

File 类

这里需要额外介绍一下File类，它是专门用于表示一个文件或文件夹，只不过它只是代表这个文件，但并不是这个文件本身。通过File对象，可以更好地管理和操作硬盘上的文件。

```
1 public static void main(String[] args) {
2     File file = new File("test.txt");    //直接创建文件对象，可以是相对路径，也可以是绝对
    路径
3     System.out.println(file.exists());    //此文件是否存在
4     System.out.println(file.length());    //获取文件的大小
5     System.out.println(file.isDirectory());    //是否为一个文件夹
6     System.out.println(file.canRead());    //是否可读
7     System.out.println(file.canWrite());    //是否可写
8     System.out.println(file.canExecute());    //是否可执行
9 }
```

通过File对象，我们就能快速得到文件的所有信息，如果是文件夹，还可以获取文件夹内部的文件列表等内容：

```
1 File file = new File("/");
2 System.out.println(Arrays.toString(file.list()));    //快速获取文件夹下的文件名称列表
3 for (File f : file.listFiles()){    //所有子文件的File对象
4     System.out.println(f.getAbsolutePath());    //获取文件的绝对路径
5 }
```

如果我们希望读取某个文件的内容，可以直接将File作为参数传入字节流或是字符流：

```
1 File file = new File("test.txt");
2 try (FileInputStream inputStream = new FileInputStream(file)){    //直接做参数
3     System.out.println(inputStream.available());
4 }catch (IOException e){
5     e.printStackTrace();
6 }
```

练习:

拷贝一个视频文件

```
1     public static void main(String[] args) {
2         File video = new File("test.mp4"); //创建文件系统
3         try(FileInputStream inputStream = new FileInputStream("test.mp4"); //创
        建字节流
4             FileOutputStream outputStream = new FileOutputStream("Copy.mp4")){
```

```
5         long total = video.length(); //计算文件大小
6         byte[] bytes = new byte[1024]; //每次复制的字节数
7         int len; //每次复制的大小
8         long sum=0; //计算一共复制了多少
9         while ((len = inputStream.read(bytes))!=-1){
10             outputStream.write(bytes,0,len); //复制
11             sum += len;
12             System.out.println("拷贝进度:"+(sum * 100 /total)+"%"); //进度条
13         }
14     }catch (IOException e){
15         throw new RuntimeException();
16     }
17 }
```