

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 деструктора класса Class.....	8
3.3 Алгоритм функции Print.....	9
3.4 Алгоритм функции main.....	9
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
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:

- ...
- 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

Объекты потока ввода/вывода cin, cout;

Оператор функция new;

Оператор функция delete;

Объекты o1 и o2 класса Class;

Класс Class:

- Свойства/поля:
 - Поле, предназначенное для объявления указателя на целочисленный массив mas:
 - Наименование - mas;
 - Тип - int*;
 - Модификатор доступа - открытый;
- Функционал:
 - Class(int n) - конструктор, принимает целочисленный параметр n и заполняет массив длиной n числами n;
 - ~Class() - деструктор, освобождает память;
 - Print(int n) - метод, выводящий текущее состояние массива.

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: принимает целочисленный параметр n и заполняет массив длиной n числами n ;

Параметры: $\text{int } n$.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Class

№	Предикат	Действия	№ перехода
1		Создание массива длиной n	2
2		Инициализация целочисленной переменной i значением 0	3
3	$i < n$	$\text{mas}[i] = n$	4
			Ø
4		$i += 1$	3

3.2 Алгоритм деструктора класса Class

Функционал: Освобождение памяти.

Параметры: .

Алгоритм деструктора представлен в таблице 2.

Таблица 2 – Алгоритм деструктора класса Class

№	Предикат	Действия	№ перехода
1		Удаление массива по указателю mas при помощи оператора функции delete	∅

3.3 Алгоритм функции Print

Функционал: Вывод текущего состояния массива.

Параметры: int n.

Возвращаемое значение: void.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции Print

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной i значением 0	2
2	$i < n$		3
			∅
3	$i \neq n - 1$	Вывод значения mas[i] с пробелом	2
		Вывод mas[i]	2

3.4 Алгоритм функции main

Функционал: 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. Присвоение первому объекту второго, вывод каждого из них..

Параметры: .

Возвращаемое значение: int - код возврата.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление целочисленных переменных n1, n2	2
2		Ввод значения переменной n1	3
3	n1 <= 4	Вывод значения переменной n1 со знаком вопроса	∅
			4
4		Ввод значения переменной n2	5
5	n2 <= 4	Вывод значения переменной n2 со знаком вопроса	∅
			6
6		Создание объекта o1 класса Class с передачей в конструктор параметра значение переменной n1	7
7		Создание объекта o2 класса Class с передачей в конструктор параметра значение переменной n2	8
8		Инициализация указателя mas адресом, который содержится в поле mas объекта o1	9
9		o1 = o2	10
10		Присваивание полю mas объекта o1 значение указателя mas	11
11		Вызов метода Print(int n1) объекта o1	12
12		Вывод переноса на новую строку	13
13		Вызов метода Print(int n2) объекта o2	14

№	Предикат	Действия	№ перехода
1 4		Возвратить 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

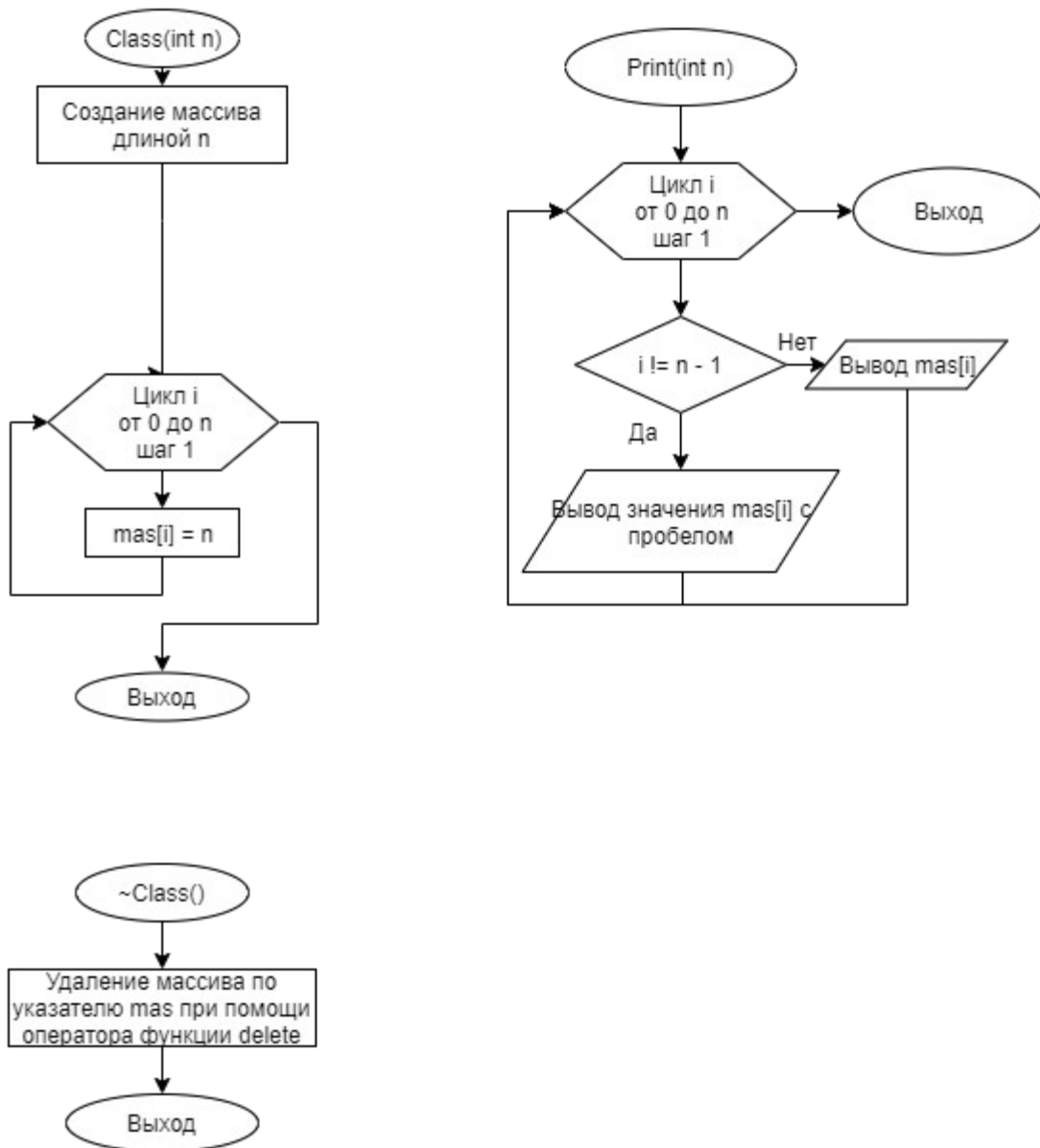


Рисунок 1 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Class.cpp

Листинг 1 – Class.cpp

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

5.2 Файл Class.h

Листинг 2 – Class.h

```
#ifndef CLASS_H
#define CLASS_H
using namespace std;
class Class {
public:
    int* mas;
    Class(int n);
    void Print(int n);
    ~Class();
};
```

```
#endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <iostream>
#include "Class.h"
using namespace std;
int main() {
    int n1, n2;
    cin >> n1;
    if (n1 <= 4) {
        cout << n1 << "?";
        return 0;
    }
    cin >> n2;
    if (n2 <= 4) {
        cout << n2 << "?";
        return 0;
    }
    Class o1(n1);
    Class o2(n2);
    int *mas = o1.mas;
    o1 = o2;
    o1.mas = mas;
    o1.Print(n1);
    cout << endl;
    o2.Print(n2);
    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 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
4 8	4?	4?
8 4	4?	4?

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).