Grapes 常用操作工具

使用说明

（JSPGen4.0 工具包）

联系方式

QQ 群：122818143

在线 API 文档： http://help.jspgen.com/api/grapes4/

www.JSPGen.com

二零一四年三月

(共 25 页)

目 录

一、介绍

Grapes（葡萄）：是一个用于 Java 编程的工具包，为开发人员提供高效的、功能丰富的常用工具集合，可多次重复调用，以此来解决平时编程会经常遇到的问题，减少重复劳动，由 JSPGen 软件开发框架第 4.0 版时提出（前身为 JSPGen3.0 时的工具类 JSPGenUtils 包）。

在这个版本中，对原很多方法进行了重新规范式命名以及对性能进行优化处理，这里选一些比较常用的方法做简单介绍（更多方法及参数含义可参考 API 文档）：基础操作（含基础判断、基础获取、字符串处理)、日期时间操作、文件操作、图片操作、邮件发送、缓存操作、常用工具。

注：部分操作（grapes.pack包内的方法）需要第三方组件的支持，建议到官网下载第三方组件包。

以下内容中【绿色】为我们需要留意的关键内容(不含代码注释)，【红色】为我们可修改的内容，【蓝色】为提醒色或为不可更改的内容(不含代码注释)，【灰色】为某种功能效果的另一种实现方式。

1.1、Grapes 下载

JSPGen 官网：http://www.jspgen.com/

1.2、GrapesAPI

在线地址：http://help.jspgen.com/api/grapes4/

二、基础操作（Grapes）

基础操作，本节含基础判断、基础获取、字符串处理，文件地址：grapes. Grapes

2.1、基础判断

1、判断对象是否为空，返回 boolean 型：

方法：isEmpty(Object obj)　　例：Grapes.isEmpty(obj)

返回：true 为空、false 不为空

2、判断对象是否为空并返回默认值，如果为空返回默认值，如果不为空返回原值：

方法：isEmpty(Object obj , Object init)　　例：Grapes.isEmpty(obj ,init)

返回：如果为空返回 init，如果不为空返回 obj

3、判断对象是否不为空，返回 boolean 型：

方法：isNotEmpty(Object obj)　　例：Grapes.isNotEmpty(obj)

返回：true 不为空、false 为空

4、判断网络是否通畅，返回 boolean 型：

方法：isPing(String url)　　例：Grapes.isPing(“http://www.jspgen.com/”)

返回：true 通畅、false 不通

5、判断是否是移动设备，返回 boolean 型：

方法：isMobile(String userAgent)　　例：Grapes. isMobile(“***”)

返回：true 是移动设备、false 不是移动设备

2.2、基础获取

1、获取某类运行根目录全路径(绝对 URI 路径)：

方法：getResourcePath (Object obj)　例：Grapes. getResourcePath (this)

返回：A：.../WEB-INF/classes/　或 B：.../workplace/xxx/bin/ 类似路径

2、获取 Class 对象：

方法：getClazz(String name)　　例：Grapes. getClazz（“grapes.Grpes”）

返回：该路径类的 Class 型对象

3、获取 Class 对象实例：

方法：getBean (String name)　　例：Grapes. getBean（“grapes.Grpes”）

返回：该路径类的 Object 型实例

4、获取随机 UUID 字符串，参数为是否含中划线（默认为否）：

方法：uuid (boolean isflag)　　例：Grapes. uuid ()

返回：uuid 字符串

5、获取某字符串中的随机字符，第一个参数为原字符串，第二个参数为获取随机字符长度：

方法：random (String str，int len)　　例：Grapes. random (5)

返回：长度为 5 的字符串（在无原字符串的情况下，默认为 0-9 和 A-Z 字符串）

6、获取固定范围的随机整数，第一个参数为最小值，第二个参数为最大数：

方法：rand(int min, int max)　　例：Grapes. rand(0, 100)

返回：0-100 之间的随机整数（在无第一个参数的情况下，默认为 0）

7、获取字符串 Boolean 型值，参数可为字符型 1，0、true，false、on，off、yes，no：

方法：getBoolean(String str)　　例：Grapes. getBoolean(“on”)

返回：boolean 型 true

8、获取字符串长度,一个汉字按 2 个字节计算：

方法：length (String str)　　例：Grapes. length(“***”)

返回：0-100 之间的随机整数（在无第一个参数的情况下，默认为 0）

2.3、字符串处理

1、取消字符串两边空格，含纯控制符：

方法：trim(String str)　　例：Grapes. trim（“***”）

2、取消字符串前面的 0：

方法：unZero (String str)　　例：Grapes. unZero（“08”）

返回：字符串 8

3、将字符串内容进行反向转换，大写变小写、小写变大写：

方法：swapCase (String str)　　例：Grapes. swapCase（“JSPGen”）

返回：jspgEN

4、将字符串按间隔符拆分为 List<String>，间隔符默认为半角逗号，多参数调用请参考 API 文档：

方法：splitList (String str)　　例：Grapes. splitList（“JSP，Gen”）

返回：[JSP，Gen]

5、在原字符串开头添加字符串，如果原字符串开头存在要添加的字符串则输出原字符串：

方法：addStart (String str，String add)　　例：Grapes. addStart（“Gen”,”JSP”）

返回：JSPGen

6、在原字符串结尾添加字符串，如果原字符串结尾存在要添加的字符串则输出原字符串：

方法：addEnd (String str，String add)　　例：Grapes. addEnd（“JSP”,” Gen “）

返回：JSPGen

7、截取字符串并补充后缀字符串，多参数调用请参考 API 文档：

方法：substring(String str, int len, String postfix)

例：Grapes. substring（“JSPGen.com”，7, “...”）

返回：JSPGen...

三、日期时间操作（Dates）

日期时区有两种标准，其中 UTC 为高精度世界时间标准：

GMT：格林尼治平时(Greenwich Mean Time)英文缩写

UTC：协调世界时(Universal Time Coordinated)英文缩写，保持的以秒为基础的时间标度

UTC = GMT +/- 0.9s

在此我们推荐使用 UTC 标准来表示日期时间（GMT 时区获取仅作参考，以 UTC 为主）。

日期时间操作，本节含日期获取、单位匹配、格式转换、日期计算、日期判断，文件地址：

grapes. Dates

3.1、日期获取

a、获取 GMT 时区集合：

方法：getTimezoneGMT()　例：Dates. getTimezoneGMT()

返回：Map<String, String[]>型

b、获取当前默认时区 Id(GMT)：

方法：getDefaultTimezoneGMTId ()　例：Dates. getDefaultTimezoneGMTId ()

返回：GMT+08:00

1、获取 UTC 时区集合：

方法：getTimezoneIds()　例：Dates. getTimezoneIds()

返回： String[]型

2、获取当前默认时区 Id（UTC）：

方法：getDefaultTimezoneId () 例：Dates. getDefaultTimezoneId ()

返回：Asia/Shanghai

3、获取当前时间(日期型)：

方法：getTime() 例：Dates. getTime()

返回：Date型日期时间

4、获取当前时间（13 位时间戳）：

方法：getTimeMillis() 例：Dates. getTimeMillis()

返回：Long型时间（精确至毫秒）

5、获取当前时间（10 位时间戳）：

方法：getTimes() 例：Dates. getTimes()

返回：Long型时间（精确至秒）

6、获取 CST 日期时间的时间截：

方法：getCSTMillis(String dateTime)

说明： CST为中国标准时间格式：EEE MMM dd HH:mm:ss ZZZ yyyy

7、获取 RFC 日期时间的时间截：

方法：getRFCMillis(String dateTime)

说明：RSS2.0标准时间格式：E,dd MMM yyyy HH:mm:ss

8、获取某时间戳在某时区的日期时间：

方法：getDateTime(long time, String pattern, String timeZoneId)

说明：第一个参数为时间戳，第二个为日期表现格式（默认：yyyy-MM-dd HH:mm:ss），第三个为

时区id（默认为当前所在地时区）

3.2、单位匹配

1、获取两个日期间隔时间并按某一单位表示：

方法：getInterval(long fistTime, long secondTime, TimeUnit key)

说明：key默认：milli，可用值： milli(毫秒)、second（秒）、minute（分）、hour（时）、day（日）

2、获取日期时间单位(自动匹配)：

方法：getUnitTime (long interval, boolean isflag)

说明：interval 时间戳、isflag 显示中文单位(默认true)

3.3、格式转换

1、将日期型转字符型：

方法：parseString(Date date, String pattern, String timeZoneId)

说明：date 日期型日期、pattern 转换后日期格式(默认：yyyy-MM-dd HH:mm:ss)、timeZoneId 时区Id(默认为当前时区)

2、将 long 型转字符型：

方法：parseString(long time, String pattern, String timeZoneId)

说明：time 时间戳、pattern 转换后日期格式(默认：yyyy-MM-dd HH:mm:ss)、timeZoneId 时区Id(默认为当前时区)

3、将字符型转日期型：

方法：parseDate(String dateTime, String[] pattern, String timeZoneId)

说明：dateTime 字符型日期、pattern 转换前日期格式集合、timeZoneId 时区Id(默认为当前时

区)

4、将字符型转日期型：

方法：parseDate(String dateTime, String pattern, String timeZoneId)

说明：dateTime 字符型日期、pattern 转换前日期格式、timeZoneId 时区Id(默认为当前时区)

5、将 long 型转日期型：

方法：parseDate(long time)　　说明：time 时间戳

6、将字符型转 long 型：

方法：parseLong(String dateTime, String pattern, String timeZoneId)

说明：dateTime 字符型日期、pattern 转换前日期格式、timeZoneId 时区Id(默认为当前时区)

7、将日期型转 long 型：

方法：parseLong(Date date)　　说明：date 日期型日期

3.4、日期计算

1、日期加减(由 field 参数指定计算成员)：

方法：compute(String dateTime, String pattern, int field, int amount)

例：如果dateTime为 2000年8月20日，那么：

compute (dateTime,Calendar.YEAR,-10)：值为1990年8月20日

compute (dateTime,Calendar.YEAR,+10)：值为2010年8月20日

说明：

dateTime 日期处理前格式：yyyy-MM-dd HH:mm:ss

pattern 目标格式：yyyy-MM-dd HH:mm:ss

field 日期成员，主要有:

年:Calendar.YEAR　　月:Calendar.MONTH

周:Calendar.WEEK_OF_YEAR　　日:Calendar.DATE

时:Calendar.HOUR　　分:Calendar.MINUTE

秒:Calendar.SECOND　　毫秒:Calendar.MILLISECOND

amount 加减幅度(+n=加n个由参数field指定的日期成员值; -n=减n个由参数field代表的日期成员值)

3.5、日期判断

1、日期比较(检查 dateTimeA 是否在 dateTimeB 之前)：

方法：isBefore(String dateTimeA, String dateTimeB, boolean isflag)

说明：isflag是否容错处理(默认为是)，如 dateA 与 dateB 相同，则返回true

2、日期比较(检查 dateTimeA 是否在 dateTimeB 之后)：

方法：isAfter (String dateTimeA, String dateTimeB, boolean isflag)

说明：isflag是否容错处理(默认为是)，如 dateA 与 dateB 相同，则返回true

3、日期判断，检查某日期是否在某两个日期之间（日期型日期）：

方法：between(Date startDate, Date endDate, Date thisDate)

说明：返回整型，0:格式出错 1:进行中 2:已结束 3:未开始

4、日期判断，检查某日期是否在某两个日期之间（字符型日期）：

方法：between(String startTime, String endTime, String thisTime)

说明：返回整型，0:格式出错 1:进行中 2:已结束 3:未开始

四、文件操作（Files）

文件操作，本节含文件操作（创建、重命名、移动、复制、删除、内容读取写入）、属性获取、文件解压缩，文件地址：grapes. Files

4.1、文件操作

1、路径格式化：

方法：format(String filepath)　　例：Files. format(“D:///\\test\\\\\\\\\\\//\\\\text.txt”)

返回：D:/test/text.txt

2、创建目录(自动创建多级子目录)：

方法：createFolder (String filepath)　　例：Files. createFolder (“***”)

返回：创建后的目录 File 对象，若为 null 则说明创建失败

3、创建文件(自动创建多级父目录)：

方法：createFile(String filepath)　　例：Files. createFile(“***”)

返回：创建后的文件 File 对象，若为 null 则说明创建失败

4、创建目录并返回是否创建成功：

方法：isCreateFolder (String filepath)　　例：Files. isCreateFolder (“***”)

返回：创建成功为 true，失败为 false

5、创建文件并返回是否创建成功：

方法：isCreateFile (String filepath)　　例：Files. isCreateFile (“***”)

返回：创建成功为 true，失败为 false

6、文件重命名：

方法：rename(String filepath, String newname)　　返回：命名成功为 true，失败为 false

说明：第一个参数为要重命名的文件路径地址，第二个参数为文件新名称，无路径

7、移动文件(文件、目录自动识别)：

方法：move(String filepath, String newfilepath)　返回：移动成功为 true，失败为 false

8、复制文件(文件、目录自动识别)：

方法：copy (String filepath, String newfilepath)　返回：复制成功为 true，失败为 false

9、删除文件(含目录)：

方法：delete (String filepath)　返回：复制成功为 true，失败为 false

10、清空目录里所有子文件(不删除此目录)：

方法：delete (String filepath)　返回：复制成功为 true，失败为 false

11、删除目录里符合时间截操作类型的子文件(含当前目录)：

方法：delete(String filepath, long time, String genre)

说明：第一个参数为目标路径，第二个参数为时间戳，第三个参数为操作类型，默认 all(基于时间截，之前：before、之后：after、当前：current、全部：all)

12、删除目录里所有符合时间截的子文件(含当前目录)：

方法：delete(String filepath, long startTime, long endTime)

说明：第一个参数为目标路径，第二个参数为起始时间戳，第三个参数为结束时间戳

13、写入内容：

方法：writeFile(String filepath, String content, String charset, boolean isflag)

说明：第一个参数为文件路径（可为文件对象），第二个参数为写入内容，第三个参数为写入编码（默认 UTF-8），第四个参数为是否追加写入(默认否)。

14、读取内容：

方法：readFile(String filepath, String charset)

说明：第一个参数为目标路径，第二个参数为读取编码（默认 UTF-8）

15、读取固定长度内容：

方法：readFile(String filepath, String charset, int len)

说明：第一个参数为目标路径，第二个参数为读取编码（默认 UTF-8），第三个参数为读取长度，如：1024、2048 等。

16、读取内容按行存到 List：

方法：readFileList(String filepath, String charset)　　返回：List<String>

说明：第一个参数为目标路径，第二个参数为读取编码（默认 UTF-8）

4.2、属性获取

1、获取单个文件属性：

方法：getAttr(String filepath)　　返回：Map<String, Object>对象，其 key 为：

isExists:是否存在、isRead:是否可读、isWrite:是否可写、isDirectory:是否为文件夹

isFile:是否为文件、isHidden:是否隐藏、isAbsolute:是否绝对路径、name:文件名称

exte:扩展名、type:文件类型、path:文件路径、absolutePath:文件绝对路径

parent:上一级目录的名称、lastTime:最后修改时间、list:子目录列表

size:文件大小(有单位)、 length:文件大小(无单位)

2、获取某目录中的所有子文件信息（含子文件属性）：

方法： getFileList(String filepath, String extes, long time, String genre, String order, int startNum, int endNum)

返回：List<Map<String, Object>>对象

说明：filepath 目录路径；time 时间戳，为 0 时获取当前时间；

extes 组成成员：

*：列出所有文件名(含目录) 、x 扩展名集合：列出扩展名为 x 中的文件(以.为间隔，dir:目录扩展名)；genre 操作类型，默认 all(基于时间截，之前：before、之后：after、当前：current、全部：all)；order 排序方式 默认 :time desc (name:以文件名排序,目录排在前面 size:大小 ,目录排在前面 time:最后修改时间、asc 升序 desc 降序)；startNum 列表开始位置，默认 0；endNum 列表结束位置，默认 20。

3、获取某目录中的所有文件名：

方法：getList(String filepath) 返回：String[]数组

4、获取某目录、文件大小：

方法：getFileSize(String filepath) 返回：long 型

5、文件单位转换(无单位到有单位)：

方法：getFileSizeUnit(long length) 返回：String 字符串

6、文件单位转换(有单位到无单位)：

方法：getFileSizeByte(String length) 返回：long 型

7、获取文件类型：

方法：getFileType (String exte) 返回：传递文件扩展名返回文件类型字符串

4.3、文件解压缩

1、文件压缩：

方法：zip(String filepath, String zipfilepath)、zip(String filepath)

返回：boolean 型，成功 true，失败 false

2、文件解压：

方法：unzip(String zipfilepath, String filepath)、unzip(String zipfilepath)

返回：boolean 型，成功 true，失败 false

五、图片操作（Images）

图片操作，支持图片格式：GIF、JPEG、PNG、BMP；

本节含图片判断、水印添加、图片缩放、图片裁切、EXIF 信息获取，文件地址：

grapes.image. Images 与 grapes.image. EXIFImages

5.1、图片判断

1、判断图片是否为同一张：

方法：isSame(Object obj1, Object obj2)　　返回：boolean 型

说明：参数 obj 可为文件路径字符串，也可为文件 File 对象

2、判断图片是否能正常显示：

方法：isShow(Object obj)　　返回：boolean 型

说明：参数 obj 可为文件路径字符串，也可为文件 File 对象

5.2、水印添加

操作前需初始化 ImagesData 属性对象，详细元素参见 API。

方法：watermark(String srcFile, String background, int position, int posX, int posY, int alpha, boolean isflag)

说明：srcFile 文件地址字符串，background 空白背景色(无#号十六色值:FFFFFF)，

position 定位方式(-1-10，-1 为自定义、0 为随机、10 为平铺)，alpha 透明度，isflag 文字水印

5.3、图片缩放

方法：zoom(String srcFile, String background, int destWidth, int destHeight, int degree, boolean isamp, boolean isratio, ImagesDatum datum)

说明：srcFile 文件地址字符串，background 空白背景色(无#号十六色值:FFFFFF)，destWidth 缩放宽度，destHeight 缩放高度，degree 旋转角度，isamp 是否放大处理(小于指定宽高则进行放大)，isratio 是否成正比缩放，datum 成正比缩放基数(auto:自动, width、height:以宽度、高度为准成正比缩放)

5.4、图片裁切

方法：crop(String srcFile, String background, int cutX, int cutY, int cropWidth, int cropHeight)

说明：srcFile 文件地址字符串，background 空白背景色(无#号十六色值:FFFFFF)，裁切起始位置，裁切宽度，裁切高度

5.5、EXIF 信息获取

仅 JPG 格式文件有此信息，使用时需 metadata-extractor2.6.2.jar、xmpcore.jar 组件包支持，文件：grapes.image. EXIFImages。

方法：getInfo(Object obj)　　返回：Map<String, String>集合

说明：参数 obj 可为文件路径字符串，也可为文件 File 对象，返回 key 元素说明：

相片信息【宽度(像素)：width 高度(像素)：height】

设备信息【品牌：make、型号：model、标题：winTitle、作者：winAuthor、

X 轴分辨率(像素)：xResolution、Y 轴分辨率(像素)：yResolution】

拍摄信息【焦距长度：focalLength、快门速度：exposureTime、光圈大小：fNumber、

闪光灯：flash、曝光补偿：exposurBias、曝光程序：exposureMode、

ISO 感光度：isoEquivalent、拍摄日期：dateTimeOriginal】

六、邮件发送（Mail）

邮件发送，本节含认证发送、MX 发送，文件地址：grapes.mail. Mail

6.1、认证发送

```
// 初始化，内容格式(text、html、url)
Mail mail = new Mail("text");
// 认证信息，发送邮件服务器(如:smtp.163.com)、服务器端口(默认25)、用户名、密码
mail.setSMTP("smtp.163.com", 25，" test100", " test100");
```

注：认证用户名有些服务器需要的是整个发送邮件地址。

```
// 发送邮件，更多方法请参考API
// 发送者地址、发送者昵称、接收地址(以;间隔)、地址类型(普通：to、抄送：cc、密送：bcc)、邮件主题、邮件内容、发送日期（为空则为当前时间）
if(mail.sender("test100@qq.com", "test100", "test200@qq.com", "to", "SMTP测试", "你好Mail", null, false)){
    System.out.println("SMTPMail OK");
}else{
    System.out.println("SMTPMail Error");
}
```

6.2、MX 发送

```
// 初始化，内容格式(text、html、url)
Mail mail = new Mail("text");
// 发送者地址、发送者昵称、接收地址(以;间隔)、地址类型(普通：to、抄送：cc、密送：bcc)、邮件主题、邮件内容、发送日期（为空则为当前时间）
if(mail.sender("test100@qq.com", "test100", "test200@qq.com", "to", "SMTP测试", "你好Mail", null)){
    System.out.println("MXMail OK");
}else{
    System.out.println("MXMail Error");
}
```

七、缓存操作（Cache）

缓存操作，本节含缓存配置、缓存演示；

缓存配置文件地址： cache-config.xml(类文件根目录下)、缓存操作文件地址：

grapes. cache.Cache

7.1、缓存配置

```
<?xml version="1.0" encoding="UTF-8"?>
<cache version="4.0">
    <!-- 缓存状态(是否启用) -->
    <status>false</status>
    <!-- 缓存类型：M，内存、HD，硬盘、MHD，先内存再硬盘(默认M) -->
    <genre></genre>
    <!-- 缓存文件保存路径(默认项目根目录下) -->
    <path></path>
    <!-- 缓存文件扩展名 -->
    <exte></exte>
    <!-- 内存剩余极限(默认10M)，内存小于此值则清除缓存或转存至硬盘(由缓存类型来区分) -->
    <free></free>
    <!-- 时间周期(<=0时不限制，单位豪秒，默认0 ) -->
    <time></time>
    <!-- 访问累积次数(<=0时不限制，默认10次) -->
    <count></count>
</cache>
```

7.2、缓存演示

```
public static void main(String[] argc){
        Map<String, Object> m0 = new HashMap<String, Object>();
        m0.put("name", "张三");
        m0.put("age", 20);
        Map<String, Object> m1 = new HashMap<String, Object>();
        m1.put("name", "李四");
        m1.put("age", 21);
        Map<String, Object> m2 = new HashMap<String, Object>();
        m2.put("name", "王麻子");
        m2.put("age", 22);
```

```
        List<Map<String, Object>> temp = new ArrayList<Map<String, Object>>();
        temp.add(m0);
        temp.add(m1);
        temp.add(m2);

        System.out.println("------ 元数据 ---------");
        for(int i=0; i<temp.size(); i++){
            Map<String, Object> map = (Map<String, Object>)temp.get(i);
            System.out.println("用户名: "+map.get("name")+" 年龄:
"+map.get("age"));
        }

        System.out.println("------ 演示开始 ---------");

        for(int j=0; j<10; j++){
            System.out.println("序列: "+j);
            System.out.println("-----------------------");

            Cache c = Cache.getInstance();      // 初始化一个缓存对象
            String key = c.getKey("cache", "main", new Object[]{"参数1","参数2"});
    // 组装缓存key
            System.out.println("缓存 KEY: "+key);

            List<Map<String, Object>> test = new ArrayList<Map<String,
Object>>();
            test = (List<Map<String, Object>>) c.getData(key); // 获取缓存数据
            if (Grapes.isNotEmpty(test)) {
                System.out.println("== 获取缓存 ==");
                System.out.println("缓存数据: ");
                for(int i=0; i<test.size(); i++){
                    Map<String, Object> map = (Map<String, Object>)test.get(i);
                    System.out.println("用户名: "+map.get("name")+" 年龄:
"+map.get("age"));
                }
                System.out.println("");
            }else{
                test = temp;
                System.out.println("【添加缓存】");
                c.addData(key, test); // 添加缓存数据

                System.out.println("原始数据: ");
                for(int i=0; i<test.size(); i++){
                    Map<String, Object> map = (Map<String, Object>)test.get(i);
                    System.out.println("用户名: "+map.get("name")+" 年龄:
```

```
"+map.get("age"));
                }
                System.out.println("");
            }

            if(j==5){
                System.out.println("== 更新缓存 ==");
                System.out.println("更新前"+c.toString());
                c.remove("cache", "main"); // 移除缓存数据(批量)
                System.out.println("更新后"+c.toString());
            }
            System.out.println("------ Path ---------"+c.getKeyFile(key)); // 获
取缓存文件地址
        }

        System.out.println("------ 演示结束 ---------");
    }
}
```

八、常用工具（Tools）

常用信息，本节含 List 对象排序、Map 对象排序、客户端信息获取、服务端信息获取、数据大写操作、身份证号码判断；

插件目录：grapes. tools.*

8.1、List 对象排序

支持对象多属性排序、升降序排序及中文排序，文件地址：grapes. tools.OrderList

List 属性支持 Bean 对象、Map 对象，这里以 Map 对象演示，示例：

```
Map<String, Object> m1 = new HashMap<String, Object>();
m1.put("name", "张三");      m1.put("age", 20);
m1.put("salary", 2000);      m1.put("birthday", "1999-02-10");

Map<String, Object> m2 = new HashMap<String, Object>();
m2.put("name", "李四");      m2.put("age", 22);
m2.put("salary", 1800);      m2.put("birthday", "1998-02-10");

List<Map<String, Object>> temp = new ArrayList<Map<String, Object>>();
```

```
temp.add(m1);                temp.add(m2);     //…
System.out.println("按照生日从小到大排序，生日相同再按年龄从小到大排序(Map型) ");
OrderList<Map<String, Object>> olm = new OrderList<Map<String, Object>>();
olm.sort(temp, new String[]{"birthday", "age"}, "asc");
for(int i=0; i<temp.size(); i++){
    Map<String, Object> person = (Map<String, Object>)temp.get(i);
    System.out.println("用户名："+person.get("name"));
    System.out.println(" 年龄："+person.get("age"));
    System.out.println(" 月薪："+person.get("salary"));
    System.out.println(" 生日："+person.get("birthday"));
}
```

8.2、Map 对象排序

支持 key、value 属性排序、升降序排序及中文排序，文件地址：grapes. tools.OrderMap

示例：

```
Map<String, Object> maps = new HashMap<String, Object>();
maps.put("boy", 8);          maps.put("cat", 7);
maps.put("dog", 1);          maps.put("apple", 5);
// 排序前的输出
Set set = maps.entrySet();
Iterator i = set.iterator();
while (i.hasNext()) {
    Map.Entry<String, Object> entry = (Map.Entry<String, Object>) i.next();
    System.out.println(entry.getKey() + "--->" + entry.getValue());
}
System.out.println("----------------");
// 按key排序后的输出
Map<String, Object> km = (Map<String, Object>) OrderMap.sort(maps, "key", "asc");
Set keySet = km.entrySet();
Iterator ki = keySet.iterator();
while (ki.hasNext()) {
    Map.Entry<String, Object> entry = (Map.Entry<String, Object>) ki.next();
    System.out.println(entry.getKey() + "--->" + entry.getValue());
}
```

8.3、获取客户端信息

A、PC 端信息【文件地址：grapes.tools.Browser】

1、获取操作系统名称

方法：getOSName(String userAgent)

说明：参数为浏览器头信息，由 request.getHeader("User-Agent")获得

2、获取浏览器版本

方法：getBrowserVer (String userAgent)　　说明：参数为浏览器头信息

3、获取客户端真实 IP

方法：getIP(HttpServletRequest request)　　说明：参数为浏览器请求对象

4、隐藏 IP 结尾片段

方法：hide(String ip, int level)

说明：第一个参数为 IP 地址，第二个参数为隐藏级别，如：

hide("127.0.0.1", 1) 返回：127.0.0.*　　hide("127.0.0.1", 2) 返回：127.0.*.*

B、移动端信息【文件地址：grapes.tools.Mobile】

1、判断是否移动设备

方法：isMobile(String userAgent)　　说明：参数为浏览器头信息

2、获取移动设备名称

方法：getMobileName(String userAgent)　　说明：参数为浏览器头信息

8.4、获取服务端信息

文件地址：grapes.tools. Server

A、获取系统属性

方法：getSystemProperty(String property)

说明：参数可为如下字符

【os.arch：系统型号 32 或 86、os.name：系统名称、os.version：系统版本...】

B、获取服务端的 Mac 地址(Windows、Linux)

方法：getMac()

C、获取某域名 IP

方法：getIP(String domain)

D、获取服务端 IP

方法：getServerIP()

8.5、获取数字大写（数额）

文件地址：grapes.tools. Amount

方法：getAmount (double tempNumber)、getAmount (BigDecimal tempNumber)

8.6、身份证号码判断

文件地址：grapes.tools.IDCard

示例：

```
String idCard = "652829198009185813x";
System.out.println("合法："+ IDCard.isIDCard(idCard));
System.out.println("年龄："+ IDCard.getAges(idCard));
System.out.println("生日："+ IDCard.getBirthDay(idCard));
System.out.println("性别："+ IDCard.getGender(idCard));
System.out.println("省份："+ IDCard.getProvince(idCard));
```