**ComplexNumber 类设计说明**

**一、类设计概述**

* **类名**：ComplexNumber
* **核心功能**：实现复数的基本运算（加减乘除/共轭/模长/指数运算）
* **设计原则**：
  + 不可变性：所有运算返回新对象不修改原对象
  + 运算符重载：支持复数与复数/实数之间的混合运算
  + 数学规范：严格遵循复数数学定义（如 $a+bi$ 的共轭为 $a-bi$）

**二、核心方法实现**

| **方法名** | **功能描述** | **数学公式实现** |
| --- | --- | --- |
| conjugate() | 计算共轭复数 | $ a+bi → a-bi $ |
| abs() | 计算模长（绝对值） | $ |
| add(...) | 支持复数+复数、复数+实数 | $ (a+bi)+(c+di)= (a+c)+(b+d)i $ |
| sub(...) | 支持复数-复数、复数-实数 | $ (a+bi)-(c+di)= (a-c)+(b-d)i $ |
| mult(...) | 支持复数*复数、复数*实数 | $ (a+bi)(c+di)= (ac-bd)+(ad+bc)i $ |
| reciprocal() | 计算倒数（分母为模长平方） | $ \frac{1}{a+bi} = \frac{a-bi}{a^2+b^2} $ |
| div(...) | 支持复数/复数、复数/实数 | $ \frac{a+bi}{c+di} = \frac{(a+bi)(c-di)}{c^2+d^2} $ |
| exponentiation() | 实现复数指数运算（欧拉公式） | $ e^{a+bi} = e^a(\cos b + i\sin b) $ |

**三、运算符重载机制**

* 采用多态方法实现双参数类型适配：

cangjie

public func add(other: ComplexNumber) // 复数+复数

public func add(other: Float64) // 复数+实数

**四、测试用例设计**

1. **基础验证**：创建 $3+4i$ 和 $1+2i$ 作为测试对象
2. **边界测试**：包含与实数的混合运算（如加减乘除）
3. **格式化输出**：使用 format(".2") 保留两位小数
4. **完整覆盖**：验证所有方法调用及输出格式

**五、设计亮点**

* **模块化设计**：每个方法独立实现单一数学功能
* **可扩展性**：方法命名规范便于后续扩展（如添加三角函数）
* **用户体验**：通过字符串插值实现直观的复数输出格式
* **性能优化**：重复使用的中间计算结果（如模长平方）提取为局部变量

**六、依赖关系**

* 基础库依赖：
  + std.math.\*：提供 sqrt/pow/exp/sin/cos 等数学函数
  + std.format.\*：支持浮点数格式化输出

该设计完整覆盖了复数运算的核心需求，通过面向对象的设计实现了良好的可读性和可维护性，测试用例充分验证了各方法的正确性。