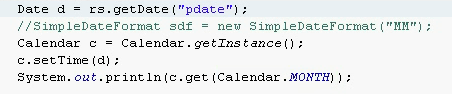
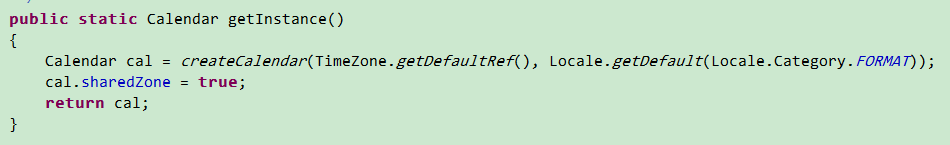
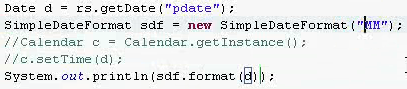
1. 涉及到的类：java.util.Date();java.text.SimpleDateFormat();java.util.Calendar
2. 例：



不能new Calendar的对象，因为**Calendar是抽象的**，但我们还是能够不受限制的调用Calendar的静态方法，这是因为静态的方法是在类上，而不是在某个特定的实例上。所以我们对Calendar调用**getInstance()会返回给我们具体子类的实例**。那是某种继承过Calendar(也就是Calendar的多态变化版本)并且会依据合约来响应Calendar应有的方法。



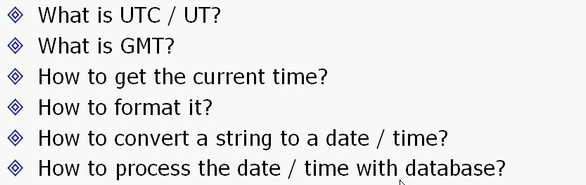




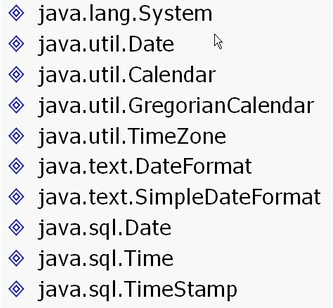
rs.getDate():只能得到date信息，不能得到时间信息

rs.getTimeStamp():都能得到。常用

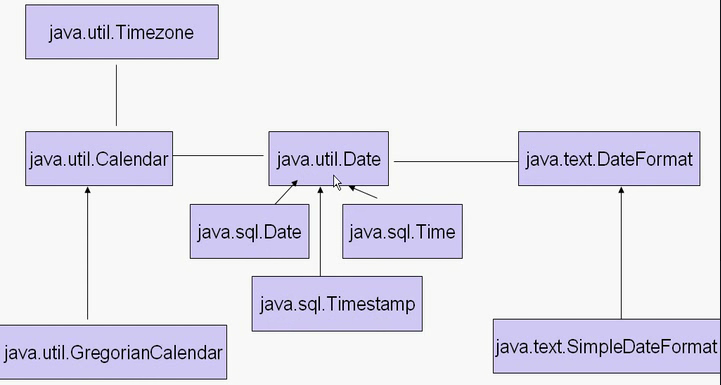
1. 总结：



* 1. UTC：世界时间(按标准原子钟)=UT：UTC科学上的简称=GMT：格林威治时间
  2. 得到当前系统时间

note:Timestamp

以上类之间的关系：

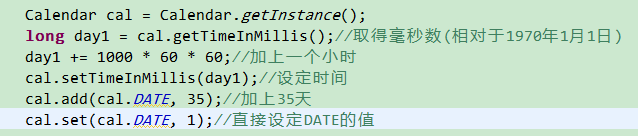




* 1. 怎样将字符串转换为日期



1. 使用Calendar类：



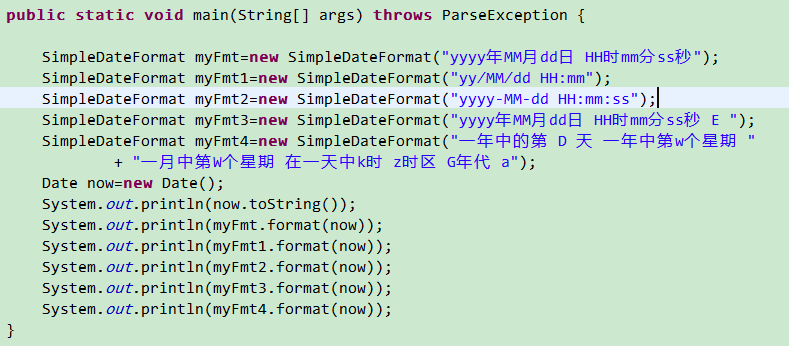
1. 使用**SimpleDateFormat**格式化日期字符串：

SimpleDateFormat 是一个以国别敏感的方式格式化和分析数据的具体类。 它允许格式化 (date -> text)、语法分析 (text -> date)和标准化。

**SimpleDateFormat函数语法**：

|  |  |
| --- | --- |
| G | 年代标识符 |
| y | 年 |
| M | 月 |
| d | 日 |
| h | 时 在上午或下午(1-12) |
| H | 时 在一天中(0-23) |
| m | 分 |
| s | 秒 |
| S | 毫秒 |
| E | 星期 |
| D | 一年中的第几天 |
| F | 一月中第几个星期几 |
| w | 一年中的第几个星期几 |
| W | 一月中的第几个星期几 |
| a | 上午/下午 标记符 |
| k | 时 在一天中(1-24) |
| K | 是 在上午或下午(0-11) |
| z | 时区 |

例：



输出：

