链滴

工具类之——StringUtils

作者：zml2015
原文链接：https://ld246.com/article/1484277574485
来源网站：链滴

有时候根据实际需求，要将数据进行一定的格式转换，在这里提供一个实际开发中使用的一个工类，工具类中方法会持续更新
<pre class="brush: java">package top.wys.developerclub.utils;
/**
 * @author 郑明亮
 * @Time：2016年12月22日 上午11:04:44
 * @Description 字符串相关工具类
 */
public class StringUtils {
<pre><code class="highlight-chroma">private StringUtils() {
    throw new Uns pportedOperationException("Can not be instantiated...");
}

/**
* 判断字符串是否null或长度为0
*
* @param s 待校字符串
* @return {@cod true}: 空
 {@code false}: 不为空
*/
public static bool an isEmpty(CharSequence s) {
    return s == null | s.length() == 0;
}

/**
* 判断字符串是否null或全为空格
*
* @param s 待校字符串
* @return {@cod true}: null或全空格
 {@code false}: 不为null且不全空格
*/
public static bool an isSpace(String s) {
    return (s == nul || s.trim().length() == 0);
}

/**
* 判断两字符串是相等
*
* @param a 待校字符串a
* @param b 待验字符串b
* @return {@cod

```
 true}: 相等&amp;lt;br&amp;gt;{@code false}: 不相等
</span></span><span class="highlight-line"><span class="highlight-cl"> */
</span></span><span class="highlight-line"><span class="highlight-cl">public static bool
an equals(CharSequence a, CharSequence b) {
</span></span><span class="highlight-line"><span class="highlight-cl">    if (a == b) retur
 true;
</span></span><span class="highlight-line"><span class="highlight-cl">    int length;
</span></span><span class="highlight-line"><span class="highlight-cl">    if (a != null &a
p;amp;&amp;amp; b != null &amp;amp;&amp;amp; (length = a.length()) == b.length()) {
</span></span><span class="highlight-line"><span class="highlight-cl">        if (a instance
f String &amp;amp;&amp;amp; b instanceof String) {
</span></span><span class="highlight-line"><span class="highlight-cl">            return a.e
uals(b);
</span></span><span class="highlight-line"><span class="highlight-cl">        } else {
</span></span><span class="highlight-line"><span class="highlight-cl">            for (int i =
; i &amp;lt; length; i++) {
</span></span><span class="highlight-line"><span class="highlight-cl">                if (a.cha
At(i) != b.charAt(i)) return false;
</span></span><span class="highlight-line"><span class="highlight-cl">            }
</span></span><span class="highlight-line"><span class="highlight-cl">            return true

</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    return false;
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">/**
</span></span><span class="highlight-line"><span class="highlight-cl">* 判断两字符串忽
大小写是否相等
</span></span><span class="highlight-line"><span class="highlight-cl">*
</span></span><span class="highlight-line"><span class="highlight-cl">* @param a 待校
字符串a
</span></span><span class="highlight-line"><span class="highlight-cl">* @param b 待
验字符串b
</span></span><span class="highlight-line"><span class="highlight-cl">* @return {@cod
 true}: 相等&amp;lt;br&amp;gt;{@code false}: 不相等
</span></span><span class="highlight-line"><span class="highlight-cl">*/
</span></span><span class="highlight-line"><span class="highlight-cl">public static bool
an equalsIgnoreCase(String a, String b) {
</span></span><span class="highlight-line"><span class="highlight-cl">    return (a == b) |
 (b != null) &amp;amp;&amp;amp; (a.length() == b.length()) &amp;amp;&amp;amp; a.regio
Matches(true, 0, b, 0, b.length());
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">/**
</span></span><span class="highlight-line"><span class="highlight-cl">* null转为长度为
的字符串
</span></span><span class="highlight-line"><span class="highlight-cl">*
</span></span><span class="highlight-line"><span class="highlight-cl">* @param s 待转
符串
</span></span><span class="highlight-line"><span class="highlight-cl">* @return s为nul
转为长度为0字符串，否则不改变
</span></span><span class="highlight-line"><span class="highlight-cl">*/
</span></span><span class="highlight-line"><span class="highlight-cl">public static String
```

```
null2Length0(String s) {
</span></span><span class="highlight-line"><span class="highlight-cl">    return s == null
? "" : s;
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">/**
</span></span><span class="highlight-line"><span class="highlight-cl">  * 返回字符串长度
</span></span><span class="highlight-line"><span class="highlight-cl">  *
</span></span><span class="highlight-line"><span class="highlight-cl">  * @param s 字符

</span></span><span class="highlight-line"><span class="highlight-cl">  * @return null返
0，其他返回自身长度
</span></span><span class="highlight-line"><span class="highlight-cl">  */
</span></span><span class="highlight-line"><span class="highlight-cl">public static int le
gth(CharSequence s) {
</span></span><span class="highlight-line"><span class="highlight-cl">    return s == null
? 0 : s.length();
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">/**
</span></span><span class="highlight-line"><span class="highlight-cl">  * 首字母大写
</span></span><span class="highlight-line"><span class="highlight-cl">  *
</span></span><span class="highlight-line"><span class="highlight-cl">  * @param s 待转
符串
</span></span><span class="highlight-line"><span class="highlight-cl">  * @return 首字
大写字符串
</span></span><span class="highlight-line"><span class="highlight-cl">  */
</span></span><span class="highlight-line"><span class="highlight-cl">public static String
upperFirstLetter(String s) {
</span></span><span class="highlight-line"><span class="highlight-cl">    if (isEmpty(s) ||
Character.isLowerCase(s.charAt(0))) return s;
</span></span><span class="highlight-line"><span class="highlight-cl">    return String.va
ueOf((char) (s.charAt(0) - 32)) + s.substring(1);
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">/**
</span></span><span class="highlight-line"><span class="highlight-cl">  * 首字母小写
</span></span><span class="highlight-line"><span class="highlight-cl">  *
</span></span><span class="highlight-line"><span class="highlight-cl">  * @param s 待转
符串
</span></span><span class="highlight-line"><span class="highlight-cl">  * @return 首字
小写字符串
</span></span><span class="highlight-line"><span class="highlight-cl">  */
</span></span><span class="highlight-line"><span class="highlight-cl">public static String
lowerFirstLetter(String s) {
</span></span><span class="highlight-line"><span class="highlight-cl">    if (isEmpty(s) ||
Character.isUpperCase(s.charAt(0))) return s;
</span></span><span class="highlight-line"><span class="highlight-cl">    return String.va
ueOf((char) (s.charAt(0) + 32)) + s.substring(1);
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">/**
</span></span><span class="highlight-line"><span class="highlight-cl">  * 反转字符串
</span></span><span class="highlight-line"><span class="highlight-cl">  *
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> * @param s 待反
字符串
</span></span><span class="highlight-line"><span class="highlight-cl"> * @return 反转
符串
</span></span><span class="highlight-line"><span class="highlight-cl"> */
</span></span><span class="highlight-line"><span class="highlight-cl">public static String
reverse(String s) {
</span></span><span class="highlight-line"><span class="highlight-cl">    int len = length
s);
</span></span><span class="highlight-line"><span class="highlight-cl">    if (len &amp;amp;lt;
 1) return s;
</span></span><span class="highlight-line"><span class="highlight-cl">    int mid = len &
mp;gt;&amp;amp;gt; 1;
</span></span><span class="highlight-line"><span class="highlight-cl">    char[] chars = s.
oCharArray();
</span></span><span class="highlight-line"><span class="highlight-cl">    char c;
</span></span><span class="highlight-line"><span class="highlight-cl">    for (int i = 0; i
amp;lt; mid; ++i) {
</span></span><span class="highlight-line"><span class="highlight-cl">        c = chars[i];
</span></span><span class="highlight-line"><span class="highlight-cl">        chars[i] = cha
s[len - i - 1];
</span></span><span class="highlight-line"><span class="highlight-cl">        chars[len - i -
1] = c;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    return new Stri
g(chars);
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">/**
</span></span><span class="highlight-line"><span class="highlight-cl">  * 转化为半角字符
</span></span><span class="highlight-line"><span class="highlight-cl">  *
</span></span><span class="highlight-line"><span class="highlight-cl">  * @param s 待转
符串
</span></span><span class="highlight-line"><span class="highlight-cl">  * @return 半角
符串
</span></span><span class="highlight-line"><span class="highlight-cl">  */
</span></span><span class="highlight-line"><span class="highlight-cl">public static String
toDBC(String s) {
</span></span><span class="highlight-line"><span class="highlight-cl">    if (isEmpty(s)) re
urn s;
</span></span><span class="highlight-line"><span class="highlight-cl">    char[] chars = s.
oCharArray();
</span></span><span class="highlight-line"><span class="highlight-cl">    for (int i = 0, len
= chars.length; i &amp;amp;lt; len; i++) {
</span></span><span class="highlight-line"><span class="highlight-cl">        if (chars[i] ==
12288) {
</span></span><span class="highlight-line"><span class="highlight-cl">            chars[i] = '
;
</span></span><span class="highlight-line"><span class="highlight-cl">        } else if (652
1 &amp;amp;lt;= chars[i] &amp;amp;amp;&amp;amp;amp; chars[i] &amp;amp;lt;= 65374) {
</span></span><span class="highlight-line"><span class="highlight-cl">            chars[i] = (
har) (chars[i] - 65248);
</span></span><span class="highlight-line"><span class="highlight-cl">        } else {
</span></span><span class="highlight-line"><span class="highlight-cl">            chars[i] =
```

```
hars[i];
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    return new Stri
g(chars);
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">/**
</span></span><span class="highlight-line"><span class="highlight-cl">* 转化为全角字符
</span></span><span class="highlight-line"><span class="highlight-cl">*
</span></span><span class="highlight-line"><span class="highlight-cl">* @param s 待转
符串
</span></span><span class="highlight-line"><span class="highlight-cl">* @return 全角
符串
</span></span><span class="highlight-line"><span class="highlight-cl">*/
</span></span><span class="highlight-line"><span class="highlight-cl">public static String
toSBC(String s) {
</span></span><span class="highlight-line"><span class="highlight-cl">    if (isEmpty(s)) re
urn s;
</span></span><span class="highlight-line"><span class="highlight-cl">    char[] chars = s.
oCharArray();
</span></span><span class="highlight-line"><span class="highlight-cl">    for (int i = 0, len
= chars.length; i &amp;lt; len; i++) {
</span></span><span class="highlight-line"><span class="highlight-cl">        if (chars[i] ==
' ') {
</span></span><span class="highlight-line"><span class="highlight-cl">            chars[i] = (
har) 12288;
</span></span><span class="highlight-line"><span class="highlight-cl">        } else if (33 &
mp;lt;= chars[i] &amp;amp;&amp;amp; chars[i] &amp;lt;= 126) {
</span></span><span class="highlight-line"><span class="highlight-cl">            chars[i] = (
har) (chars[i] + 65248);
</span></span><span class="highlight-line"><span class="highlight-cl">        } else {
</span></span><span class="highlight-line"><span class="highlight-cl">            chars[i] =
hars[i];
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    return new Stri
g(chars);
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">/**
</span></span><span class="highlight-line"><span class="highlight-cl">* @author 郑明亮
</span></span><span class="highlight-line"><span class="highlight-cl">* @Time 2017年
月12日 下午1:16:39
</span></span><span class="highlight-line"><span class="highlight-cl">* @Description &
mp;lt;p&amp;gt;去掉浮点类型的小数点，并进行四舍五入  &amp;lt;/p&amp;gt;
</span></span><span class="highlight-line"><span class="highlight-cl">*              &amp;
t;p&amp;gt;应用场景：将浮点型转换为整型数据，并保留1位小数&amp;lt;/p&amp;gt;
</span></span><span class="highlight-line"><span class="highlight-cl">* @param amoun
String 浮点型金额字符串
</span></span><span class="highlight-line"><span class="highlight-cl">* @return 整型
字符串
</span></span><span class="highlight-line"><span class="highlight-cl">*/
</span></span><span class="highlight-line"><span class="highlight-cl">public static String
```

```
removeAmountPoint(String amountString){
</span></span><span class="highlight-line"><span class="highlight-cl">    String noPointS
ring = "";
</span></span><span class="highlight-line"><span class="highlight-cl">    if (amountStrin
 == null) {
</span></span><span class="highlight-line"><span class="highlight-cl">        return noPoi
tString;
</span></span><span class="highlight-line"><span class="highlight-cl">    }else{
</span></span><span class="highlight-line"><span class="highlight-cl">        Integer amo
nt;
</span></span><span class="highlight-line"><span class="highlight-cl">        if (amountStr
ng.indexOf("." )!= -1) {
</span></span><span class="highlight-line"><span class="highlight-cl">            String poin
Num =  amountString.substring(amountString.indexOf("." ));
</span></span><span class="highlight-line"><span class="highlight-cl">            amount  =
nteger.parseInt(amountString.replace(pointNum, ""));
</span></span><span class="highlight-line"><span class="highlight-cl">            if (pointN
m.length()&amp;gt;2) {
</span></span><span class="highlight-line"><span class="highlight-cl">                int num
= Integer.parseInt(pointNum.substring(1, 2));
</span></span><span class="highlight-line"><span class="highlight-cl">                if (num
&amp;gt;=5) {
</span></span><span class="highlight-line"><span class="highlight-cl">                    amou
t ++;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">                }
</span></span><span class="highlight-line"><span class="highlight-cl">            }
</span></span><span class="highlight-line"><span class="highlight-cl">            noPointStr
ng = ""+amount;
</span></span><span class="highlight-line"><span class="highlight-cl">        }else {
</span></span><span class="highlight-line"><span class="highlight-cl">            return am
untString;
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">        return noPoi
tString;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>/**<br>
* @<a href="https://ld246.com/member/author" aria-name="author" class="tooltipped__use
" target="_blank">author</a> 郑明亮<br>
* @<a href="https://ld246.com/member/Time" aria-name="Time" class="tooltipped__user" t
rget="_blank">Time</a> 2017 年 1 月 13 日 下午 4:18:14<br>
@param amount 金额<br>
* @param pattern 将数字转换为指定格式  {@value AmountPattern}<br>
* &lt;p&gt;如 amount:  10000000 ，当传入金额为 null 时，默认为 0 &lt;/p&gt;<br>
* &lt;p&gt;如 pattern:  \u00A5,###.00 ，当 pattern 为 null 时，使用默认 pattern ,即"\u00A5,#
0.00" &lt;/p&gt;<br>
* &lt;p&gt;如 return:  ¥10,000,000  &lt;/p&gt;<br>
* @return 格式化后的数字字符串<br>
* @throws ParseException<br>
*/<br>
public static String formateAmount(Object amount,String pattern) throws ParseException{<b
```

```
>
if (amount == null) {//当传入金额为 null 时，默认为 0<br>
return formateAmount(0,pattern);<br>
}<br>
if (pattern == null) {//当 pattern 为 null 时，使用默认 pattern<br>
pattern = "\u00A5,##0.00";<br>
}<br>
if (amount instanceof String) {<br>
return formateAmount(Double.parseDouble(amount.toString()),pattern);<br>
}<br>
DecimalFormat format = new DecimalFormat(pattern);<br>
String number = format.format(amount);<br>
return number;<br>
}<br>
}</p></pre><p></p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>
```