

File类

第1节 【文件】

- 知识点1: File类的相关概念
- 知识点2: File类的构造方法
- 知识点3: File类的常用方法



知识点1: File类的相关概念

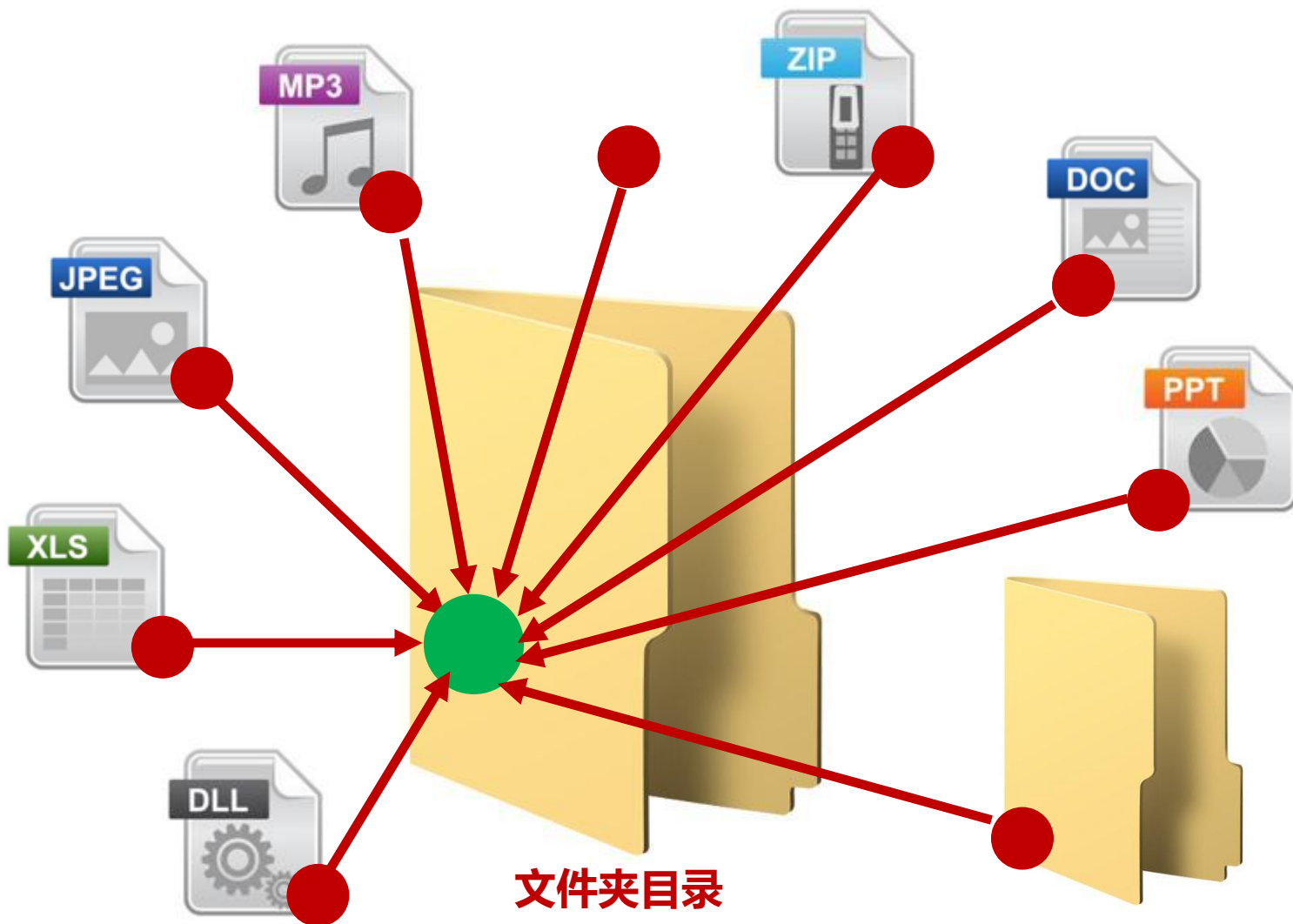


- 什么是文件?
 - 文件是存储**数据的集合**
 - 文件夹是存文件的集合
- 文件一般是存放在存储设备上的，例如：硬盘、光盘和移动存储设备等等



知识点1: File类的相关概念

- 文件系统的一般文件组织形式:



文件系统中由
文件夹目录和
数据文件构建
成一颗树

知识点1: File类的相关概念



- JDK的java.io包，其中包含一系列对文件和目录的属性进行操作，对文件进行读写操作的类；
- 程序中如果要使用到该包中的类，对文件进行操作，则必须显式地声明如下语句：

```
import java.io.*;
```

知识点1: File类的相关概念

- java.io.File类的**对象可以表示文件和目录**，在程序中一个File类对象可以**代表一个文件或目录**
- 当创建一个File对象后，就可以利用它来**对文件或目录的属性进行操作**，如：文件名、最后修改日期、文件大小等等
- 注意：File对象**并不能直接对文件内容进行读/写操作**，只能查看文件的属性

知识点2：File类的构造方法



- File类的构造方法有4种重载方式，常用的如下：

构造方法	说明
<code>File(String pathname)</code>	指定文件（或目录）名和路径创建文件对象

- 如：

```
File f1 = new File("chinasofti.txt");  
File f2 = new File("D:\\Java\\Hello.java");
```

对当前项目根目录中的chinasofti.tx
构建了一个File对象

通过绝对路径构建File对象

- 请注意：双斜线或用反斜杠
- File 类里个属性public static final String separator：得到与系统相关的默认分隔符字符

知识点2：File类的构造方法

- 构建File对象是需要注意的要点：
- 提供给构造方法的路径可以指向一个具体的文件，这时候File对象能够操作这个文件的属性，也可以指向一个文件夹，这时候File对象操作的就是文件夹的属性
- 注意上例第二个对象的路径表达，由于在Java中“\”符号表示转意，因此如果使用“\”作为路径分割符，则实际需要编写“\\”，当然一个更好的替代方法是使用Unix系统中常用的“/”作为路径分割，则不需要转意
- 特别注意，Java中的相对路径体系和我们日常所见的文件系统相对路径体系有较大的区别：
 - 如果以路径以“/”或“\\”开头，则相对路径的根为**当前项目所在磁盘的根目录**（Unix没有磁盘分区概念，因此直接使用/，即文件系统的根作为相对路径的根）
 - 如果不以“/”开头则**相对路径的根为项目根目录，而不是当前类所在目录**，这一点非常容易引起误区，因为类从属于某个包之后，类文件实际是位于项目中的某个子文件夹中的，如com.chinasoft.Hello这个类是位于项目中的com\chinasoft\子文件夹中，如果在Hello类中构建一个File对象：File f = new File("icss/chinasoft.txt")，那么这个文件位于项目根目录的icss子文件中，跟当前类自己的位置无关

知识点2: File类的构造方法



- 构建File对象是需要注意的要点:

```
File file =new File("d:\\Test.txt");
```

等价的 \\ 可以写成/

```
File file =new File("d:/Test.txt");
```

文件创建在d盘下

```
File file =new File("d:"+File.separator+"Test.txt");  
file.createNewFile();
```

自动选择当前分隔符

```
File file =new File("/Test.txt"); //当前项目盘符的根目录, 如 D:\  
file.createNewFile();
```

```
File file =new File("d:"+File.separator+"Test.txt");  
file.createNewFile();
```

```
File file =new File("Test.txt");//当前项目的根目录与src 同级  
file.createNewFile();
```

知识点3：File类的常用方法

例1

- File类型提供的常见操作方法：

获取文件信息：

方法原型	说 明
String getName()	获得文件的名称
String getAbsolutePath()	获得文件的绝对路径
long length()	获得文件的长度（字节数）
public long lastModified()	返回此抽象路径名表示的文件上次修改的时间

知识点3：File类的常用方法

例2

- File类型提供的常见操作方法：
- 创建与删除（文件，文件夹）

方法原型	说 明
<code>boolean createNewFile() throws IOException</code>	创建新文件，创建成功返回true，否则返回false，若文件不存 则创建， 如果文件存在则不创建
<code>public boolean mkdir()</code>	创建由此抽象路径名命名的目录。（只能创建单层）
<code>public boolean mkdirs()</code>	创建由此抽象路径名命名的目录，（可以创建多层）
<code>boolean delete()</code>	删除文件，删除成功返回true，否则返回false，如果文件夹里有文件 则删除失败返回false

- 注意：不能在创建文件时创建文件夹，也不能在创建文件夹时创建文件

当删除多层文件夹时，直接删除最外层，返回false，要先从里向外删除

知识点3：File类的常用方法

例3

- File类型提供的常见操作方法：
- 判断

方 法 原 型	说 明
boolean exists()	判断文件是否存在，存在返回true，否则返回false
boolean isFile()	判断是否为文件，是文件返回true，否则返回false
boolean isDirectory()	判断是否为目录，是目录返回true，否则返回false

知识点3：File类的常用方法

例4

- File类型提供的常见操作方法：
- **重命名**

方法原型	说 明
<code>public boolean renameTo(File dest)</code>	重命名由此抽象路径名表示的文件。

知识点3：File类的常用方法

例5

- File类型提供的常见操作方法：
- 获得盘符及盘符属性

方法原型	说 明
public long getTotalSpace()	盘符的总容量
public long getFreeSpace()	剩余空间
public long getUsableSpace()	可用空间

知识点3：File类的常用方法

例6

- File类型提供的常见操作方法：
- 获取文件夹内容

方法原型	说 明
<code>public String[] list()</code>	返回一个字符串数组，命名由此抽象路径名表示的目录中的文件和目录 如果访问的是c盘下的系统级文件夹，会报空指针 如果访问的是文件夹是空的，那返回的数组长度是0
<code>public File[] listFiles()</code>	返回文件夹内的子文件与子文件夹的数组

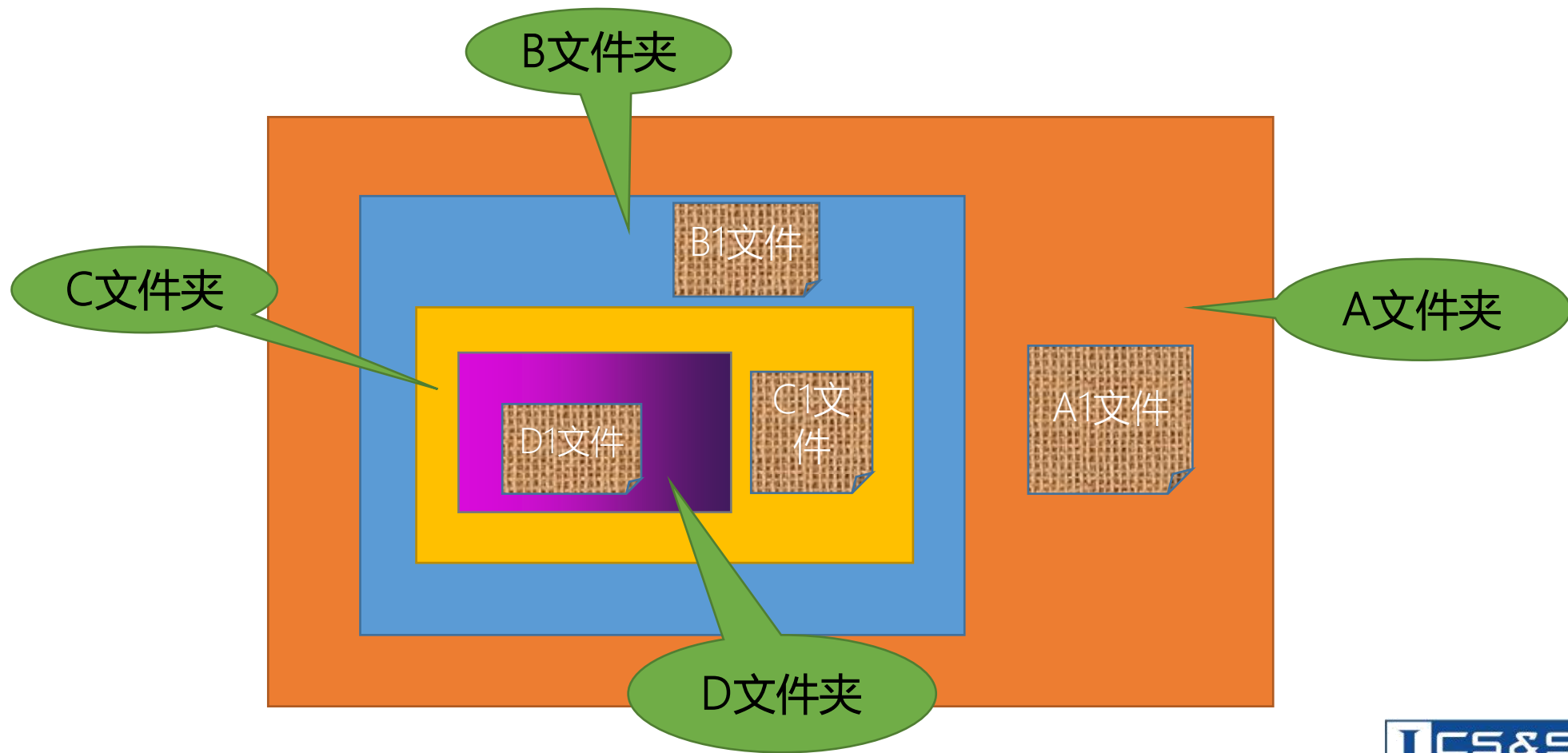
知识点3：File类的常用方法

例7.8

- File类型提供的常见操作方法：
- 对文件有过滤功能的方法

方法原型	说 明
<code>public File[] listFiles(FileFilter filter)</code>	文件过滤器
<code>public File[] listFiles(FilenameFilter filter)</code>	文件名过滤器

- 写方法，打印出A文件夹下面的所有文件及文件夹的名称：



本节总结提问【文件】



- File对象可不可以描述文件夹?
- Java中的相对路径有什么规则?
- 列出所有文件的方法?
- 带过滤功能的方法?

本节总结【文件】

- 在程序中一个File类对象可以代表一个文件或目录
- Java中的相对路径体系和我们日常所见的文件系统相对路径体系有较大的区别：如果以路径以“/”或“\”开头，则相对路径的根为当前项目所在磁盘的根目录（Unix没有磁盘分区概念，因此直接使用/，即文件系统的根作为相对路径的根），如果不以“/”开头则相对路径的根为项目根目录，而不是当前类所在目录，这一点非常容易引起误区，因为类从属于某个包之后，类文件实际是位于项目中的某个子文件夹中的，如com.chinasoft.Hello这个类是位于项目中的com\chinasoft子文件夹中，如果在Hello类中构建一个File对象：File f = new File("icss/chinasoft.txt")，那么这个文件位于项目根目录的icss子文件中，跟当前类自己的位置无关