

# Java知识随笔

## 一、文件

```
// 创建:
boolean createNewFile();    // 创建文件
boolean mkdir();           // 创建文件夹
boolean mkdirs();          // 创建多级文件夹。
// 删除:
boolean delete();
void deleteOnExit(); // 在程序退出时删除文件。
// 判断:
boolean canExcute();       // 判断是否可执行
boolean exists();          // 文件是否存在。
boolean isFile();          // 文件
boolean isDirectory();     // 文件夹
boolean isHidden(); // java能得到文件中的隐藏文件但是对隐藏文件时不能访问的
boolean isAbsolute();      // 绝对路径即时不存在也能得到。
// 获取信息:
getName();
getPath();
getParent();
// 三种文件创建方式:
File file = new File("E:/..."); // 文件/文件夹路径对象
File file = new File("..." , "..."); // 父目录绝对路径 + 子目录名称
File file = new File("...", "..."); // 父目录File对象 + 子目录名称
// 获取当前文件夹
File directory = new File(""); // 设定为当前文件夹
String currentPath = System.getProperty("user.dir"); // 当前文件夹
```

```
public class MainTest {
    public static void main(String[] args) throws IOException {
        String currentPath = System.getProperty("user.dir");
        String directory = currentPath + File.separator + "blockingQueueFile";
        File dir = new File(directory);
        if(dir.isDirectory()){
            for(int i = 0; i < 20; i++){
                File file = new File(directory + File.separator + "file" + i +
                    System.currentTimeMillis() + ".txt");
                file.createNewFile();
            }
        }
        else{
            System.out.println("dir is not a dir");
        }
    }
}
```

## 二、日历和时间

### Instant

```
public static void main(String[] args) {
    // 获取当前时间戳
    Instant timestamp = Instant.now();
    long ts = timestamp.toEpochMilli();

    // ts 转为 Instant对象
    long ts2 = 1584700633000L;
    Instant instant = Instant.ofEpochMilli(ts2);
    // instant 可以转化为LocalDateTime 表示当地的时间
    LocalDateTime ldt3 = LocalDateTime.ofInstant(instant,
        ZoneId.systemDefault());
    System.out.println(ldt3);

    // LocalDateTime 可以进一步转化为ZonedDateTime, 加上了时区的信息
    ZonedDateTime zonedDateTime = ZonedDateTime.of(ldt3, ZoneId.systemDefault());
    // ZonedDateTime 又可以转换从Instant
    Instant l = zonedDateTime.toInstant();
    System.out.println(l);
    boolean equals = instant.equals(l);
    System.out.println(equals);

    Instant epoch = EPOCH;
    System.out.println(epoch);
    System.out.println(MIN); //-10000000000-01-01T00:00:00Z
    System.out.println(MAX); //+10000000000-12-31T23:59:59.999999999Z

    // 一小时以后
    Instant oneHourLater = timestamp.plus(1, ChronoUnit.HOURS);
    LocalDateTime ldt = LocalDateTime.ofInstant(oneHourLater,
        ZoneId.systemDefault());
    System.out.printf("%s %d %d at %d:%d%n", ldt.getMonth(), ldt.getDayOfMonth(),
        ldt.getYear(), ldt.getHour(), ldt.getMinute());

    // 两天前
    Instant twoDaysAgo = timestamp.minus(2, ChronoUnit.DAYS);
    LocalDateTime ldt2 = LocalDateTime.ofInstant(twoDaysAgo,
        ZoneId.systemDefault());
    System.out.printf("%s %d %d at %d:%d%n", ldt2.getMonth(),
        ldt2.getDayOfMonth(),
        ldt2.getYear(), ldt2.getHour(), ldt2.getMinute());
}
```