Институт информационных технологий и управления

в технических системах

Кафедра информационных технологий и компьютерных систем

ОТЧЕТ

по лабораторной работе № 10

**«РАЗРАБОТКА РЕКУРСИВНЫХ И ИТЕРАЦИОННЫХ АЛГОРИТМОВ. КЛАССЫ – БИБЛИОТЕКИ**

**СТАТИЧЕСКИХ МЕТОДОВ»**

по дисциплине «Программирование. Базовые процедуры обработки информации»

Выполнил студент группы ИВТ/б-11д

 Орлов И.В.

Проверил доцент Петров И.И.

Севастополь

2020

**1. ЦЕЛЬ РАБОТЫ**

Целью данной работы является исследование и сравнение рекурсивных и итерационных алгоритмов обработки одномерных массивов, получение навыков в создании классов, являющихся библиотеками статических методов.

2. **ПОСТАНОВКА ЗАДАЧИ**

1) Ознакомиться с теоретическими сведениями, приведенными в пункте 2.4 методических указаний: рекурсивные и итерационные алгоритмы, виды классов и отличия между ними, статические члены класса и ограничения в их использовании.

2) Разработать программу, реализующую обработку массива данных по заданию, указанному в таблице 2.1, в соответствии с номером варианта. Программа должна удовлетворять нижеперечисленным требованиям.

Программа должна содержать два класса: первый класс, содержащий метод main() (стартовая точка программы), и второй класс, содержащий два статических метода обработки одномерных массивов (рекурсивный и итерационный), заданные в таблице 2.1, а также статический метод вывода одномерного массива в окно терминала. Второй класс будет выполнять роль библиотеки алгоритмов.

В методе main() должна быть предусмотрена инициализация различных массивов для полной проверки разработанных методов обработки массивов (набор тестов), соответствующие вызовы методов обработки массивов и вывод результатов их работы.

Вариант задания V вычислен по формуле

 V = (N%14 != 0) ? N%14 : 14; ,

где N – номер студента в списке группы:

1%14=1;

Данные варианта задания приведены в таблице 2.1.

Таблица 2.1 – Вариант задания

|  |  |  |
| --- | --- | --- |
| НВ | Рекурсивная функция | Итерационная функция |
| 1 | Сумма элементов, значение которых больше A. | Количество нулевых элементов. |

**3. АНАЛИЗ ЗАДАЧИ**

**3.1. Входные данные, выходные данные, функции**

**и структура программы**

Согласно заданию, основная программа (метод main) должна инициализировать несколько массивов, а потом для каждого из них последовательно вызвать два метода обработки – итерационный и рекурсивный.

Результаты обработки массивов двумя методами будут выводиться на терминал. Для этого нужно предусмотреть соответствующий метод вывода элементов массива.

Указанные три метода по условию задачи должны быть определены в отдельном классе. Метод main (статический) должен обращаться к методам из другого класса напрямую (т.е. без создания объектов данного класса), поэтому методы должны быть объявлены статическими. Статические методы являются принадлежностью класса, а не объекта, запускаются от имени класса, не работают с переменными экземпляра, заданными в классе. Вся информация, необходимая для работы статического метода передается ему через параметры. Класс, в котором будут заданы три статических метода (переменные экземпляра в этом классе задаваться не будут) будем условно считать примером класса-библиотеки статических алгоритмов.

**3.2. Входные данные, выходные данные, выполняемые действия**

**для вызываемых методов**

Первый метод – method1 – предназначен для вычисления значения суммы элементов массива arr, значение которых больше А. По заданию он должен быть реализован как функция. Метод должен реализовывать рекурсивный алгоритм. Схема алгоритма изображена на рисунке 3.1.

Метод имеет четыре параметра.

 Первый формальный параметр метода: arr – локальная ссылочная переменная (ссылка на одномерный массив элементов, для которых выбран тип int). При вызове метода переменной arr будет присвоено значение соответствующего фактического параметра, т.е. адрес массива, созданного в вызывающем методе (main). Таким образом, передача параметра методу осуществляется по ссылке.

Второй формальный параметр (обязательный для рекурсивного метода) – start – номер элемента, с которого начинается просмотр массива (имеет тип int). При рекурсивных вызовах метода, в каждом из которых значение этого параметра увеличивается на 1, рано или поздно должен осуществиться нерекурсивный случай, в котором рассматриваемый массив будет состоять из одного элемента, и для которого будет вычислена функция с помощью простой операции (сумме присваивается значение элемента массива, если оно больше A, в противном случае – 0). После этого осуществится серия возвратов из метода, в процессе которых каждый раз к сумме будет прибавляться очередной элемент, если его значение больше заданного.

Третий формальный параметр – end (имеет тип int) – ограничивает число рассматриваемых элементов массива. Этот формальный параметр в принципе можно исключить из списка параметров, если требуется вычислять сумму для всех элементов массива.

Четвертый формальный параметр – А (типа int) – значение, с которым будут сравниваться элементы массива.

Передача аргументов (фактических параметров) второму, третьему и четвертому формальным параметрам при вызове метода будет осуществляться по значению, т.к. это параметры простого типа.

Метод возвращает значение типа int.

Второй метод – method2 – предназначен для вычисления числа нулевых элементов массива на основе итерационного алгоритма. Схема алгоритма изображена на рисунке 3.2. По заданию метод должен быть реализован как функция. Формальный параметр метода: arr – локальная ссылочная переменная (ссылка на одномерный массив элементов типа int). При вызове метода переменной arr будет присвоено значение соответствующего фактического параметра, т.е. адрес массива, созданного в вызывающем методе (main). Таким образом, передача параметра методу осуществляется по ссылке.

По аналогии с предыдущим методом данный метод имеет еще два параметра типа int: start – номер элемента, с которого начинается просмотр массива и end – номер элемента, которым заканчивается просмотр массива. Для итерационного метода эти параметры не являются необходимыми, если требуется обрабатывать весь массив.

Тип возвращаемого значения – int.

Третий метод – putArr – предназначен для вывода элементов массива в одну строку. Формальный параметр метода: arr – локальная ссылочная переменная (ссылка на одномерный массив элементов типа int). При вызове метода переменной arr будет присвоено значение соответствующего фактического параметра, т.е. адрес массива, созданного в вызывающем методе (main). Таким образом, передача параметра методу осуществляется по ссылке. Метод является процедурой и не возвращает значения.

***Схема алгоритма должна быть изображена аккуратно и достаточно крупно***

Рисунок 3.1 – Схема алгоритма метода method1 (рекурсивный)

***Схема алгоритма должна быть изображена аккуратно и достаточно крупно***

 Рисунок 3.2 – Схема алгоритма метода method2 (итерационный)

**4. ТЕСТОВЫЕ ПРИМЕРЫ И РЕЗУЛЬТАТЫ ИХ ОБРАБОТКИ ВРУЧНУЮ**

*В данном пункте нужно привести примеры двух массивов и ожидаемые результаты их обработки двумя методами. Если есть частные случаи, в которых данные обрабатываемых массивов существенно влияют на результат обработки, например, в массиве нет элементов, больших А), нужно придумать тестовые примеры для таких частных случаев и привести ожидаемые результаты обработки для каждого тестового примера.*

**5. СТРУКТУРА ПРОЕКТА**

Структура проекта изображена на рисунке 5.1.



Рисунок 5.1 – Структура проекта

**6.ТЕКСТ ПРОГРАММЫ**

Согласно пункту 2.8.1. методических указаний к лабораторным работам, текст программы должен быть отформатирован следующим образом:

1. шрифт: arial, 12 пт, полужирный;
2. абзац:
* выравнивание по левому краю;
* уровень – основной текст;
* отступ слева – 0;
* отступ справа – 0;
* интервал перед – 0;
* интервал после – 0;
* первая строка – нет (отступа или выступа),
* междустрочный интервал – множитель, значение – 1,2;
1. комментарии (обязательно должны присутствовать в тексте программы) выделить курсивом и синим цветом.

**7. СВЕДЕНИЯ ОБ ОТЛАДКЕ ПРОГРАММЫ И ПРОВЕРКЕ**

**ЕЕ РАБОТОСПОСОБНОСТИ**

*В данном пункте сначала следует описать ошибки, которые возникли при отладке программы и способ их устранения.*

Результаты выполнения программы представлены на рисунке 7.1.

***Скриншот окна терминала (не мельчить!)***

 Рисунок 7.1 – Результаты выполнения программы

На тестовых примерах программа выдала ожидаемый результат, что позволяет сделать вывод о ее работоспособности.

**ВЫВОД**

В ходе выполнения лабораторной работы были изучены рекурсивный и итерационный принцип построения алгоритмов обработки массивов, разработана программа, состоящая из двух классов. Алгоритмы обработки массивов были определены в отдельном классе как статические методы, поэтому метод main мог обращаться к ним (вызывать их) напрямую (без создания каких-либо объектов). При обращении к этим методам из другого класса имя метода предварялось именем класса.

При программировании рекурсивного метода обязательно должен быть предусмотрен нерекурсивный случай (действия с массивом, состоящим из одного элемента), чтобы процесс рекурсивных вызовов одного и того же метода мог рано или поздно остановиться. С этой целью в списке параметров рекурсивного метода обязательно должен присутствовать параметр, значение которого равно номеру элемента, с которого начинается просмотр массива. Если использование рекурсии не дает выигрыша в быстродействии или длине кода, то целесообразнее использовать итерационный принцип при обработке массива, чтобы избежать ситуации переполнения системного стека при большой глубине рекурсивных вызовов.

***Уважаемые студенты! Обращаю ваше внимание на то, что шаблоны отчетов даются вам для того, чтобы вы научились писать технические отчеты и правильно составлять документацию к программам. Поэтому настоятельно прошу сначала внимательно прочитать текст шаблона, при необходимости творчески переработать его в соответствии с данными своего варианта и только затем вставлять результаты работы своей программы.***