1.3 抽象数据类型的表示和实现

#数据结构

复习前两节:

计算机领域

- 数值计算领域
- 非数值计算领域 (越来越多)

导致需要被处理的数据不再是存粹的数值 而是 字符,表哥,图像 ...

那么这些新的,杂乱无章的,具有一定结构的数据该如何处理呢?

- 找出数据的逻辑关系 逻辑结构
- 存入计算机 存储结构
- 有效的运算 算法

概念小结

- 数据 的个体是 数据元素
- 数据元素中性质相同的构成的集合 叫做数据对象
- 数据元素加上数据元素之间的关系 就叫做数据结构
- 数据结构中的逻辑模型叫做逻辑结构
- 把数据结构映像到内存就是存储结构
- 对数据结构加上操作就是抽象数据类型

图解: (来源于Bilibili课程)



用C语言来实现抽象数据类型

- 用已有的数据结构类型描述存储结构
- 用函数定义描述他的操作

抽象数据类型如何实现

可以通过固有的数据类型来表示和实现 用已经存在的操作来实现新的操作

<mark>注意</mark>在本门课程中使用<mark>类C语言</mark>(介于伪代码和C语言之间) 类C语言进附录

- 跟 C 语言非常的像,但是缺少语法细节,使得语言自述能力上升,但是实际上不能通过编译
- 真正上机的时候必须要使用正确的语言

例子: 用抽象数据类型"复数"的实现

```
typedef struct {
   float realpart;
   float imagpart;
} Complex;

void assign(Complex * A,float real,float imag);
void add(Complex * A,float real,float imag);
void minus(Complex * A,float real,float imag);
void multiply(Complex * A,float real,float imag);
void divide(Complex * A,float real,float imag);
void divide(Complex * A,float real,float imag);

void assign(Complex * A, float real, float imag){
   A -> realpart = real;
```

```
b -> imagpart = imag;
}

void add(Complex * c,Complex A,Complex B){
    c -> realpart = A.realpart + B.realpart;
    c -> imagpart = A.imagpart + B.imagpart;
}
```