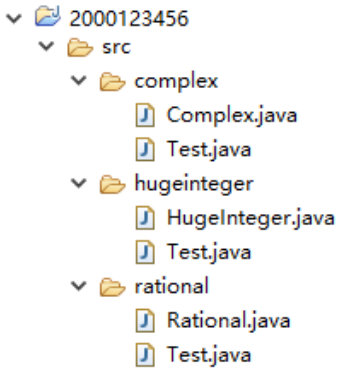
Java 语言程序设计作业 1

【实验要求】

1. 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。
2. 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。
3. 作业文件夹请打包上传；上交的作业中需包含以学号命名的文件夹，例如张三学号为 2000123456，那么该文件夹格式如图所示



【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：

1. 测试类不需要同学们编写和修改。
2. 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，会有更多的样例测试功能类是否编写正确。
3. 提交的文件夹中可以保留或不保留测试类。

第一题：复数

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

(1) 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。
(2) 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。
(3) 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);
```

此时，A 为 3+3i，B 为 2+2i。
(4) 定义 sub 成员函数，实现两个复数的减法，参数和返回值同 add。
(5) 定义 print 成员函数，无参数，返回值为一个字符串，表示当前复数的格式，打印格式为：1+1i。

第二题：分数

在包 rational 中创建功能类 Rational，用来执行分数的算术运算。

(1) 该类应当有两个成员变量，即分子和分母，均为整数类型，
(2) 定义一个构造函数，用来对分子和分母进行初始化。参数为两个整数，分别对应分子和分母的初始值。该构造函数须对输入参数进行约减。例如，若给定的分数为 2/4（即分子为 2，分母为 4），那么要把它约减为 1/2，然后存储在相应的成员变量中，即分子为 1，分母为 2。
(3) 定义 add 成员函数，实现两个分数的加法，其结果也应该是约减形式。参数为一个 Rational 对象，无返回值。支持负数运算，(3)~(6)同。例如：

```
Rational A = new Rational(1, 2);
Rational B = new Rational(1, 3);
A.add(B);
```

此时，A 为 5/6，B 为 1/3。
(4) 定义 sub 成员函数，实现两个分数的减法，结果为约减形式。
(5) 定义 mul 成员函数，实现两个分数的乘法，结果为约减形式。
(6) 定义 div 成员函数，实现两个分数的除法，结果为约减形式。
(7) 定义 printRational 函数，以分数形式（分子/分母）打印该分数，结果为约减形式，如 1/2。该函数无参数无返回值。如果该分数为整数则打印只整数部分，如 2/1 则打印 2。
(8) 定义 printReal 函数，以实数形式打印该分数，如 0.5。该函数无参数无返回值。

注意：

(1) 不需要考虑分母为 0 的情况。
(2) 约减方法建议使用“辗转相除法”。

第三题：大整数

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。

(1) 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。
(2) 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。
(3) 定义 output 成员函数，将大整数输出到屏幕上。无参数无返回值。
(4) 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。
(5) 定义 sub 成员函数，实现两个大整数的减法。参数和返回值同 add 函数。
(6) 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。

注意：

(1) 大整数运算需要考虑到正负，可以使用单独的变量存储符号位。
(2) 大数的输入：正数形式为“12345”，负数形式为“-123456”。