第二十章 API讲解

字符串

字符串

- API之
  - String
  - StringBuffer
  - StringBuilder

String类

- 方法

1 concat()
2 length()
3 equals()
4 equalsIgnoreCase()
5 toUpperCase()
6 toLowerCase()
7 indexOf()
8 lastIndexOf()
9 charAt()
10 substring(start)
11 substring(start,end)
12 trim()
13 replace(old,new)
14 startsWith()
15 endsWith()
16 compareTo()

StringBuffer类

- 方法

1. capacity()
2. trimToSize()
3. append()
4. insert();
5. setCharAt()
6. deleteCharAt();
7. delete();
8. reverse()
9. charAt()
10. indexOf()
11. lastIndexOf()

正则

正则表达式符号

符号	描述
\D	除了数字之外的任何字符，等价于[^0-9]
\d	匹配一个数字字符，等价于[0-9]
\W	任何非单词字符，等价于[^A-Za-z0-9_]
\w	任何单词字符，等价于[A-Za-z0-9_]
.	除了换行符之外的任意字符

符号	描述
{n}	匹配前一项n次
{n,m}	匹配前一项至少n次，但是不能超过m次
*	匹配前一项0次或多次，等价于{0，}
+	匹配前一项1次或多次，等价于{1，}
?	匹配前一项0次或1次，也就是说前一项是可选的，等价于{0,1}

正则表达式(Pattern、Matcher)

- 正则表达式是一个描述字符模式的对象
- 语法：
  - 定义正则表达式: Pattern.compile(regString,);
  - 表达式的模式:Matcher matcher = ptn.matcher("需要匹配的数据");
  - 验证：matcher.matches()

装箱拆箱

拆箱、装箱

- JDK提供了对所有基本数据类型的包装类

byte	------->	Byte	字节包装类
char	------->	Character	字符包装类
short	------->	Short	短整型包装类
int	------->	Integer	整型包装类
long	------->	Long	长整型包装类
double	------->	Double	双精度包装类
float	------->	Float	单精度包装类
boolean	------->	Boolean	布尔类型包装类

枚举类型

枚举类

- 为什么要用枚举
- 语法
- 好处
- 注意事项

```
enum Color{
    枚举成员;
    方法();
    构造();
}
```

Date、Calendar

Date类

- Date类
  - API文档中的大部分方法均已过时，不建议使用
- 日期格式应该使用SimpleDateFormat类；
- 提取时间分量的方法应该使用Calendar类