第九天

1.File.separatorChar

**File.separatorChar:**

在 windows 中 文件文件分隔符 用 ' \ ' 或者 ' / ' 都可以

但是在 Linux 中，是不识别 ' \ ' 的，而 File.separator 是系统默认的文件分隔符号，在 UNIX 系统上，此字段的值为 ' / '

在 Microsoft Windows 系统上，它为 ' \ ' 屏蔽了这些系统的区别。

所以用 File.separator 保证了在任何系统下不会出错。

1. @Retention

@Retention 是一个Java注解，用于指定被注解的元素在编译后如何保存。它有以下三个保留策略（Retention Policies）：

RetentionPolicy.SOURCE: 注解仅在源代码中存在，编译时会被忽略，不会包含在生成的字节码文件中。这意味着该注解对运行时没有任何影响。

RetentionPolicy.CLASS: 注解在编译时会被保留到字节码文件中，但在运行时是不可见的。这是默认的保留策略，如果没有显式指定 @Retention 的值，默认使用该策略。

RetentionPolicy.RUNTIME: 注解在编译后仍然保留并可以在运行时通过反射机制获取到。这意味着可以在运行时对标记了该注解的元素进行动态的处理或操作。

1. @DDocumented

@Documented 是一个Java注解，用于指示被注解的元素是否应该包含在生成的API文档中。它是一个元注解，可以应用于自定义注解上。

当我们使用 @Documented 注解标记一个自定义注解时，表示该注解应该被包含在生成的API文档中，并且可以被工具和IDE等解析和展示给开发者。

以下是一个使用 @Documented 的示例：

java复制代码

import java.lang.annotation.\*;

@Documented@Retention(RetentionPolicy.RUNTIME)public @interface MyAnnotation {

// 自定义注解的内容

}

在上述示例中，@MyAnnotation 注解使用了 @Documented 注解，这意味着当使用该注解修饰某个元素（类、方法等）时，在生成的API文档中将包含有关这个注解的信息，以便开发者能够看到并了解如何使用它。

1. 文件类型和文件扩展名

文件类型（File Type）和文件拓展名（File Extension）是用于标识文件种类和格式的两个相关概念。

文件类型（File Type）： 文件类型指的是根据文件内容或结构进行分类的一种方式。它描述了文件的内部组织和数据格式。文件类型通常由文件的魔术数字（Magic Number）或者特定的文件头部信息来确定。不同的文件类型可能包括文本、图像、音频、视频、二进制等等。

例如，常见的文件类型有：

文本文件：包含纯文本内容，如 .txt 文件；

图像文件：存储图像数据，如 .png、.jpg 文件；

音频文件：存储音频数据，如 .mp3、.wav 文件；

视频文件：存储视频数据，如 .mp4、.avi 文件；

可执行文件：包含可执行代码，如 .exe、.jar 文件；

文件拓展名（File Extension）： 文件拓展名是与文件名相关联的后缀，用于表示文件的类型或格式。它通常是文件名的最后一个点后面的字符串。文件拓展名是操作系统或应用程序用来识别文件类型的一种方式。通过拓展名，操作系统可以关联合适的应用程序来打开文件。

例如：

file.txt 中的拓展名为 .txt，表示文本文件；

image.png 中的拓展名为 .png，表示图像文件；

audio.mp3 中的拓展名为 .mp3，表示音频文件；

请注意，文件拓展名只是一种常见的方式来推断文件类型，并不是绝对准确的。有时候，文件拓展名可以被修改或错误地设置，因此仅依靠拓展名可能会导致误判。在确定文件类型时，最好使用更可靠的方法，如魔术数字或文件头部信息。