**第五章问题求解的基本思维**

* 计算学科是利用计算机进行问题求解的相关技术和理论的学科，问题求解的核心是算法和系统，算法和系统都可以通过计算机语言表达为机器可以理解和执行的程序。
* 计算机语言
  + 计算机语言是语法、语义与词汇的集合，它用来表达计算机程序。
  + 程序是指某种程序设计语言编制的、计算机能够执行的指令序列，表达的是让计算机求解问题的步骤和方法，计算机通过执行程序，进行问题求解，扩展计算机的功能，方便人们使用计算机。
  + 计算机语言的发展过程经历了机器语言、汇编语言、高级语言和构件化语言四个阶段。
    - 机器语言
      * 计算机的指令系统是指一组能够识别和执行的二进制编码表达的指令集合，使用二进制编码的指令编写程序的语言被称为机器语言。
    - 汇编语言
      * 汇编语言使用助记符来代替机器语言的指令码，使机器语言符号化，从而提高编码效率。
      * 使用汇编语言的助记符编写的程序称为汇编语言源程序。
      * 汇编语言源程序必须转换为机器语言程序才能够被计算机执行，汇编程序是一个编译器，用于将助记符与机器指令一一对应地翻译为机器语言程序。
      * 不同计算机的指令系统不同，因此汇编语言源程序与计算机系统有关，程序设计人员需要深入了解计算机硬件，汇编语言程序的通用性差。
    - 高级语言与编译器
      * 助记符书写的直观性差、程序的通用性差、编程繁琐。
      * 高级语言
        + 高级语言是类似于自然语言、以语句和函数为单位书写程序的编程语言，高级语言编写的程序被称为高级语言源程序。
        + 高级语言比较接近自然语言，直观、通用、易学、编程效率高。高级语言与机器无关，编程者不需要理解机器的硬件结构，程序易于移植。
      * 编译器
        + 编译器能将高级语言源程序翻译为可执行的机器语言程序。
        + 编译器先使用其编译程序将高级语言源程序转换为汇编语言源程序，再由汇编程序将汇编语言源程序转换为机器可执行的二进制语言程序。
    - 构件化语言
      * 使用高级语言编程时，需要一条一条语句地书写程序，编程效率不高，在开发复杂的大规模程序时较为困难。
      * 构件化语言开发环境中的每一个构件都是由一系列语句完成的复杂程序，能够完成一定功能，visualbasic（VB）语言开发环境中包括按钮、文本框、标签等控件，构件化语言是可视化的编程语言。
      * 在使用构件化语言编写程序时，只需要关心构件的布局、属性和功能，不需要关心构件本身的实现细节。
    - 编程语言的分层结构
      * 越高级的编程语言其结构越复杂，执行效率越低，对计算机软件、硬件系统的性能要求越高。
      * 越低级的编程语言其结构越简单，执行效率越高，对计算机软件、硬件系统的性能要求越低。
    - java虚拟机
      * java是一种面向对象的编程语言。
      * java源程序编译后会生成.class文件，称为字节码文件。
      * java虚拟机（javavirtualmachine，JVM）负责将字节码文件翻译成特定平台下的机器码然后运行，也就是说，只要在不同平台上安装对应的JVM，就可以运行java字节码文件。
      * java虚拟机有自己完善的硬件系统，如处理器、堆栈、寄存器等，还有相应的指令系统，java虚拟机屏蔽了与具体操作系统平台的相关信息，使得java程序只需生成java虚拟机上运行的目标代码（字节码），就额可以在多种平台上不加修改的运行。
* 程序设计基础
  + 程序设计的本质
    - 程序设计的本质是设计能够利用计算机的五个部件完成特定任务的指令序列。
  + 常量
    - 常量指在程序运行过程中值不能改变的量，通常是固定的数值或字符。
  + 变量
    - 在程序运行过程中，其值可以改变的量称为变量。
    - 变量占据内存中的一块存储单元，用来存放数据，存储单元中的数据可以改变，给存储单元起的名字，就是变量名，在存储单元中存放的数据就是变量的值。
  + 算术运算符
    - 算术运算符的作用是进行算术运算，用算术运算符将运算对象连接起来的表达式称为算术表达式。
    - 运算符
      * ^
        + 乘方

1级

* + - * -
        + 负号

2级

* + - * \*
        + 乘

3级

* + - * /
        + 除

3级

* + - * \
        + 整除

4级

* + - * Mod
        + 取模

5级

* + - * +
        + 加

6级

* + - * -
        + 减

6级

* + 关系运算符
    - 关系运算符用于比较两个操作数的关系，用关系运算符链接两个表达式称为关系表达式，若关系成立，则表达式值为真（TRUE），否则为假（FALSE）。
    - 关系运算符的操作数可以是数值、字符串等数据。
    - 运算符
      * =
        + 等于
      * >
        + 大于
        + 小于
      * >=
        + 大于等于
      * =
        + 小于等于
      * >
        + 不等于
  + 逻辑运算符
    - 逻辑运算符用于对操作数进行逻辑运算，用逻辑运算符链接关系表达式或逻辑值称为逻辑表达式，逻辑表达式的结果为TRUE或FALSE。
    - 运算符
      * not
        + 取反
      * and
        + 与
      * or
        + 或
      * xor
        + 异或
* 算法
  + 在编写程序之前，首先要找出解决问题的方法，并将其转换成计算机能够理解并执行的步骤，即算法。
  + 什么是算法
    - 程序=算法+数据结构
      * 算法是解决一个问题所采取的一系列步骤。
      * 数据结构是指程序中数据的类型和组织形式。
    - 算法给出了解决问题的方法和步骤，是程序的灵魂，决定如何操作数据，如何解决问题。
    - 同一个问题可以有多种不同算法。
  + 算法举例
    - 计算机程序的算法，必须是计算机能够运行的方法。
    - 判断算法是否正确的方法：跟踪上述算法的执行过程，理解变量的作用、程序设计时可用的部件和功能，验证算法的正确性。
  + 算法的表示
    - 算法的表示方法有很多种，常用的有自然语言、伪代码、传统流程图、N-S流程图等。
    - 自然语言
      * 拖沓冗长，容易出现歧义。
    - 伪代码
      * 介于自然语言和计算机语言之间的文字和符号来描述算法。
      * 描述方法比较灵活，修改方便，易于转变为程序，但是当情况比较复杂时，不够直观，而且而容易出现逻辑错误。
    - 传统流程图
      * 比较直观，它使用一些图框来表示各种操作，用箭头表示语句的执行顺序。
    - N-S流程图
      * N-S流程图又称盒图，其中所有结构都用方框表示，挥之方便安避免了使用箭头任意跳转所造成的混乱，更加符合结构化程序设计的原则。
    - 算法应该具有的特性
      * 有穷性
        + 算法中的每一个步骤都是确定的，不能含糊、模棱两可，算法中的每一个步骤不应当被解释为多种含义。
      * 输入
        + 算法可以有多个输入，也可以没有输入。
      * 输出
        + 算法必须有一个或多个输出，用于显示程序运行的结果。
      * 可行性
        + 算法中的每一个步骤都是可以执行的，都能得到确定的结果，而不能无法执行。
  + 算法类问题
    - 算法类问题求解的第一步是数学建模，数学建模是一种基于数学的思维方式，运用数学的语言和方法，通过抽象和简化建立对实际问题的描述和定义，数学模型将现实世界的问题抽象成数学模型，可以发现其本质以及能否求解，找到求解问题的方法和算法。
    - 哥尼斯堡七桥问题
      * 数学建模
        + 去除哥尼斯堡七桥问题的无关语义，将其抽象成由节点和连接节点的边构成的图。
      * 一笔画问题
        + 七桥问题所构成的图中，没有一个节点含有偶数条边，所以哥尼斯堡七桥问题无解。
    - 旅行商问题
      * TSP（travelingsalesmanproblem）是最具代表性的组合优化问题。
      * 数学建模
        + 假定有N个城市，任意两个城市之间的距离为T，TSP的本质是寻找城市的访问顺序，求最小的T和，采用遍历策略。
        + 遍历策略对于小规模的TSP是有效的，但是对于大规模的TSP，N个城市的组合路线为（N-1）！，其组合路径数随着城市数目的增长呈爆炸式增长。
      * 简化求解方法
        + 贪心算法

贪心算法策略的基本思想是：一定做出当前状况的最好选择。

贪心算法求解的并非最优解，而是可行解。

* + - * + 分治法
        + 动态规划
        + 启发式算法
  + 算法分析
    - 对于求解一个问题的算法，需要分析其正确性和复杂性
    - 算法的正确性
      * 算法的正确性是指问题求解的过程、方法是否正确，输出结果是否正确。
      * 算法的正确性相对易于分析，只需要考察计算的结果就可以了。
    - 算法的复杂性
      * 算法的时间复杂性
        + 算法的时间复杂性，指的是算法运行所需的时间，如果一个问题的规模为n，算法运行的时间记为T（n）。
        + 常用O记法表示算法的时间复杂性，O表示其量级。
      * 算法的空间复杂性
        + 算法的空间复杂性，指的是算法在执行过程中所占用的存储空间的大小，用S（n）表示。
* 算法设计
  + 顺序结构
    - 顺序结构按照语句的先后顺序执行程序，它是程序设计中最简单的控制结构，它一般包括输入、处理和输出三个步骤。
  + 选择结构
    - 选择结构用于判断给定的条件，根据判断的结果来控制程序的流程。
  + 循环结构
    - 循环结构是用于实现同一段程序多次执行的一种控制结构。
    - For…Next循环是计数型循环，主要用于循环次数已知的情况，它的本质是当型循环。
    - 穷举法又称枚举法，它的基本思路就是一一列举所有的可能性，逐个进行排查，穷举法的核心是找出问题的所有可能解，并针对每种可能逐个进行判断，最终找出问题的解。
  + 数组
    - 数组，也称线性表（LinearList），是一种数据结构，一个数组是n个具有相同特性的数据元素的有限序列。
    - 数组的相邻元素之间存在着序偶关系。
    - 数组的处理实际上就是处理数组元素的过程，按顺序对每个数组元素进行处理的过程称为数组的遍历。
      * 搜索
        + 在数组中，查找满足条件的所有元素，并求和、计数等。
        + 算法设计：遍历数组元素的过程中，判断每个元素是否满足条件，如满足条件则作出处理。
      * 求最大值
        + 算法的基本思想是使用变量max，先将第0个元素赋给max，然后遍历整个数组，max与每一个元素比较，如果maxa[i]，则使max=a[i]，这样就可以保证max中存放的是最大的数。
      * 排序
        + 排序是最常见的问题，其本质是对一组对象按照某种规则进行有序排列的过程。
        + 在计算科学中，排序是许多复杂问题求解的基础，如数据库查询、数据挖掘等，通过排序可以有效降低问题求解算法的执行时间。
        + 算法

冒泡法

每轮依次比较两个相邻元素，如果前大后小则交换。

选择法

每轮次找到最小值，做一次交换。

两种算法的时间复杂度都未O（n^2）

快速排序

从序列中任意取出一个元素作为中心，所有比它小的都放在左侧，比它大的都放在右侧，形成左右两个序列；再对各子序列重新选择中心元素并按照此规则调整；直到每个子序列只剩下一个元素。

插入排序

基数排序

桶排序

堆排序

* 函数与递归
  + 函数
    - 函数是由多条语句组成的能够实现特定功能的程序段，函数可以对程序进行模块化。
    - 函数一般包括
      * 函数名
      * 参数
      * 返回值
      * 函数体
  + 递归
    - 递归算法的基本思想是将一个大规模的复杂问题，层层转换为一个与原问题相同但是规模较小问题来求解，函数调用函数本身、高阶调用低阶。
    - 使用递归的方法进行问题求解的基础是构造递归函数。
    - 递归过程可以总结为两个阶段
      * 回推阶段
      * 递推阶段
* 程序设计
  + 类和对象
    - 对象是实体的逻辑模型，一个实体就是一个对象。
    - 类是将多个对象共有的特征抽取出来，形成这些对象的抽象模型。
    - 类对象的抽象，而对象是类的实例。
    - 对象
      * 属性
        + 属性是对象的性质，即用来描述和反映对象特征的参数。
        + 不同类的对象有不同属性，同一类的不同对象的同一个属性可以有不同值。
      * 方法
        + 对象自身可以进行的动作或行为，是对象自身包含的功能。
      * 事件
        + 预先设置好的，可以被对象触发的动作。
        + 只要用户设计好了某个事件的代码，对象在响应了该事件后，就会执行相应代码。
  + visualbasic编程
    - 类
      * VB中类可以由系统提供。
    - 对象
      * 用户在窗体上放置一个控件就是创建该控件类的一个对象。
    - 属性
      * VB中的对象都有自己的属性，常见的属性有标题、名称、字体等。
    - 事件过程
      * 为对象的事件编写程序。
    - 程序调试
      * 掌握正确的调试方法，能够迅速、有效地发现和纠正程序错误。
      * 调试过程的常用办法是：逐语句执行程序，观察循环结构程序的控制流程，并使用本地窗口，观察变量的变化。