

# 第三次作业

作业提交格式： 与第二次作业提交格式相同。

## 1. Date

**问题描述：** 创建一个日期类 Date，其功能包括：

(1) 能够以不同的格式输出日期，如：

MM/DD/YYYY	(String, String, String)
June 6, 2016	(String, int, int)
2016 年 7 月 6 日	(int, int, int)

(2) 定义三个重载的构造函数，在创建 Date 对象时，可用以上三种格式来进行初始化。

(3) 实现 Date 类的成员函数 `int dateDistance(Date d)`，用于计算两个日期相差的天数。

**说明：**

(1) 定义 Date 类的成员函数 `void print()`，用于输出日期。其中输入出的格式由构造函数决定。

(2) 需要保证构造的日期是合法的，即不能出现 1999 年 2 月 29 日。

**要求：**

编写测试类 Test，分别通过三种构造函数创建三个 Date 对象，然后输出对应的格式。日期值任选。然后计算任意两个日期之间的距离。

## 2. 银行账户

创建 SavingsAccount 类，表示银行账户。

(1) 由于银行的存款年利率是固定的，与特定的账户无关。因此，定义一个 static 类型的类变量 `annualInterestRate`，用来存放年利率；

(2) 该类的每一个对象会包含一个 private 类型的实例变量 `savingsBalance`，表示该客户的当前存款余额；

(3) 定义一个成员方法 `calculateMonthlyInterest`，用来计算客户每个月的利息收入，计算方法为 `savingsBalance*annualInterestRate/12`，该利息随后要加入到 `savingsBalance` 中；

(4) 定义一个成员方法 `modifyInterestRate`，用来将 `annualInterestRate` 设定为一个新的值。请思考该成员方法是否应该为 `static` 类型；

(5) 编写一个程序来测试 `SavingsAccount` 类，创建 2 个 `SavingsAccount` 对象 `saver1` 和 `saver2`，初始存款分别为 2000.0 和 3000.0。将 `annualInterestRate` 设置为 4%，然后计算每个账户的当月利息和新的存款余额（增加了利息收入）。然后再把 `annualInterestRate` 修改为 5%，并且计算每个账户下个月的利息和新的存款余额。