Java数据结构基于数组的表Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/644/2021_2022_Java_E6_95_B0_E6_8D_AE_c104_644780.htm 我没看过 其他语言版的数据结构,但觉得java的实现方法很巧妙--用类和对象来实现.基于数组的表,思想很简单就是定义一个类用来存储一组数据,我定义的是ArrayListClass类,在类中定义用来操作数组的方法.其实就是 这么简单,但具体操作起来就会遇到很多麻烦了! 我们这个ArrayListClass类中首先应该包括一个数组型的域list,用来存放数据,这样放在同一数组中数据之间就产生了位置上的联系,使对数据的操作便的简单.然而这个数组到底是什么数据类型的,我们期望这个表能用于所有的数据类型,我们不能将他单纯的固定成某一种.所以我们必须将这个数据普通化,解决的办法就是定义一个类,作为所有数据类型的超类.看这个DataElement: public abstract class DataElement { public abstract boolean equals(DataElement otherElement). public abstract int compareTo(DataElement otherElement). public abstract void makeCopy(DataElement otherElement). public abstract DataElement getCopy(). } 将他定义成为抽象的,再在定义其他数据类型时继承并实现它,我定义了两个数据类型IntElement和StringElement: IntElement: public class IntElement extends DataElement { protected int num. //constructors public IntElement(){ num=0. } public IntElement(int number){ num=number. } public IntElement(IntElement otherElement){ num=otherElement.num. } ///get-set Methods public void setNum(int number){ num=number. } public int getNum(){ return

num. } /* (non-Javadoc) * @see DataElement#equals(DataElement) */ public boolean equals(DataElement otherElement) { // TODO Auto-generated method stub IntElement newe=(IntElement)otherElement. return (this.num==newe.num). } /* (non-Javadoc) * @see DataElement#compareTo(DataElement) */ public int compareTo(DataElement otherElement) { // TODO Auto-generated method stub IntElement newe=(IntElement)otherElement. if(this.num==newe.num) return 0. else if(this.num 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com