**实验3 Java类库和常用类**

**一. 实验目的及实验环境**

1理解类库的概念，掌握分析、应用类库中的类的方法。

2熟练掌握Math类的常用方法。熟悉Random类的常用方法。

3理解String类的特性，熟练掌握String类的常用方法。

4能用Date类创建对象，熟练掌握Date类的常用方法。

5熟练掌握SimpleDateFormat解析日期和设置日期输出格式。

6学会查阅Java API在线参考文档和离线文档的方法。

**二. 实验内容**

1 基本内容（实验前请及时熟悉如下相关内容）

1）练习使用Math类的常用方法。

2）应用String类编程练习。

3）编写程序应用Random类生成随机数。

4）练习使用Date类的常用方法。

5）查阅Java API在线参考文档和下载Java API离线文档。

示例1. 应用SimpleDateFormat类的程序示例如下,共同学们模仿参考。

**import** java.text.\*;

**import** java.util.Date;

**public** **class** FormatDateTime {

**public** **static** **void** main(String[] args) {

SimpleDateFormat myFmt = **new** SimpleDateFormat("yyyy年MM月dd日 HH时mm分ss秒");

SimpleDateFormat myFmt1 = **new** SimpleDateFormat("yy/MM/dd HH:mm");

SimpleDateFormat myFmt2 = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss");// 等价于now.toLocaleString()

SimpleDateFormat myFmt3 = **new** SimpleDateFormat("yyyy年MM月dd日 HH时mm分ss秒 E ");

SimpleDateFormat myFmt4 = **new** SimpleDateFormat("一年中的第 D 天 一年中第w个星期 一月中第W个星期 在一天中k时 z时区");

Date now = **new** Date();//当前时间

System.***out***.println(myFmt.format(now));

System.***out***.println(myFmt1.format(now));

System.***out***.println(myFmt2.format(now));

System.***out***.println(myFmt3.format(now));

System.***out***.println(myFmt4.format(now));

System.***out***.println(now.~~toGMTString~~());//The method toGMTString() from the type Date is deprecated.

System.***out***.println(now.~~toLocaleString~~());

System.***out***.println(now.toString());

}

}

示例2. 应用GregorianCalendar类的程序示例如下,共同学们模仿参考。

**import** java.util.GregorianCalendar;

**import** java.util.Date;

**import** java.text.DateFormat;

//public class DateExample5 {

**public** **class** TestGregorian {

**public** **static** **void** main(String[] args) {

DateFormat dateFormat = DateFormat.*getDateInstance*(DateFormat.***FULL***);

// Create our Gregorian Calendar.

GregorianCalendar cal = **new** GregorianCalendar();

// Set the date and time of our calendar

// to the system´s date and time

cal.setTime(**new** Date());

System.***out***.println("System Date: " + dateFormat.format(cal.getTime()));

// Set the day of week to FRIDAY

cal.set(GregorianCalendar.***DAY\_OF\_WEEK***, GregorianCalendar.***FRIDAY***);

System.***out***.println("After Setting Day of Week to Friday: " + dateFormat.format(cal.getTime()));

**int** friday13Counter = 0;

**while** (friday13Counter <= 10) {

// Go to the next Friday by adding 7 days.

cal.add(GregorianCalendar.***DAY\_OF\_MONTH***, 7);

// If the day of month is 13 we have

// another Friday the 13th.

**if** (cal.get(GregorianCalendar.***DAY\_OF\_MONTH***) == 13) {

friday13Counter++;

System.***out***.println(dateFormat.format(cal.getTime()));

}

}

}

}

**2 综合实验：**

**2.1 朱晓龙2011《Java语言程序设计》P169: 1** . 使用Date类中的方法，编程实现从1949年10月1日至2016年8月15日之间相隔的天数。

**2.2 朱晓龙2011《Java语言程序设计》P169:** 2. 使用System类中的方法，编程实现数组的复制。

**2.3 (Y. Daniel Liang英文版八版P296：8.3**\***)** (Using the Date class) Write a program that creates a Date object, sets its elapsed time to 10000, 100000, 10000000, 10000000, 100000000, 1000000000, 10000000000, 100000000000, and displays the date and time using the toString() method, respectively.

****

**2.****4(Y. Daniel Liang英文版八版P296：8.4\*)** (Using the Random class) Write a program that creates a Random object with seed 1000 and displays the first 50 random integers between 0 and 100 using the nextInt(100) method.

****

**2.5 (Y. Daniel Liang英文版八版P298：8.10\*)** (Algebra: quadratic equations) Design a class named QuadraticEquation for a quadratic equation The class contains:

■ Private data fields a, b, and c that represents three coefficients.

■ A constructor for the arguments for a, b, and c.

■ Three get methods for a, b, and c.

■ A method named getDiscriminant() that returns the discriminant, which is b2-4ac

■ The methods named getRoot1() and getRoot2() for returning two roots of the equation



These methods are useful only if the discriminant is nonnegative. Let these methods

return 0 if the discriminant is negative.

Draw the UML diagram for the class. Implement the class. Write a test program

that prompts the user to enter values for a, b, and c and displays the result based on

the discriminant. If the discriminant is positive, display the two roots. If the discriminant is 0, display the one root. Otherwise, display “The equation has no roots.”

See Exercise 3.1 for sample runs.





**2.6附加题（供学有余力同学完成，平时成绩有加分！☻）**

(**Y. Daniel Liang英文版八版**P337：9.11\*\*) (Sorting characters in a string) Write a method that returns a sorted string using the following header:

public static String sort(String s)

For example, sort("acb") returns abc.

Write a test program that prompts the user to enter a string and displays the sorted string.



**2.7附加题（供学有余力同学完成，平时成绩有加分！☻）**

(**Y. Daniel Liang英文版八版**P338：9.15\*) (Finding the number of uppercase letters in a string) Write a program that passes a string to the main method and displays the number of uppercase letters in a string.

