Java程序设计平时作业

重要说明

- 请严格按照所给的类名、函数名进行命名。函数需要严格按照给定的名字、参数、返回值定义和实现。**严格区分大小写，不符合要求的命名视为错误。**
- 每个小题放置在不同的包中，包的命名为功能类的命名，包名需要小写。同学需要在包中实现对应功能类。
- 每个类文件(.java)必须有 package 信息。

作业说明

- 在`src`目录下，已经新建好了各题目的目录，你需要按照题目要求**新建**相应的`.java`文件并**编写**相应的类和成员函数。
- 本次作业提供相应的测试类，完成对应的题目之后，编译并运行对应包下的测试类，自行测试。
- 在实验报告中，需要提交Test.java的**运行结果截图**；对于比较复杂的题目，也可以在文档中描述实现思路(不强制要求)。

提交要求

- 提交内容：需要提交**源代码**和**实验报告**
- 作业文件夹请打包成`zip`格式上传；上交的作业的根目录为以学号+姓名命名的文件夹，例如张三学号为 2000123456，那么该文件夹格式如图所示

```
└─2000123456_张三
    | reoport.pdf
    |
    └─src
        ├─complex
        |   Complex.java
        |   Test.java
        |
        ├─hugeinteger
        |   HugeInteger.java
        |   Test.java
        |
        └─tictactoe
            TicTacToe.java
            Test.java
```

测试类使用说明

作业中的一些题会提供测试类。测试类会在每道题对应的包下，命名为 Test.java。测试类会调用同学们编写的功能类，同学们在编写完每一题的功能类后， 编译运行整个包，就可以得到功能类的运行结果。如果编译运行成功，那么说明同学 编写的功能类的接口是正确的。一些注意事项：

- **测试类不需要同学们编写和修改**
- 测试类可能会包含一些样例检查功能类是否编写正确。但是在作业批改中，没有特别说明的情况下，可能会有更多的样例测试功能类是否编写正确。

Homework 2

本次作业的测试类包含所有的测试用例

Problem 1: Complex

在包 `complex` 中创建功能类 `Complex` ，用来进行复数的算术运算。

- 该类应包含两个公共成员变量，即实部 `realPart` 和虚部 `imaginaryPart`，均为实数类型。
- 定义一个构造函数，用来对复数进行初始化。参数为两个实数，分别对应实部和虚部的初始值。
- 定义 `print` 成员函数，无参数，返回一个字符串，表示打印当前复数的格式，格式为：X+Yi，小数点后保留三位。
- 定义 add 成员函数，实现两个复数的加法，参数为一个 Complex 对象，无返回值。例如：

```
Complex A = new Complex(1, 1);
Complex B = new Complex(2, 2);
A.add(B);
System.out.println(A.print()); // output 3.000+3.000i
```

  此时，A 为 3+3i，B 为 2+2i。
- 定义 `sub` 成员函数，实现两个复数的减法，参数和返回值同 `add`。
- 定义 `multi` 成员函数，实现两个复数的乘法，参数和返回值同 `add`。
- 定义 `div` 成员函数，实现两个复数的除法(题目保证不会除0)，参数和返回值同 `add`。

Problem 2: HugeInteger

在包 `hugeinteger` 中创建功能类 `HugeInteger` ，该类用来存放和操作一个不超过 40 位的大整数。

- 定义一个构造函数，用来对大整数进行初始化。参数为一个字符串。
- 定义 `input` 成员函数，实现大整数的重新赋值。参数为一个字符串，无返回值。
- 定义 `output` 成员函数，将大整数输出到屏幕上。无参数无返回值。
- 定义 `add` 成员函数，实现两个大整数的加法。参数为一个 `HugeInteger` 对象，无返回值，例如：

```
HugeInteger A = new HugeInteger("12345");
HugeInteger B = new HugeInteger("1234");
A.add(B);
```

  此时，A 为13579，B为 1234。
- 定义若干大整数关系运算的成员函数，包括 `isEqualTo` （等于，=）、`isNotEqualTo` （不等于，≠）、`isGreaterThan` （大于，>）、`isLessThan` （小 于，<）、`isGreaterThanOrEqualTo` （大于等 于，≥）和 `isLessThanOrEqualTo` （小于等于，≤）。这些函数的参数为一个 `HugeInteger` 对象， 返回值为一个布尔类型，表示关系运算的结果，例如：

```
HugeInteger A = new HugeInteger("12345");
HugeInteger B = new HugeInteger("1234");
```

那么此时 A.isGreaterThan(B) 的结果应当为 True ， 表示 12345>1234。

题目保证不会出现负数

Problem 3: TicTacToe

在包 `tictactoe` 中创建功能类 `TicTacToe` ， 实现Tic-Tac-Toe 游戏。

- 该类包含一个 3×3 的二维数组表示棋盘，其访问类型为 private 。
- 定义一个构造函数，用来将棋盘数组初始化。
- 定义 `place` 函数，用来模拟某个玩家对某个位置进行了标记。 该函数参数为三个整数： `player` 、 `row` 、 `column` 。其中 `player` 的值为 1 或 2，表示这是玩家 1 还是玩家 2 进行的操作； `row` 和 `column` 分别表示这一步操作格子的行和列，取值都是 0~2。该函数需要返回一个整数，表示该操作后棋盘的状态（0~3）：
  - 如果此时没有玩家胜出，返回 0；
  - 如果此时玩家 1 胜出，返回 1；
  - 如果此时玩家 2 胜出，返回 2；
  - 如果这一步操作在一个已经被标记过的位置，则忽略这次操作，返回 3。
- 某位玩家胜出，当且仅当某行、某列或某条对角线上的三个位置都是该玩家的标记