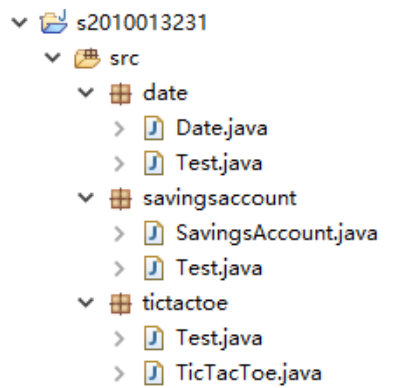


# Java 程序设计课程作业 2

实验要求：

1. 严格按照所给的类名，函数名进行命名，不符合要求的命名视为错误。
2. 每个小题放置在不同的包中，包的命名为功能类的命名，具体说明见课程文件中“JAVA新建工程和包的使用”，包名需要小写。
3. 测试类和功能类要分开，为单独的文件，集体说明见“Test类的说明和使用”。
4. 每个类文件（.java）必须有package信息。
5. 上交的文件夹的命名统一为“s 学号”，例如张三学号为 2010013231，那么上交的文件夹名称为 s2010013231，打包上传。

本次作业上交的格式如图：



## 1. Tic-Tac-Toe

问题描述

创建一个 TicTacToe 类，用于 Tic-Tac-Toe 游戏。

- (1)用字符界面来实现即可，不要求图形用户界面
- (2)该类包含一个 3×3 的二维整型数组，其访问类型为 private
- (3)定义一个构造函数，用来将该数组初始化为 0
- (4)允许 2 个玩家，当第 1 个玩家选定一个位置后，在该位置摆放一个 1；当第 2 个玩家选定一个位置后，在该位置摆放一个 2；
- (5)在每一步后，判断游戏是否结束，是否出现赢家



要求：

- (1) 编写测试类 Test:

```
package TicTacToe;
public class Test
{
    public static void main(String[] args)
    {
        //TO DO
    }
}
```

其中 args[0]包含输入的字符串，输入字符串格式示例为 :(1,1)(2,2)(3,3) 表示如下的

落子顺序：

1		

1		
	2	

1		
	2	
		1

输出结果为 **true** 或者 **false**，**true** 代表按照规定的落子顺序比赛结束，**false** 代表比赛没有结束。可以通过命令行运行 `java Test "(1,2)(3,3)(1,1)(2,2)"` 测试程序。Eclipse可以在Run -> Run Configurations...-> Arguments-> Programs arguments中填入 (1, 2) (3, 3) (1, 1) (2, 2)测试程序。

## 2. 日期

### 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：

五月/二十四日/二〇一四年

June 14, 2014

2015 年 3 月 6 日

(String, String, String)

(String, int, int)

(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。

提示：

- (1) 本题需要用到方法的重载（**overload**）

- (2) 若要比较两个 **String** 字符串是否相同，可用 `s1.equals(s2)`，结果为 **true** 或 **false**

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，然后针对每个 **Date** 对象，通过三种不同的格式分别输出每个对象。日期值任选。

## 3. 银行账户

### 问题描述

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 **Test** 类测试 **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），并输出。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，并且输出。