作业

1. Why should you have minimum scope for variables?
   1. 数据安全，不会发生被外部修改的情况
   2. 性能影响，会跟随退栈及时释放
   3. 如果有多个相同对象，那么命名会显得非常麻烦且杂乱
2. Why should you understand performance of String Concatenation?
   1. 直接使用+连接的字符串，其每个本身都是一个独立的对象存在在常量池中所以会产生非常多的对象，性能很差
   2. 由于字符串(String)底层本身是维护的一个char数组，所以使用StringBuilder来拼接字符串可以做到直接操作char数组，也不会在常量池产生多余的对象，性能最优
   3. StringBuilder本身不是线程安全的，如果要保证线程安全可以使用StringBuffer，其底层实现与StringBuilder相同，只不过操作的方法使用了synchronized修饰
3. What are the best practices with Exception Handling?
   1. 使用try-with-resource语句来关闭资源
   2. 尽可能使用自定义的异常
   3. 自定义异常要使用文档说明
   4. 自定义异常中，定义描述信息
   5. 捕获异常时，捕获具体的异常并分开处理
   6. 不能捕获Throwable，他包含了无法处理的错误
   7. 不要无视异常，捕获了应予以处理
   8. 不要捕获了异常，记录完了又继续抛出了
   9. 包装异常，但注意不要丢失原本异常的栈信息
4. When is it recommended to prefer Unchecked Exceptions?

Checked Exception应该用在可被预见的并且不可避免的错误，其可以被调用方处理的情况下，其他情况就应该使用Unchecked Exceptions

Unchecked Exceptions都属于运行时异常，是程序在运行时才可能触发异常，在编译阶段是不可能预期的异常，比如参数检查，编译时是不可能得知传入的参数到底是否合法。

1. When do you use a Marker Interface?

在早期Java版本中，标记接口是唯一的声明元数据的方式，从1.5开始可以使用注解来代替标记接口灵活的维护元数据。但是标记接口相比于注解具有2个优点：

* 1. 标记接口定义了一个由标记类实例实现的类型，注解则不会。这样标记接口类型的存在使得可以在在编译时捕获错误，注解则只能在运行时捕获
  2. 标记接口可以更精确的定位目标。注解中声明其可以使用在类上，那么他就可以使用在任意的类中，不能精确的标记其仅适用于特定接口的实现

如果想编写一个或多个只接受具有此标记的对象的方法，那么应该优先使用标记接口

1. Why are ENUMS important for Readable Code?
   1. 代码中硬编码一些常量并不合适，这些魔法值都不利于阅读，比如在做判断时，判断变量是否等于1，在阅读代码时并不能知道其代表了什么含义
   2. 枚举更利于维护，维护人员可以通过枚举名称轻易的理解其含义，而不用需要一个详细的手册来将关系一一对上，另外如果需要对某个对应关系作修改，也可以灵活的进行全局的变更
2. Why should you minimize mutability?
   1. 不可变对象非常简单，一个不可变对象可以完全只处于一种状态，也就是被创建时的状态
   2. 不可变对象本质上是线程安全的，可以被自由的共享，而且可以共享内部信息
   3. 不可变对象可以为其他对象提供良好的构建
   4. 不可变对象提供了免费的原子失败机制
   5. 减小对象的可变性，可以减少对象存在的状态，可以更容易的分析对象以及降低出错的可能行
3. What is functional programming?

函数式编程是一种编程模式，与面向对象，面向过程一样。

函数式编程，它的数据与函数是松耦合的，函数隐藏了它们的实现，抽象的是函数，可以将函数组合起来表达。

其是一种结构化的编程，主要思想就是把运算过程尽量携程一系列嵌套的函数调用

1. Why should you prefer Builder Pattern to build complex objects?

如果一个对象有太多的参数，其中一些是必须的，另一些不是必须的，那么我们就需要太多的构造器来区分这些需求，如果参数很多，那么将会是指数级增长的复杂度

如果使用建造者模式，我们就不需要知道其内部组成细节，将对象本身与对象的创建过程解耦，使得相同的创建过程可以创建不同的对象。而且可以是用户精确的控制对象的创建过程。如果需要增加新的参数，也不需要修改缘由的类库代码。

1. Why should you avoid floats for Calculations?

float，double类型在内存中并不能精确的表示一个具体的值，在运算中会损失精度，使用BigDecimal

1. Why should you build the riskiest high priority features first?

这样能避免项目进行到中部，发现有一些特性无法实现而导致整个项目流产，前期投入全部打了水漂