

2023.6.25

Java二级学习

——学习笔记

猎奇人只爱无机盐

2023.7.12~2023. .

目录

[考试题型和分值 1](#_Toc10249)

[1.单选题（40分） 1](#_Toc24182)

[公共基础（10分） 1](#_Toc31667)

[专业知识（30分） 1](#_Toc23965)

[2.操作题（60分） 1](#_Toc22892)

[基本操作（如：填空等，18分） 1](#_Toc22024)

[简单应用题（如：改错或窗体应用等，24分） 1](#_Toc21004)

[综合应用题（如：Applet程序设计等，18分） 1](#_Toc677)

[公共基础 1](#_Toc21594)

[第一章、数据结构与算法 2](#_Toc109)

[1、概述 2](#_Toc30094)

[2、数据结构 2](#_Toc21329)

[3、算法 7](#_Toc18180)

[4、常见的数据结构 12](#_Toc14245)

[第二章 、程序设计基础 19](#_Toc27270)

[第三章 、软件工程基础 19](#_Toc19397)

[第四章、数据库设计基础 19](#_Toc25040)

[Java语言程序设计 19](#_Toc15371)

**java计算机二级**

# 考试题型和分值

## 1.单选题（40分）

### 公共基础（10分）

### 专业知识（30分）

## 2.操作题（60分）

### 基本操作（如：填空等，18分）

### 简单应用题（如：改错或窗体应用等，24分）

### 综合应用题（如：Applet程序设计等，18分）

# 公共基础

## 第一章、数据结构与算法

### 1、概述

* 用计算机解决现实生活与生产问题是计算机编程的核心意义
* 数据结构与算法正是为了高效的解决问题引入的两个概念
* 程序=算法+数据结构

• 数据结构是算法实现的基础

• 算法总是要依赖某种数据结构来实现的，算法的操作对象时数据结构

• 算法是编程思想，数据结构是这些思想的基础

### 2、数据结构

* 数据结构是计算机存储、组织数据的方式。
* 数据结构研究的三个方面：

• 1）数据集合中各数据之间固有的关系，即数据的逻辑结构；

• 2）对数据进行处理时，各元素在计算机中的存储关系，即数据的存储结构；

• 3）对各种数据结构的运算

* 数据结构的主要任务

• 是通过分析数据对象的结构特征，包括逻辑结构及数据对象之间的关系，然后把逻辑结构表示成计算机可实现的物理结构，从而便于计算机处理。

• 通常情况下，精心选择的数据结构可以带来更高的运行或者存储效率。

* 2.1、数据结构基本术语

• 1）数据(Data)：

• 是描述客观事物的符号，

• 是计算机中可以操作的对象，

• 是能被计算机识别，并输出给计算机处理的符号集合。

• 数据不仅仅包括整型、实型等数值类型，还包括字符及声音、图像、视频等非数值类型。

• 　2）数据元素（Data Element）：

• 数据的基本单位。

• 一个数据元素由若干数据项构成，在计算机中通常作为整体处理，也被称为记录。

• 　3）数据项（Data Item）：

• 数据项是数据不可分割的最小单位。

• 比如：数据(是一张表)->数据元素(行)->数据项(字段)。

• 　4）数据对象（Data Object）：

• 是性质相同的数据元素的集合，

• 是数据的子集。

• 5）数据类型（Data Type）：

• 是指一组性质相同的值的集合及定义在此集合上的一些操作的总称。

• 　6）数据结构（Data Structure）：

• 是数据的组织形式

• 是数据元素相互之间存在的一种或多种特定关系的数据元素的集合

* 2.2、数据结构的分类

• 2.2.1、逻辑结构

• 数据的逻辑结构，面向的是用户，是从具体问题抽象出来的数学模型，

• 是描述数据元素及其关系的数学特性的。

• 在任何问题中，数据元素都不会是孤立的，在它们之间存在着这样或那样的关系，这种数据元素之间存在的关系称为数据的逻辑结构。

• 分类

• 2.2.1.1、线性结构

• 在该结构中，数据元素除了同属于一个集合外，数据元素之间还存在着一对一的顺序关系。

• 每一个节点都有一个前驱和一个后继,表示为R={d1,d2...dn}，各节点按逻辑关系排列,形成一条'链'。

• 如果一个数据满足下面两个条件：

• 1）有且只有一个根节点；

• 2）每一个节点最多有一个前件，也最多有一个后件

• 则称该数据为线性结构。线性结构又叫线性表。

• 常见的线性结构包括：

• 串（字符串）、队列、链表和栈等。

• 2.2.1.2、非线性结构

• 如果一个数据结构不是线性结构，则称之为非线性结构。

• 树状结构：

• 树形结构中的数据元素之间存在一种一对多的层次关系，每一个元素最多有一个前驱,能够有多个后继,表示为(D,{R}),就像一个树干长了多个树枝具有分支。

• 图状结构：

• 该结构的数据元素之间存在着多对多的任意关系

• 图状结构也称为网状结构。

• 集合结构：

• 在集合结构中，数据元素之间的关系是“属于同一个集合”，表示为R{}。

• 数据元素之间除了同属一个集合外，不存在其他关系。

• 常见的非线性结构包括：

• 广义表，树结构，图结构，多维数组等。

• 2.2.2、存储结构

• 数据结构在计算机中的表示称为存储结构(或物理结构)

• 存储结构主要分为

• 顺序存储、链式存储、索引存储和散列(哈希)存储。

• 1）顺序存储（线性结构）：

• 数据被存储在一组连续的存储单元中，其数据间的逻辑关系和物理关系是一致的。

• 特点：易查难插（删）。

• 2）链式存储（线性结构）：

• 借助元素地址指针表示数据的逻辑结构,每一个元素都会包含指向下一个元素的指针

• 这种结构须要在节点上附加一个指针项,指出后继节点的位置

• 即每一个节点存储单元包含两个部分:[数据项,指针项]。

• 特点：易插（删）难查。

• 3）索引存储：

• 指存储在数据元素信息的同时还建立附加的索引表。

• 索引表中的所有索引项是有序排列的。

• 特点：查找效率高，但额外增加了空间的开销。

• 4）散列存储(哈希表)：

• 根据元素的关键字通过哈希（或散列）函数直接计算出一个值，并将这个值作为该元素的存储地址。

• 特点：查找速度快。

• 一个完整的存储结构必须能够存储数据元素,以及元素之间的逻辑关系。

• 2.2.3、数据的运算

• 数据的运算是指：施加在数据上的运算包括运算的定义和实现。

• 运算的定义是针对逻辑结构的，指出运算的功能；

• 运算的实现是针对存储结构的，指出运算的具体操作步骤。

• 常见的数据运算有：

• 插入（新增）、删除、修改（更新）、查找（检索）、排序等。

### 3、算法

* 概念：

• 算法，字面意思即计算方法，一个算法就是对特定问题求解步骤的一种描述。

• 算法规定了求解给定类型问题所需的全部处理步骤以及执行顺序,使得问题能在优先时间内机械的求解

• 再具体一点,算法是一段有穷的指令序列，算法必须能使用某种语言描述。

• 注：算法并不等于公式，公式是一种高度精简的算法，算法的形式可以比公式更复杂，解决的问题更加广泛。

* 3.1、算法特性

• 1）有穷性：

• 指算法在执行有限的步骤之后，自动结束而不会出现无限循环，并且每一个步骤在可接受的时间内完成；

• 2）确定性：

• 算法的每一步骤都具有确定的含义，不会出现二义性；

• 3）可行性：

• 算法的每一步都必须是可行的，也就是说，每一步都能够通过执行有限次数完成；

• 4）输入：

• 是指算法具有零个或多个输入；

• 5）输出：

• 是指算法至少有一个或多个输出；

* 3.2、算法的分类

• 1）根据应用分类：

• 按照算法的应用领域，可以分为

• 基本算法

• 数据结构相关算法

• 几何算法

• 图论算法

• 规划算法

• 数值分析算法

• 加密解密算法

• 排序算法

• 查找算法

• 并行算法

• 数值算法

• 等等

• 2）根据确定性分类：

• 确定性算法：

• 有限时间内完成，得到结果唯一；

• 非确定性算法：

• 有限时间内完成，得到结果不唯一，存在多值性。

• 3）根据算法的思路分类：

• 递推算法

• 递归算法

• 穷举算法

• 贪婪算法

• 分治算法

• 动态规划算法

• 迭代算法

• 等

* 3.3、算法的表示

• 一般有三种流程结构：

• 顺序结构

• 分支结构

• 循环结构

* 3.4、算法的性能

• 算法的效率作为判断算法优劣的标准。一个算法的优劣往往通过算法复杂度来衡量

• 分类：

• 时间复杂度

• 时间复杂度是指执行算法所需要的计算工作量

• 即通常所说的算法执行所需要耗费的时间，时间越短，算法越好。

• 算法的时间度量，记作：T（n）=O（f（n））。

• 它表示随问题规模n的增大，算法执行时间的增长率和f（n）的增长率相同。

• 常见的时间复杂度量级有：

•



• 空间复杂度

• 空间复杂度是指执行这个算法所需要的内存空间。

• 是对一个算法在运行过程中临时占用存储空间大小的量度，记做S(n)=O(f(n))。

• 其中，n为问题的规模，f(n)为语句关于n所占存储空间的函数。

• 一般而言，常常以算法执行时间作为算法优劣的主要衡量指标。

### 4、常见的数据结构

* 4.1、数组

• 4.1.1、概述

• 数组是一种聚合数据类型，它是将具有相同类型的若干变量有序地组织在一起的集合。

• 数组可以说是最基本的数据结构，在各种编程语言中都有对应。

• 一个数组可以分解为多个数组元素

• 按照数据元素的类型，数组可以分为

• 整型数组

• 字符型数组

• 浮点型数组

• 指针数组

• 结构数组

• 等

• 数组还可以有一维、二维(比较重要，大题必考)以及多维等表现形式。

• 4.1.2、数组运算

• 遍历：

• 遍历所有元素；

• 插入：

• 将一个或多个元素插入数组；

• 删除：

• 从数组中删除元素；

• 搜索：

• 在数组中搜索元素。您可以按元素的值或索引搜索元素；

• 更新：

• 在给定索引处更新现有元素的值。

• 4.1.3、数组的应用

• 用作构建其他数据结构的基础

• 例如数组列表，堆，哈希表，向量和矩阵；

• 用于不同的排序算法

• 例如插入排序，快速排序，冒泡排序和合并排序。

* 4.2、链表

• 4.2.1、概述

• 链表是一种顺序结构，由相互链接的线性顺序项目序列组成。

• 这种存储结构具有在物理上存在非连续的特点。

• 链表由一系列数据结点构成，每个数据结点包括数据域和指针域两部分。

• 其中，指针域保存了数据结构中下一个元素存放的地址。

• 链表结构中数据元素的逻辑顺序是通过链表中的指针链接次序来实现的。

• 链式存储方式既可用于表示线性结构，也可用于表示非线性结构。

• 4.2.2、链表分类

• 单链表：

• 只能沿正向遍历项目；

• 双链表：

• 可以在前进和后退方向上遍历项目。节点由一个称为上一个的附加指针组成，指向上一个节点；

• 循环链接列表：

• 链接列表，其中头的上一个指针指向尾部，尾号的下一个指针指向头；

• 静态链表：

• 线性存储结构的一种，兼顾顺序表和链表的优点，是顺序表和链表的升级；

• 静态链表的数据全部存储在数组中(顺序表)，但存储的位置是随机的。（因为物理位置连续的，逻辑位置不一定连续）

• 4.2.3、链表操作

• 搜索：

• 通过简单的线性搜索在给定的链表中找到键为k的第一个元素，并返回指向该元素的指针

• 插入：

• 在链表中插入一个元素x。

• 插入可以通过3种不同的方式完成

• 在列表的开头插入

• 在列表的末尾插入

• 在列表的中间插入

• 头结点用来记录链表的基地址，是链表遍历的起点

• 尾结点的后继指针不指向任何结点，而是指向一个空地址NULL。

• 删除：

• 从给定的链表中删除元素x。

• 删除可以通过3种不同方式完成

• 从列表的开头删除

• 从列表的末尾删除

• 从列表的中间删除

• 4.2.4、链表的应用

• 用于编译器设计中的符号表管理。

• 用于在使用Alt Tab（使用循环链表实现）的程序之间进行切换。

• 注：

• 数组无法动态扩容

• 如果申请空间不够，需要申请更大的空间，需要数据复制拷贝进入新的数组，非常耗时；

• 而链表天然支持动态扩容，因为他不要内存空间的连续性

• LinkedHashMap的底层实现就是用的双向链表结构。

• 多线程下Hashmap扩容时出现循环链表，jdk1.8把头插法换成了尾插法。

* 4.3、栈

• 4.3.1、概述

• 栈是一种特殊的线性表，

• 它只能在一个表的一个固定端进行数据结点的插入和删除操作。

• 栈按照"后进先出"或“先进后出”的原则来存储数据

• 先插入的数据将被压入栈底，

• 最后插入的数据在栈顶，

• 读出数据时，从栈顶开始逐个读出。

• 栈中没有数据时，称为空栈。

• 4.3.2、栈的顺序存储及其运算

• 栈的基本运算有3种：

• 入栈

• 入栈运算：在栈顶位置插入一个新元素；

• 退栈

• 退栈运算：取出栈顶元素并赋值给一个指定的变量；

• 读栈顶元素

• 读栈顶元素：将栈顶元素赋值给一个指定的变量；

* 4.4、队列

• 4.4.1、概述

• 队列和栈类似，也是一种特殊的线性表。

• 和栈不同的是，队列只允许在表的一端进行插入操作，而在另一端进行删除操作。

• 一般来说，进行插入操作的一端称为队尾，进行删除操作的一端称为队头。

• 队列中没有元素时，称为空队列。

• 4.4.2、队列操作

• 进队：

• 将元素插入队列的末尾。

• 出队：

• 从队列的开头删除元素。

• 4.4.3、队列的应用

• 用于管理多线程中的线程。

• 用于实施排队系统（例如：优先级队列）。

• 扩展理解：例题

• 循环队列的存储空间为 Q(1:40) ，初始状态为 front=rear=40 。经过一系列正常的入队与退队操作后， front=rear=15 ，此后又退出一个元素，则循环队列中的元素个数为（ ）。

• A、39，或0且产生下溢错误

• B、14

• C、40

• D、15

• 注：当队列front=rear=15时，此时队列可能满了或空了，再退出元素，可能是39，或0且产生下溢错误。

• 子主题 5



• 循环队列

• 基本概念：

• 队列是一种先进先出（FIFO）的线性表

• 循环队列即为队列的顺序表示和实现，

• 其优点主要是入队和出队的时间复杂度均为O（1）

• 这归功于“循环队列将存储结构的理解由序列型转换为环形”的思维方式

## 第二章 、程序设计基础

## 第三章 、软件工程基础

## 第四章、数据库设计基础

# Java语言程序设计