模块一：线性结构

连续存储【数组】

离散存储【链表】

线性结构的两种常见应用之一 栈

线性结构的两种常见应用之二 队列

专题：递归

1. 1+2+3+…+100的和
2. 求阶乘
3. 汉诺塔
4. 走迷宫

模块二：非线性结构

数

图

模块三：查找和排序

折半查找

排序：冒泡

插入

选择

快速排序

归并排序

Java中容器和数据结构相关知识

Iterator接口

Map

哈希表

数据结构概述：

定义：我们如何把现实中大量而复杂的问题以特定的数据类型和特定的存储结构保存到主存储器（内存）中，以及在此基础上为实现某个功能而执行的相应操作，这个相应的操作也叫算法。

（任何一个复杂的问题，要保存到计算机里分两块来保存，一个是独立的事物，一个是事物和其他事物的关系，两个问题解决数据的存储就解决了。）

数据结构 = 个体 + 个体的关系

算法 = 对存储数据的操作

算法：解题的方法和步骤

衡量算法的标准

1. 时间复杂度：大概程序要执行的次数，而非执行的时间
2. 空间复杂度：算法执行过程中大概所占用的最大内存
3. 难易程度
4. 健壮性

数据结构的地位

数据结构是软件中最核心的课程

程序 = 数据的存储 + 数据的操作 +可以被计算机执行的语言

{字段：反映属性

记录：反映一个事物

表：反映事物的集合}

堆是用来动态分配内存的，栈是用来存储局部变量和函数调用的。栈上的对象是自动管理的，函数结束时会自动销毁，而堆上的对象需要程序员用 delete 手动释放。

预备知识

指针

指针的重要性：指针是C语言的灵魂。

定义：地址是内存单元的编号，从0开始的非负整数，0-FFFFFFFF[4G-1]

指针就是地址，地址就是指针。指针变量是存放内存单元地址的变量

指针的本质就是一个操作首先的非负整数

分类：1.基本类型的指针。

2.指针和数组的关系

结构体

为什么有结构体：

为了表示一些复杂的数据，而普通的基本类型变量无法满足要求

什么叫结构体：

结构体是用户根据实际需要自己定义的符合数据类型

如何使用结构体：

两种方式

struct Student st = {1000,”zhangsan”,20};

struct Student \* stp = &st;

// st.sid = 99第一种方式

//pst -＞sid = 99第二种方式，pst所指向的结构体变量中的sid这个成员

注意事项：

结构体变量不能加减乘除，但可以相互赋值

Cpu控制内存靠三根线：地制线、控制线、数据线

int main(void)

{

struct Student st = {1000,”zhangsan”,20};

//st.sid = 99第一种方式

struct Student \* pst;

pst = &st;

pst -＞sid = 99; // pst -＞sid等价于(\*pst).sid

return 0;

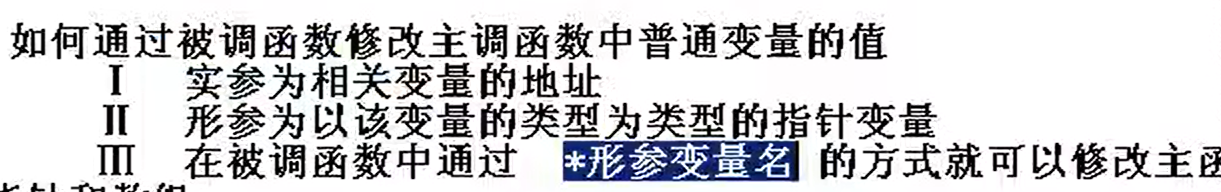
}

动态内存的分配和释放：动态分配的内存必须手动释放。

int \*p = &i; //等价于int \*p; p = &i;即p（指针变量）等于取地址i

p存放了i的地址，\*p就是p地址对应的内容，即\*p=i; 但p和i是完全不同的两个变量，修改p的值不影响i的值。如果一个指针变量存放了某个普通变量，则可以说：“p指向了i“，则\*指针变量 就完全等价于 该普通变量。只有指针变量前面可以加\*，且常量和表达式前不能加&（取地址符）

int \* p; //p是个变量名字，int \* 表示该p变量只能存储int类型变量的地址



#include [stdio.h]

Void f(int \* p)//不是定义了一个名字为\*p的形参，而是定义了一个形参，该形参名字叫做p，它的类型是int \*。

{

\*p = 100

}

Int main(void)

{

Int I =9;

F(&i)

Printf(“i=%d/n”,i);

Return 0;

}

int a[5] = {1,2,3,4,5}里a存放了第一个元素即a[0]（即1）的地址。a指向了a[0]

a[i]等价于\*(a+i) // a + i 表示数组首地址 a 向后偏移 i 个元素的地址。a 的值是 &a[0]。一个整数占4个字节。\*a就是a指向的元素，a指向的元素就是a[0]。

