007.RandomAccessFile

Java还提供了专门处理文件内容的类，即RandomAccessFile（随机访问文件）类。
该类是Java语言中功能最为丰富的文件访问类，它提供了众多的文件访问方法。
RandomAccessFile类支持“随机访问”方式，这里“随机”是指可以跳转到文件的任意位置处读写数据。

RandomAccessFile对象类有个位置指示器，指向当前读写处的位置，
当前读写n个字节后，文件指示器将指向这n个字节后面的下一个字节处。
刚打开文件时，文件指示器指向文件的开头处，可以移动文件指示器到新的位置，随后的读写操作将从新的位置开始。

RandomAccessFile类在随机读取时有很大的优势，但该类仅限于操作文件，不能访问其他的I/O设备，如网络、内存映像等。

RandomAccessFile类的构造方法如下所示：

```
RandomAccessFile(File file ,  String mode)
//创建随机存储文件流，文件属性由参数File对象指定

RandomAccessFile(String name ,  String mode)
//创建随机存储文件流，文件名由参数name指定
```

这两个构造方法均涉及到一个String类型的参数mode，它决定随机存储文件流的操作模式，其中mode值及对应的含义：
“r”：以只读的方式打开，调用该对象的任何write（写）方法都会导致IOException异常
“rw”：以读、写方式打开，支持文件的读取或写入。若文件不存在，则创建之。
“rws”：以读、写方式打开，与“rw”不同的是，还要对文件内容或元数据的每次更新都同步更新到底层的存储设备中去。
“rwd”：以读、写方式打开，与“rw”不同的是，还要对文件内容的每次更新都同步更新到底层的存储设备中去。

任何文件系统中的数据分为数据和元数据。数据是指普通文件中的实际数据，即文件的实际内容；
而元数据指用来描述一个文件的特征的系统数据，诸如访问权限、文件拥有者以及文件数据块的分布信息(inode…)等等。

代码演示案例：

```
public class FileTest1 {
    public static void main(String[] args) throws Exception{
```

```
        Employee e1 = new Employee("zhangsan" , 23);
        Employee e2 = new Employee("lisi" , 24);
        Employee e3 = new Employee("wangwu" , 25);

        RandomAccessFile ra  = new RandomAccessFile("d:\\employee.txt" ,
"rw");
        ra.write(e1.name.getBytes());
        ra.writeInt(e1.age);
        ra.write(e2.name.getBytes());
        ra.writeInt(e2.age);
        ra.write(e3.name.getBytes());
        ra.writeInt(e3.age);
        ra.close();

        RandomAccessFile raf = new RandomAccessFile("d:\\employee.txt" ,
"r");
        int len = 8;
        raf.skipBytes(12);//跳过第一个员工的信息，其姓名8字节，年龄4字节
        System.out.println("第二个员工信息：");
        String str = "";
        for (int i = 0 ; i < len ; i++){
            str = str + (char)raf.readByte();
        }
        System.out.println("name:" + str);
        System.out.println("age:" + raf.readInt());

        System.out.println("第一个员工信息：");
        raf.seek(0);//将文件指针移动到文件开始位置
        str = "";
        for (int i = 0 ; i < len ; i++){
            str = str + (char)raf.readByte();
        }
        System.out.println("name:" + str);
        System.out.println("age:" + raf.readInt());

        System.out.println("第三个员工信息：");
        raf.skipBytes(12);//跳过第二个员工的信息
        str = "";
        for (int i = 0 ; i < len ; i++){
            str = str + (char)raf.readByte();
        }
        System.out.println("name:" + str);
        System.out.println("age:" + raf.readInt());
        raf.close();
    }
}
class Employee {
    String name;
    int age;
    final static int LEN = 8;

    public Employee(String name, int age) {
        if (name.length() > LEN) {
```

```
            name = name.substring(0, 8);
        } else {
            while (name.length() < LEN) {
                name = name + "\u0000";   // Unicode编码 \u0000表示空格
            }
            this.name = name;
            this.age = age;
        }
    }
}
```