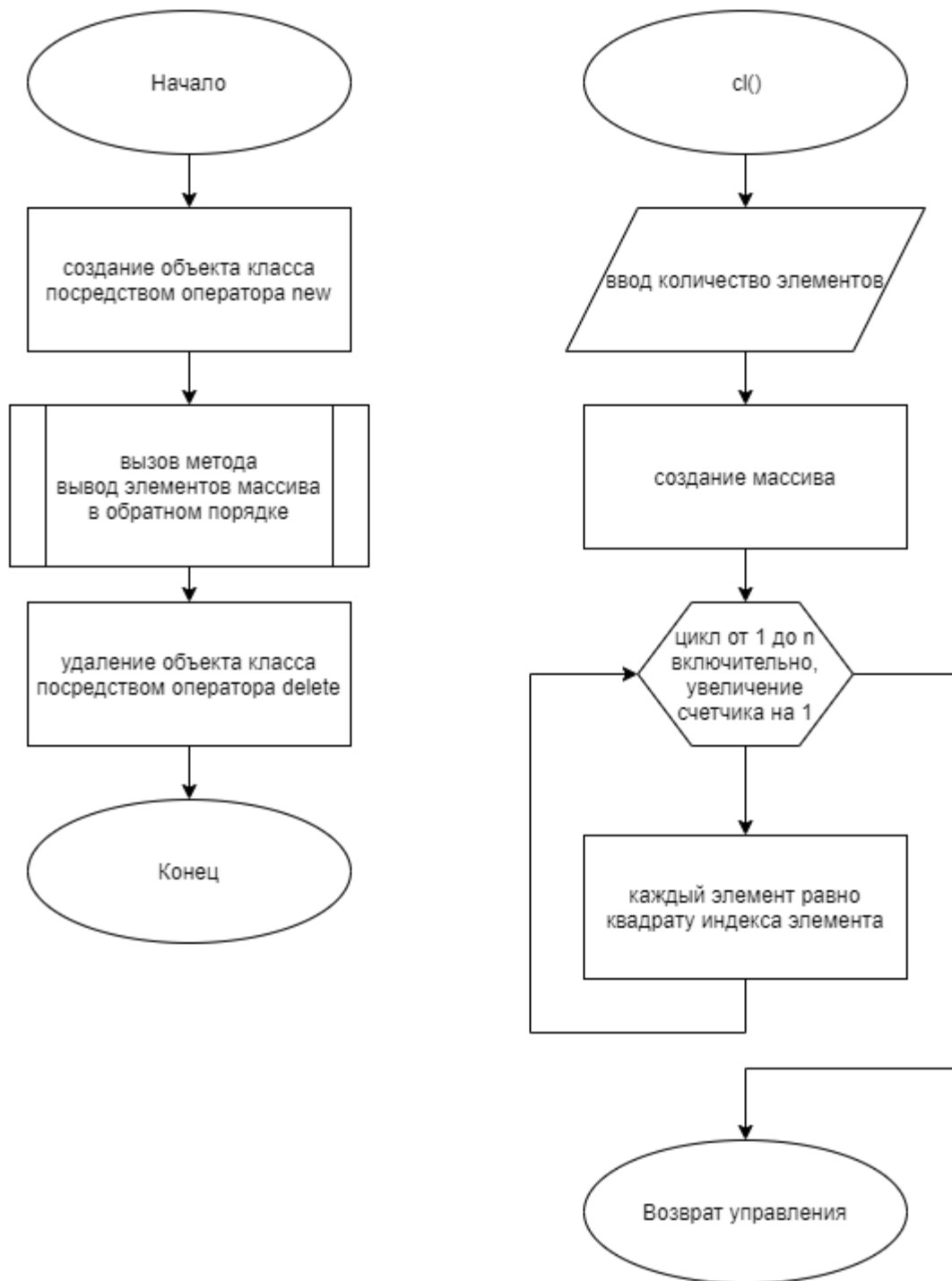


Рисунок 1 – Блок-схема алгоритма



Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.0 Файл cl.cpp

Листинг 1 – cl.cpp

```
#include <iostream>
#include <iomanip>
#include "cl.h"

using namespace std;

cl::cl()
{
    cin >> n;
    array = new int[n];
    for (int i = 1; i <= n; i++)
    {
        array[i] = (i * i);
    }
}

void cl::f()
{
    for (int i = (n - 1); i >= 0; i--)
    {
        cout << setw(5) << array[i];
    }
}
```

5.1 Файл cl.h

Листинг 2 – cl.h

```
#ifndef __CL_H
#define __CL_H

using namespace std;

class cl
{
private:
    int n = 0;
```

```
        int* array;
public:
    cl();
    void f();
};

#endif
```

5.2 Файл main.cpp

Листинг 3 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "cl.h"

using namespace std;

int main()
{
    cl* object = new cl();
    object -> f();
    delete object;
    return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 4.

Таблица 4 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные					Фактические выходные данные				
5	16	9	4	1	0	16	9	4	1	0

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. C++: 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/methodicheskoe_posobie_dlya_laboratornyh_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).