**GregorianCalendar类**

Calendar类实现了公历日历，GregorianCalendar是Calendar类的一个具体实现。

Calendar 的getInstance（）方法返回一个默认用当前的语言环境和时区初始化的GregorianCalendar对象。GregorianCalendar定义了两个字段：AD和BC。这是代表公历定义的两个时代。

下面列出GregorianCalendar对象的几个构造方法：

序号 构造函数和说明

1 GregorianCalendar()

在具有默认语言环境的默认时区内使用当前时间构造一个默认的 GregorianCalendar。

2 GregorianCalendar(int year, int month, int date)

在具有默认语言环境的默认时区内构造一个带有给定日期设置的 GregorianCalendar

3 GregorianCalendar(int year, int month, int date, int hour, int minute)

为具有默认语言环境的默认时区构造一个具有给定日期和时间设置的 GregorianCalendar。

4 GregorianCalendar(int year, int month, int date, int hour, int minute, int second)

为具有默认语言环境的默认时区构造一个具有给定日期和时间设置的 GregorianCalendar。

5 GregorianCalendar(Locale aLocale)

在具有给定语言环境的默认时区内构造一个基于当前时间的 GregorianCalendar。

6 GregorianCalendar(TimeZone zone)

在具有默认语言环境的给定时区内构造一个基于当前时间的 GregorianCalendar。

7 GregorianCalendar(TimeZone zone, Locale aLocale)

在具有给定语言环境的给定时区内构造一个基于当前时间的 GregorianCalendar。

这里是GregorianCalendar 类提供的一些有用的方法列表：

序号 方法和说明

1 void add(int field, int amount)

根据日历规则，将指定的（有符号的）时间量添加到给定的日历字段中。

2 protected void computeFields()

转换UTC毫秒值为时间域值

3 protected void computeTime()

覆盖Calendar ，转换时间域值为UTC毫秒值

4 boolean equals(Object obj)

比较此 GregorianCalendar 与指定的 Object。

5 int get(int field)

获取指定字段的时间值

6 int getActualMaximum(int field)

返回当前日期，给定字段的最大值

7 int getActualMinimum(int field)

返回当前日期，给定字段的最小值

8 int getGreatestMinimum(int field)

返回此 GregorianCalendar 实例给定日历字段的最高的最小值。

9 Date getGregorianChange()

获得格里高利历的更改日期。

10 int getLeastMaximum(int field)

返回此 GregorianCalendar 实例给定日历字段的最低的最大值

11 int getMaximum(int field)

返回此 GregorianCalendar 实例的给定日历字段的最大值。

12 Date getTime()

获取日历当前时间。

13 long getTimeInMillis()

获取用长整型表示的日历的当前时间

14 TimeZone getTimeZone()

获取时区。

15 int getMinimum(int field)

返回给定字段的最小值。

16 int hashCode()

重写hashCode.

17 boolean isLeapYear(int year)

确定给定的年份是否为闰年。

18 void roll(int field, boolean up)

在给定的时间字段上添加或减去（上/下）单个时间单元，不更改更大的字段。

19 void set(int field, int value)

用给定的值设置时间字段。

20 void set(int year, int month, int date)

设置年、月、日的值。

21 void set(int year, int month, int date, int hour, int minute)

设置年、月、日、小时、分钟的值。

22 void set(int year, int month, int date, int hour, int minute, int second)

设置年、月、日、小时、分钟、秒的值。

23 void setGregorianChange(Date date)

设置 GregorianCalendar 的更改日期。

24 void setTime(Date date)

用给定的日期设置Calendar的当前时间。

25 void setTimeInMillis(long millis)

用给定的long型毫秒数设置Calendar的当前时间。

26 void setTimeZone(TimeZone value)

用给定时区值设置当前时区。

27 String toString()

返回代表日历的字符串。

实例

实例

import java.util.\*;

public class GregorianCalendarDemo {

public static void main(String[] args) {

String months[] = {

"Jan", "Feb", "Mar", "Apr",

"May", "Jun", "Jul", "Aug",

"Sep", "Oct", "Nov", "Dec"};

int year;

// 初始化 Gregorian 日历

// 使用当前时间和日期

// 默认为本地时间和时区

GregorianCalendar gcalendar = new GregorianCalendar();

// 显示当前时间和日期的信息

System.out.print("Date: ");

System.out.print(months[gcalendar.get(Calendar.MONTH)]);

System.out.print(" " + gcalendar.get(Calendar.DATE) + " ");

System.out.println(year = gcalendar.get(Calendar.YEAR));

System.out.print("Time: ");

System.out.print(gcalendar.get(Calendar.HOUR) + ":");

System.out.print(gcalendar.get(Calendar.MINUTE) + ":");

System.out.println(gcalendar.get(Calendar.SECOND));

// 测试当前年份是否为闰年

if(gcalendar.isLeapYear(year)) {

System.out.println("当前年份是闰年");

}

else {

System.out.println("当前年份不是闰年");

}

}

}

运行实例 »

以上实例编译运行结果如下：

Date: Apr 22 2009

Time: 11:25:27

当前年份不是闰年