**第四章：程序设计导引**

程序设计概述：

1. 编程的目的：让计算机帮助我们解决问题；
2. 程序的由来：算法—解决问题的方法

高级语言（python）

1. 待处理的数据不是单纯的数值，程序十分庞大；
2. 基本数据结构

# 线性表：

概念：n个数据元素的有限序列；

数据元素之间仅有线性关系，每个数据元素只有一个直接前驱一个直接后继；

最常用且最简单

特点： 四个“唯一”（数据元素的非空有限集）

存在唯一的一个被称作“第一个”的数据元素；

存在唯一一个被称为“最后一个”的数据元素；

除第一个之外的数据元素均只有一个前驱

除最后一个之外的数据元素均只有一个后继。

## 数组

（1）、物理位置的连接关系隐含在位置中

（2）、优点：关系隐含不浪费

缺点：需挪动大量元素

（3）、插入：后继依次往后挪，长度加一

删除：后继依次往前挪，长度减一 （遍历）

## 链表

1. 数据Data

指针Next

1. 优点：不需挪动数据

缺点：占空间大

1. 插入：先挪后面的，判断插入位置是否合法

删除：从后向前覆盖 （遍历）

## 栈（逻辑操作受限）

1. 近限定在表尾进行插入或删除操作（只规定了插入删除的顺序，未规定进入的顺序）
2. 后进先出LIFO结构（Last in first out）

## 队列

1. 限定在表的一端插入，另一端删除
2. 先进先出FIFO结构（First in first out）

# 树形结构：

## 1、概念：

数据元素之间有明显的层次关系，并且每一层上的数据元素可能和下一层中多个元素（即其孩子结点）相关，但只能和上一层中一个元素（即其双亲结点）相关；

## 2、特点：“两个唯一”

除根节点外每个节点均有唯一前驱（有非唯一后继）

存在唯一的一个被称为“第一个”的数据元素（非唯一被称为“最后一个”的数据元素；叶子：没有后继的点）

## 3、双亲：直接前驱

**孩子**：直接后继 （**兄弟**：并列的孩子）

## 4、二叉树：支叉最多两个（每节点最多两个后继）

If分支：false/true；左孩子/右孩子（仅一个即左又右）

# 图形结构：

有若干顶点，结点之间的关系可以是任意的，图中任意两个数据元素之间都可能相关；

1. 程序设计方法

**导引：**程序复杂性：①功能复杂：小模块细化分解，内部实现。

②持续时间长

1. 研究问题的思路：自上而下地分解复杂问题；自下而上地解决问题。
2. **模块化方法（传统）**
3. 待开发软件划分成功能相对独立模块；
4. 各个模块之间分别单独开发、调试、运行、测试；
5. 多个模块组合起来整体测试。
6. **程序设计方法：**
7. **面向过程（结构化）**一次实现，精准到位。自上而下，逐步求精。

优点：模块易于识别，每个模块符合单入口、单出口。

（顺序结构。选择结构，循环结构）

1. **面向对象：**尽可能模仿现实中人类思维模式，接近人类解决问题的方 法和过程；**四个要点**：
   1. 对象是组成客观世界的基本元素（Kitty，Tom客观世界存在）
   2. 对象是属于某个类的（Kitty，Tom都是猫）
   3. 继承性（子类继承父类的行为和属性）
   4. 消息传递（狗叫引起人的注意）

（3）**基本手段—抽象**（选择性忽略，有动态过程：图书馆借书10-30）

**三大特征**：①封装：行为，属性放到一个class里

②继承：子类继承父类的属性（父类改变，子类均改）

③多态：动物叫声多态（同一个命令不同对象不同反应）

1. 软件生命周期

**1、软件生命周期**

（1）**定义阶段**：计划，需求分析

（2）**开发阶段**：设计，编码，测试

（3）**维护阶段**：运行和维护

**2**、可行性研究、需求分析、系统设计、编码实现、软件测试、部署、软件维护

**3、瀑布模型**

（1）**优点**：①提供按阶段划分的检查点；

②当前阶段完成后，只需关注后续阶段。

（2）**缺点**：①各阶段间极少有反馈

* 1. 项目生命后期才看见结果
  2. 通过强制完成日期和里程碑来跟踪各个项目阶段

④不适应用户需求变化