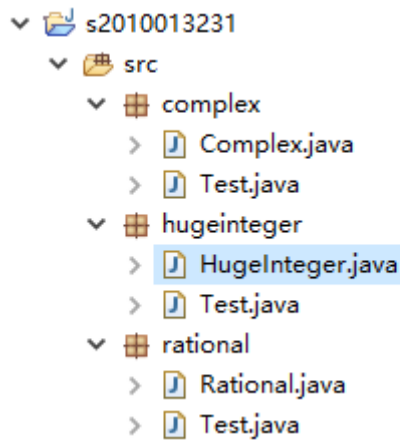


第二次作业

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，具体说明见课程文件中“**Test 类的说明和使用**”。
4. 每个类文件(.java)必须有 **package** 信息。例如 **Complex.java**,第一行为 **package complex;**
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：



注意：本次作业的函数需要严格按照给定的名字，参数，返回值定义和实现

1. 复数

问题描述

创建一个复数类 **Complex**，用来执行复数的算术运算，并编写一个程序来测试该类。

- (1) 该类有两个成员变量，即实部 **realPart** 和虚部 **imaginaryPart**，均为实数类型
- (2) 定义一个构造函数，用来对类对象进行初始化，构造函数的函数原型为：

Complex(double r, double i);

- (3) 定义一个 **add** 成员函数，实现两个复数的加法

Complex add(Complex); (注意参数是 **Complex** 类型，且仅有一个参数)

例如：

```
Complex A = new Complex(1,1);
```

```
Complex B = new Complex(1,1);
```

```
Complex C = A.add(B);
```

C 的值为 $2+2i$, A 为 $1+i$, B 为 $1+i$

后续函数功能与此例子相同

- (4) 定义一个 **sub** 成员函数，实现两个复数的减法

Complex sub(Complex);

- (5) 定义一个 **void print()** 成员函数，在屏幕上打印当前复数，打印格式为： $1+1i$

2. 分数

问题描述

创建一个分数类 **Rational**，用来执行分数的算术运算，并编写一个程序来测试该类。

(1) 该类有两个成员变量，即分子和分母，均为整数类型

(2) 定义一个构造函数，用来对类对象进行初始化。该构造函数必须对输入参数进行约减。例如，若给定的分数为 $\frac{2}{4}$ （即分子为 2，分母为 4），那么要把它约减为 $\frac{1}{2}$ ，然后存储在相应的成员变量中，即分子为 1，分母为 2。

函数原型：**Rational** (int 分子, int 分母);

(3) 定义一个 **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；

函数原型：**Rational add(Rational)**，支持负数运算，(3) - (6) 同。

(4) 定义一个 **sub** 成员函数，实现两个分数的减法，结果为约减形式

(5) 定义一个 **mul** 成员函数，实现两个分数的乘法，结果为约减形式

(6) 定义一个 **div** 成员函数，实现两个分数的除法，结果为约减形式

(7) 定义一个 **printRational** 函数，以分数形式打印该分数，如 $\frac{1}{2}$ ，函数原型为：

void printRational()

(8) 定义一个 **printReal** 函数，以实数形式打印该分数，如 0.5，函数原型为：

void printReal()

注意：

1. $\frac{2}{1}$ 不是约减形式,需要化简为 2
2. 需要考虑分母为 0 和除 0 的情况

3. 大整数

问题描述

创建一个大整数类 **HugeInteger**，该类用一个 40 个元素的数组来存放一个大整数（最多不超过 40 位）。

构造函数原型：

HugeInteger(String);

(1) 定义几个大整数算术运算的成员函数，包括 **input**、**output**、**add** 和 **sub**，

add, **sub** 的函数原型为 **HugeInteger operation(HugeInteger)**

input 的函数原型为 **void input(String)** 功能是改变大数的值为 **String** 的值。

output 的函数原型为 **void output()** 功能是将大数输出,即打印到屏幕上。

(2) 定义几个大整数关系运算的成员函数，包括 **isEqualTo**、**isNotEqualTo**、**isGreaterThan**、**isLessThan**、**isGreaterThanOrEqualTo** 和 **isLessThanOrEqualTo**。每个函数的返回值为布尔类型。

函数原型为 **boolean operation(HugeInteger)**

注意：

1. 大数运算需要考虑到正负，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”
3. 如果运算溢出，截取低 40 位结果。
4. 包名 **hugeinteger** 中 **i** 为小写，类名 **HugeInteger** 中 **I** 为大写。