**第二次检查要求**

**1. 要求综述**

  第二周要求实现racket计算器的数值运算的大部分内容，新增复数（Complex）类，具体类要求参考Scheme文档。

**2. 需实现类**

  本周要求大家实现Complex类。Float类和Rational类都可以看成是特殊的Complex类。关于Complex类的exactness参见后文。

**3. 需实现函数**

|  |  |
| --- | --- |
| **功能** | **过程** |
| 基本算术运算 | +, -, \*, /, abs, quotient, remainder, modulo, gcd, lcm, expt(输入保证double范围), sqrt(输入保证double范围) |
| 分数运算 | numerator, denominator |
| 近似值 | floor, ceiling, truncate, round |
| 精确性 | inexact->exact, exact->inexact |
| 最大与最小值 | max, min |
| 复数运算 | real-part, imag-part |

**4. 特殊要求**

a. 输入输出要求

需额外支持复数的输入及输出，格式如下：

X±Xi

  上述函数具体参数要求详见Scheme说明文档。

e.g(第一行为输入，第二行为输出):

(real-part 1/3+1/2i)

1/3

b. 疑难点详解

  A)关于复数类的exactness说明如下：

1：无论real-part还是imag-part，只要二者中有一个是inexact属性（即Float类），则整体复数为inexact属性。

2：只要real-part或imag-part中有一个或两个是inexact，则运算、存储和输出时需将real-part和imag-part统一转成inexact属性。

B)关于min、max函数中同时涉及精确值和非精确值时的情况

详见Scheme文档中的说明规则，同时请明确对绝大部分函数来说都存在exact和inexact的priority比较问题(有inexact参与运算的函数一般结果也是inexact)。

C) 关于expt()和sqrt()等函数如何实现。

对于该类无法手工实现函数，可借助C++系统函数。计算时请将参数转化为double类型数据后调用系统函数求的结果，并转化为Scheme中的Float类型。