

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 функции main.....	8
3.2 Алгоритм конструктора класса ChiefKiev.....	9
3.3 Алгоритм деструктора класса ChiefKiev.....	9
3.4 Алгоритм метода get_array класса ChiefKiev.....	10
3.5 Алгоритм метода set_array класса ChiefKiev.....	10
3.6 Алгоритм метода print_array класса ChiefKiev.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	12
5 КОД ПРОГРАММЫ.....	15
5.1 Файл ChiefKiev.cpp.....	15
5.2 Файл ChiefKiev.h.....	16
5.3 Файл main.cpp.....	16
6 ТЕСТИРОВАНИЕ.....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19

# 1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В закрытом 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 из закрытой области и каждому элементу присваивается это же значение.

Объект имеет функциональность, по которой выводит содержимое целочисленного массива. Вывод производит последовательно, разделяя значения двумя пробелами.

Функциональность объекта можно расширить по усмотрению разработчика не более чем на два метода.

Спроектировать систему, которая содержит два объекта. Для построения системы последовательно, с новых строк вводятся целочисленные значения. Если значение меньше или равно 4, то создание системы прекращается и выводится сообщение. Если система построена, то посредством параметризованного конструктора создаются объекты.

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 `cin` класса `istream` предназначен для ввода значений с клавиатуры;
- объект `cout` класса `ostream` предназначен для вывода текста;
- объект класса `ChiefKiev` предназначен для ;
- `if..else` - условный оператор;
- `new` - оператор резервирования памяти;
- `delete` - оператор освобождения памяти;
- `for` - оператор цикла со счетчиком.

Класс `ChiefKiev`:

- свойства/поля:
  - поле обрабатываемый динамический массив:
    - наименование — `arr`;
    - тип — `int*`;
    - модификатор доступа — `private`;
- функционал:
  - метод `ChiefKiev` — параметризованный конструктор;
  - метод `~ChiefKiev` — деструктор, освобождение выделенной памяти под массив `arr`;
  - метод `get_array` — геттер для закрытой переменной `arr`;
  - метод `set_array` — сеттер для закрытой переменной `arr`;
  - метод `print_array` — последовательный вывод элементов 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 функции main

Функционал: точка входа в программу.

Параметры: нет.

Возвращаемое значение: целочисленное - индикатор корректности выполнения программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной i	2
2		Ввод значения переменной i с клавиатуры	3
3	i <= 4	Вывод «i»?	4
			Ø
4		Объявление объекта lovesosa класса ChiefKiev	5
5		Ввод значения переменной i с клавиатуры	6
6	i <= 4	Вывод «i»?	7
			Ø
7		Объявление объекта maybebaby класса ChiefKiev	8
8		Инициализация целочисленного указателя temp путем вызова метода get_array()	9
9		Присваивание объекту lovesosa объекта maybebaby	10
10		Вызов метода set_array() с передачей аргумента temp	11

№	Предикат	Действия	№ перехода
11		Вызов метода print_array() у объекта lovesosa	12
12		Вызов метода print_array() у объекта maybebaby	∅

### 3.2 Алгоритм конструктора класса ChiefKiev

Функционал: параметризованный конструктор.

Параметры: целочисленный - arr\_size.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса ChiefKiev

№	Предикат	Действия	№ перехода
1		Выделение памяти под динамический массив размером arr_size	2
2		Инициализация целочисленной переменной i	3
3	i < arr_size	Присваивание i-тому элементу массива значения arr_size; инкрементация i	3
			∅

### 3.3 Алгоритм деструктора класса ChiefKiev

Функционал: деструктор, освобождение выделенной памяти под массив arr.

Параметры: нет.

Алгоритм деструктора представлен в таблице 3.

Таблица 3 – Алгоритм деструктора класса ChiefKiev

№	Предикат	Действия	№ перехода
1	arr != nullptr	Удаление динамического массива по указателю *arr	∅

### 3.4 Алгоритм метода `get_array` класса `ChiefKiev`

Функционал: геттер для закрытой переменной `arr`.

Параметры: нет.

Возвращаемое значение: целочисленный указатель.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода `get_array` класса `ChiefKiev`

№	Предикат	Действия	№ перехода
1		Возврат значения закрытого поля <code>*arr</code>	Ø

### 3.5 Алгоритм метода `set_array` класса `ChiefKiev`

Функционал: сеттер для закрытой переменной `arr`.

Параметры: целочисленный указатель - `p_array`.

Возвращаемое значение: ничего не возвращает.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода `set_array` класса `ChiefKiev`

№	Предикат	Действия	№ перехода
1		Присвоение закрытому полю <code>arr</code> значения параметра <code>p_array</code>	Ø

### 3.6 Алгоритм метода `print_array` класса `ChiefKiev`

Функционал: последовательный вывод элементов массива.

Параметры: нет.

Возвращаемое значение: ничего не возвращает.

Алгоритм метода представлен в таблице 6.



Таблица 6 – Алгоритм метода *print\_array* класса *ChiefKiev*

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной <i>size</i> равной <i>arr[0]</i>	2
2	<i>i &lt; size</i>	Вывод значения <i>arr[i]</i> ; инкрементация <i>i</i>	2
			∅

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

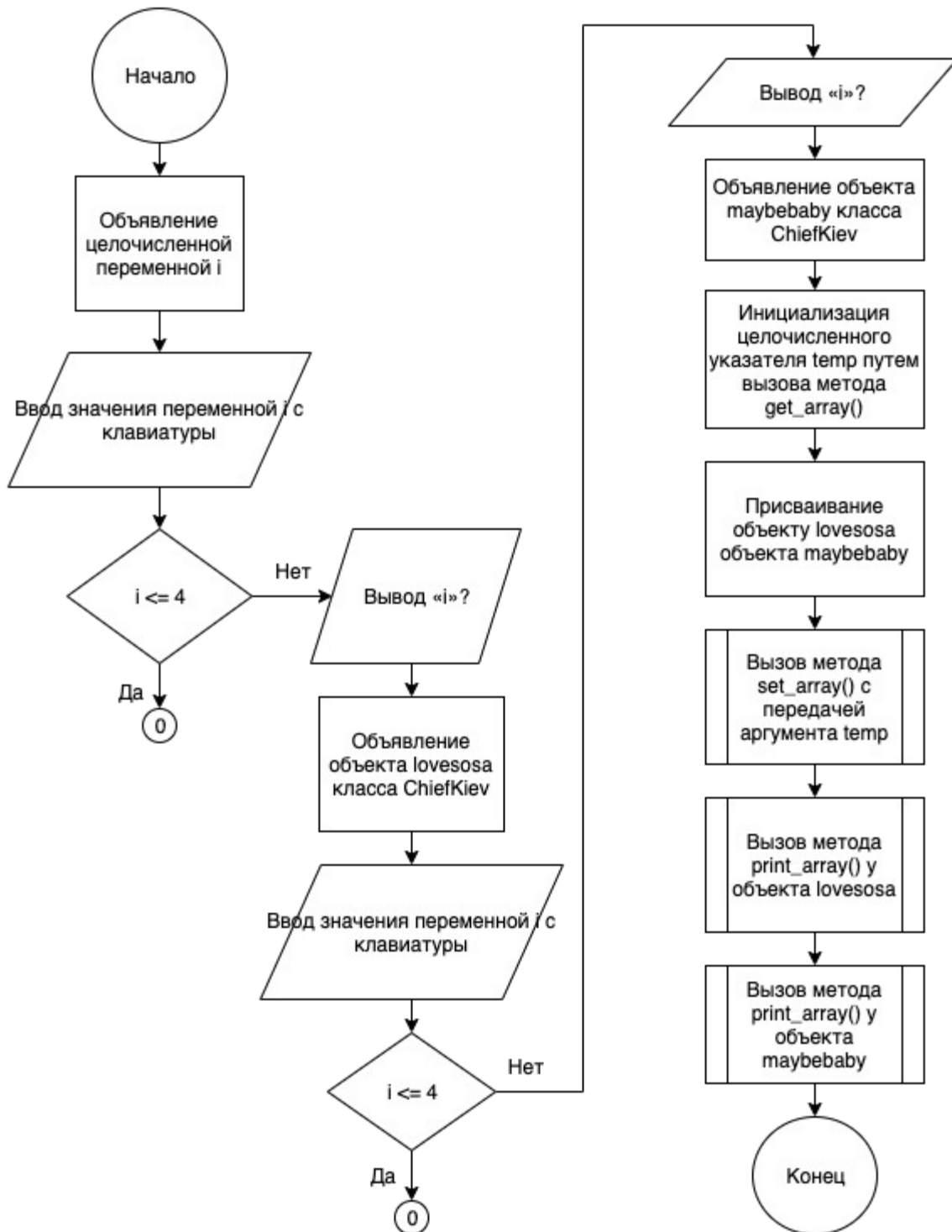


Рисунок 1 – Блок-схема алгоритма

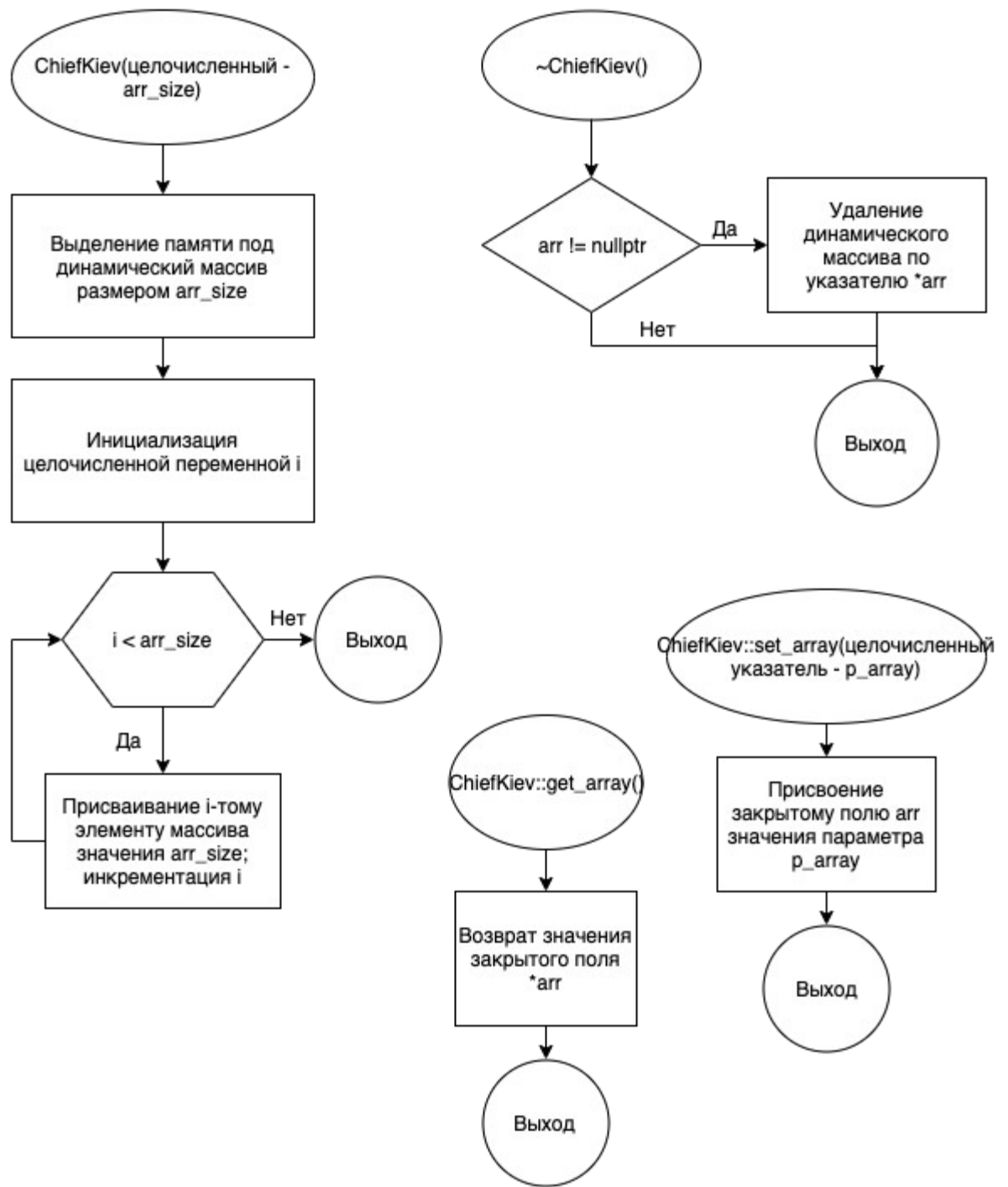
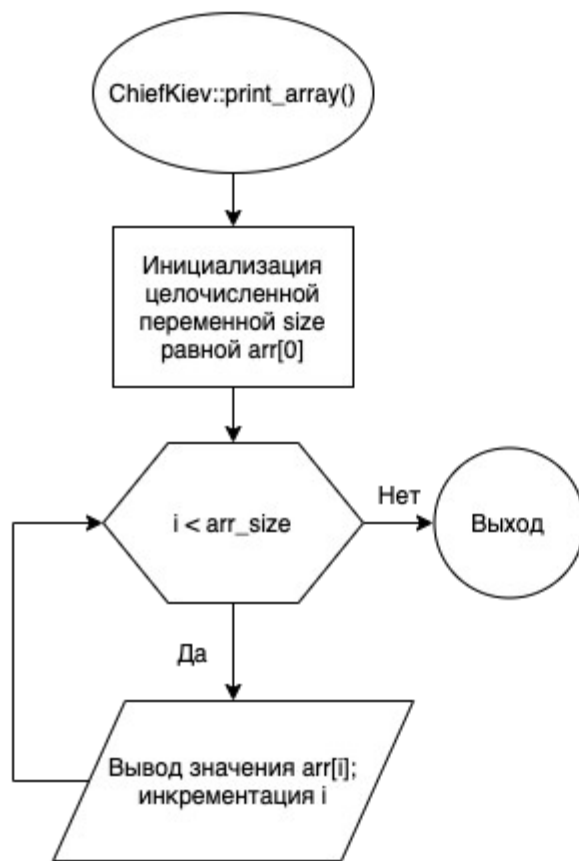


Рисунок 2 – Блок-схема алгоритма



**Рисунок 3 – Блок-схема алгоритма**

## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл ChiefKiev.cpp

*Листинг 1 – ChiefKiev.cpp*

```
#include "ChiefKiev.h"
#include <iostream>

ChiefKiev::ChiefKiev(int arr_size) {
    arr = new int[arr_size];

    for (int i = 0; i < arr_size; i++) {
        arr[i] = arr_size;
    }
}

void ChiefKiev::print_array() {
    int size = arr[0];

    for (int i = 0; i < size; i++) {
        if (i != size - 1) {
            std::cout << arr[i] << " ";
        } else {
            std::cout << arr[i];
        }
    }
}

void ChiefKiev::set_array(int* p_array) {
    arr = p_array;
}

int* ChiefKiev::get_array() {
    return arr;
}

ChiefKiev::~~ChiefKiev() {
    if (arr != nullptr) {
        delete[] arr;
    }
}
```

## 5.2 Файл ChiefKiev.h

Листинг 2 – ChiefKiev.h

```
#ifndef __CHIEFKIEV__H
#define __CHIEFKIEV__H

class ChiefKiev {
private:
    int *arr;
public:

    ChiefKiev(int arr_size);

    ~ChiefKiev();

    void print_array();

    void set_array(int* p_array);

    int* get_array();

};

#endif
```

## 5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <iostream>
#include "ChiefKiev.h"

int main() {
    int i;
    std::cin >> i;

    if (i <= 4) {
        std::cout << i << "?" << std::endl;
        return 0;
    }

    ChiefKiev lovesosa(i);

    std::cin >> i;

    if (i <= 4) {
        std::cout << i << "?" << std::endl;
```

```
        return 0;
    }

    ChiefKiev maybebaby(i);
    int *temp = lovesosa.get_array();
    lovesosa = maybebaby;
    lovesosa.set_array(temp);

    lovesosa.print_array();
    std::cout << std::endl;
    maybebaby.print_array();

    return 0;
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 7.

Таблица 7 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
1	1?	1?
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](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](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).