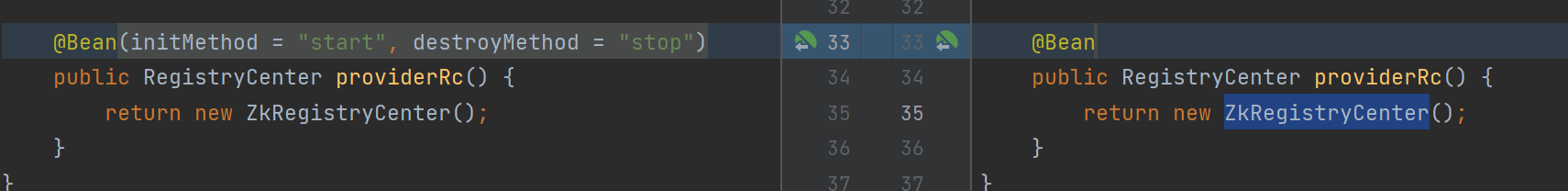
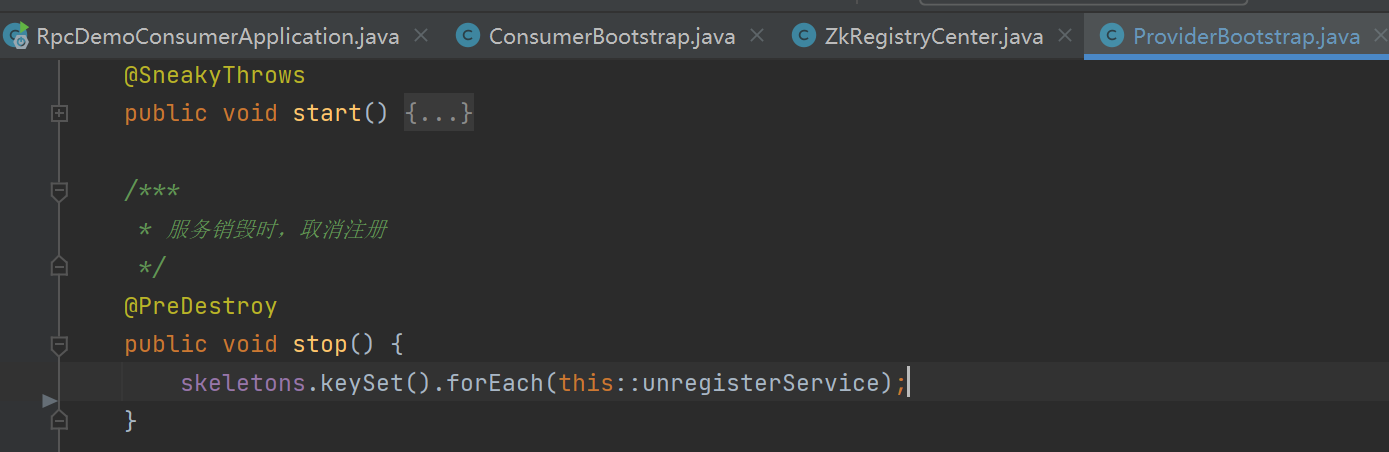
**06 代码重构（20240320）**

1. ZK注册中心ZkRegistryCenter的的启动与销毁

重构前，初始化和销毁方法依赖于bean的生命周期。



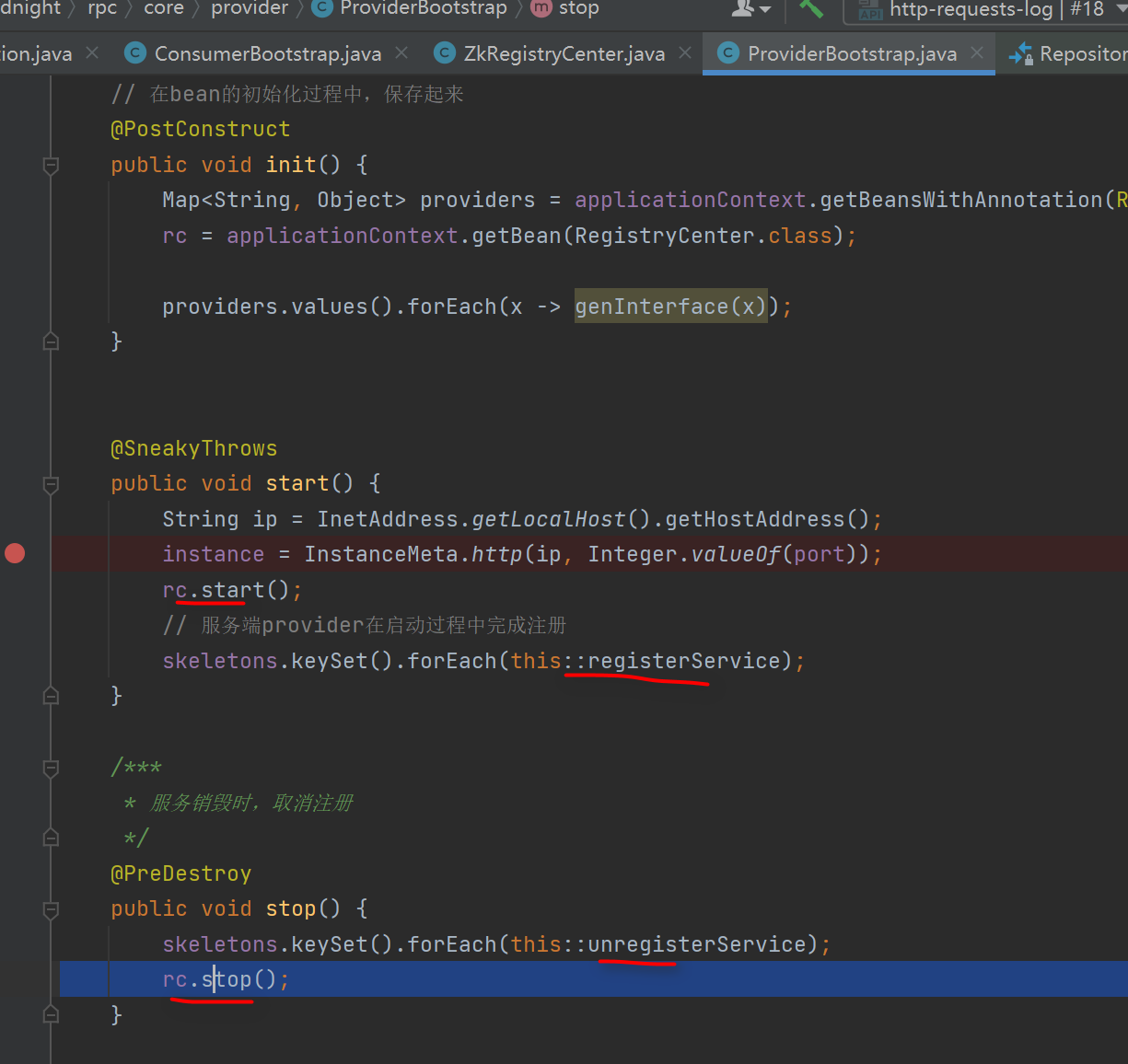
这样带来的问题是当整个服务停止时，销毁方法先执行，客户端和zk服务端已经断开连接，而服务取消注册的逻辑再执行时，就不会成功。



由于取消注册使用的是quietly()方式，出错了也不会报错，最终就是服务没有取消成功，消费者还可能调用到。

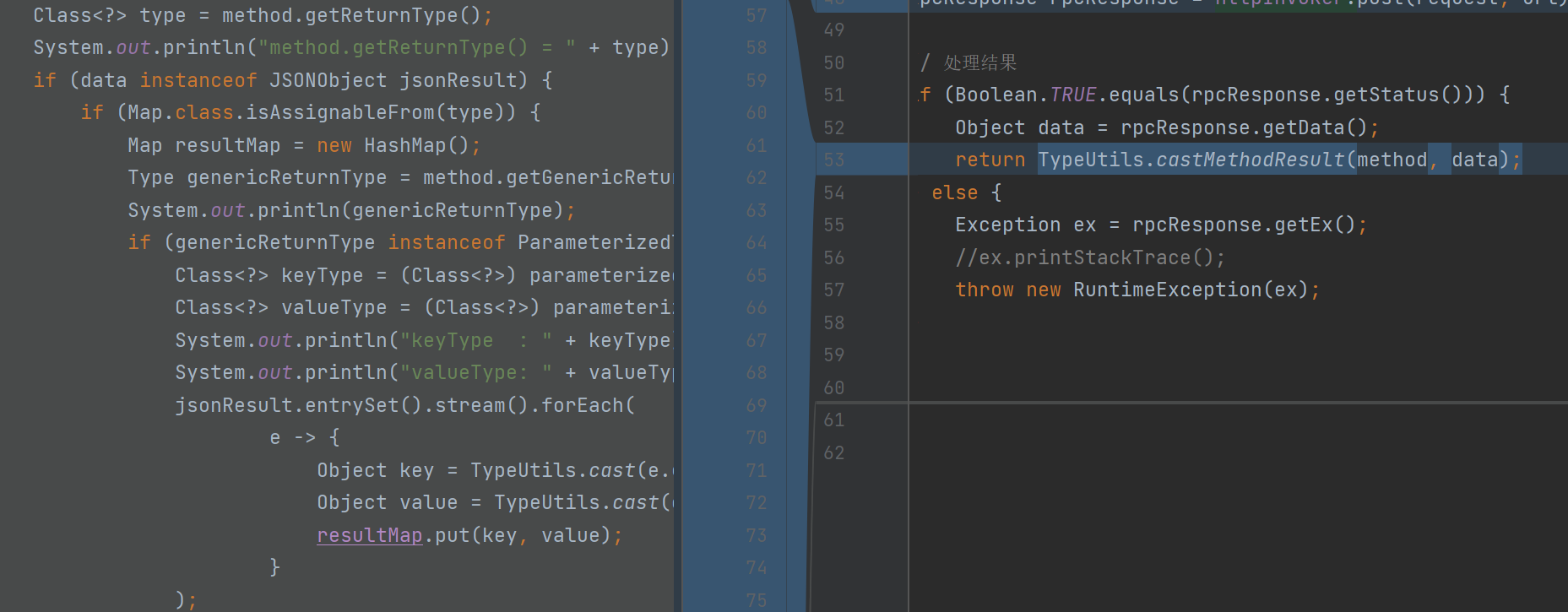


重构后，注册中心先启动成功，在进行服务注册。服务销毁时，先取消注册，在关闭连接。



1. 抽取类型转换逻辑到工具类TypeUtils中。

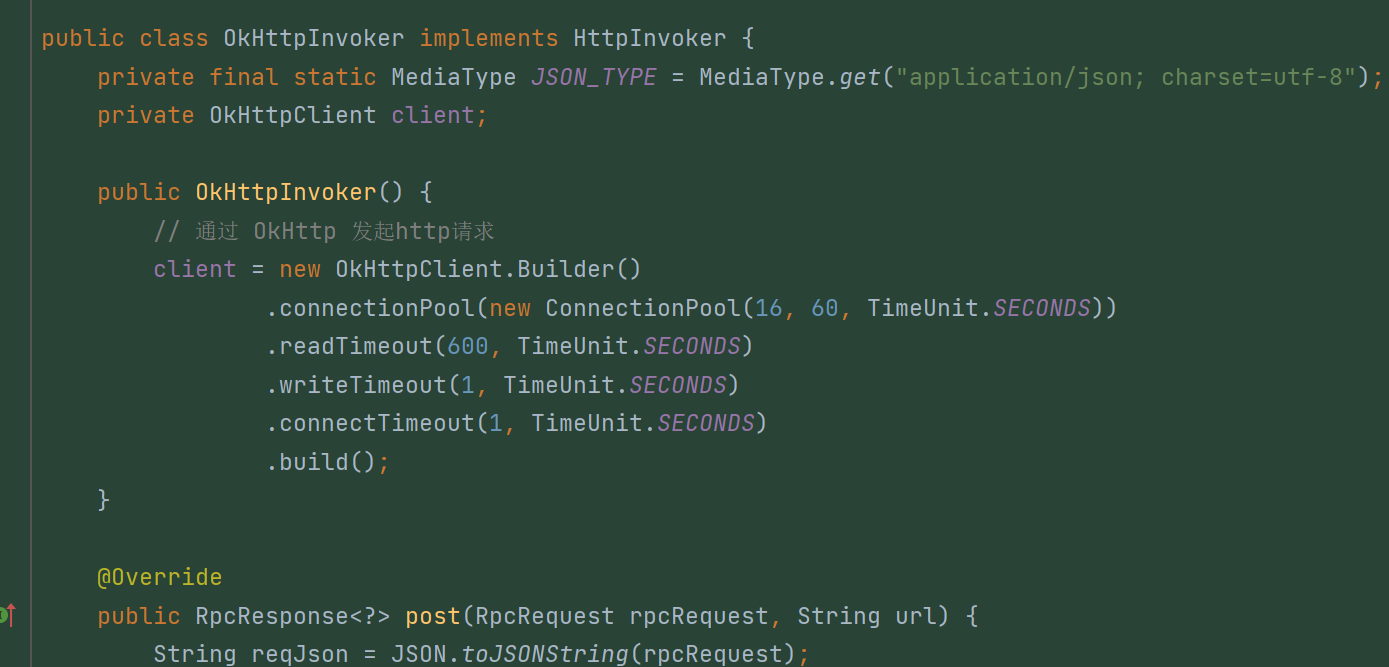
这样带来的好处是业务功能和非业务功能逻辑分离，提高代码可读性和可复用性。



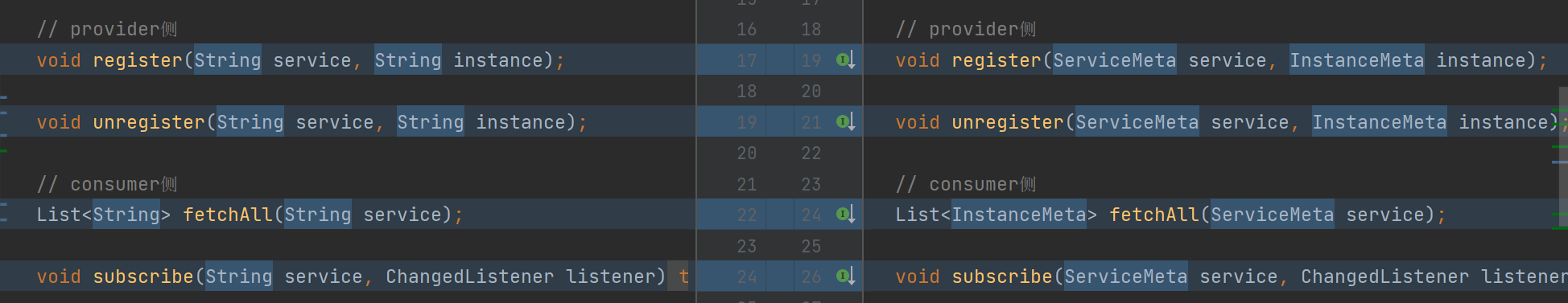
1. 网络客户端封装为接口



定义HttpInvoker接口，添加当前使用的方式OkHttp作为客户端。这样基于接口设计带来的好处时想使用其他客户端时就很方便替换，不需要改变上层逻辑，添加新的实现类即可。



1. 封装ServiceMeta和InstanceMeta



使用ServiceMeta代替String类型，来表示一个服务的元数据，表达含义更加丰富，支持更多非功能性场景。

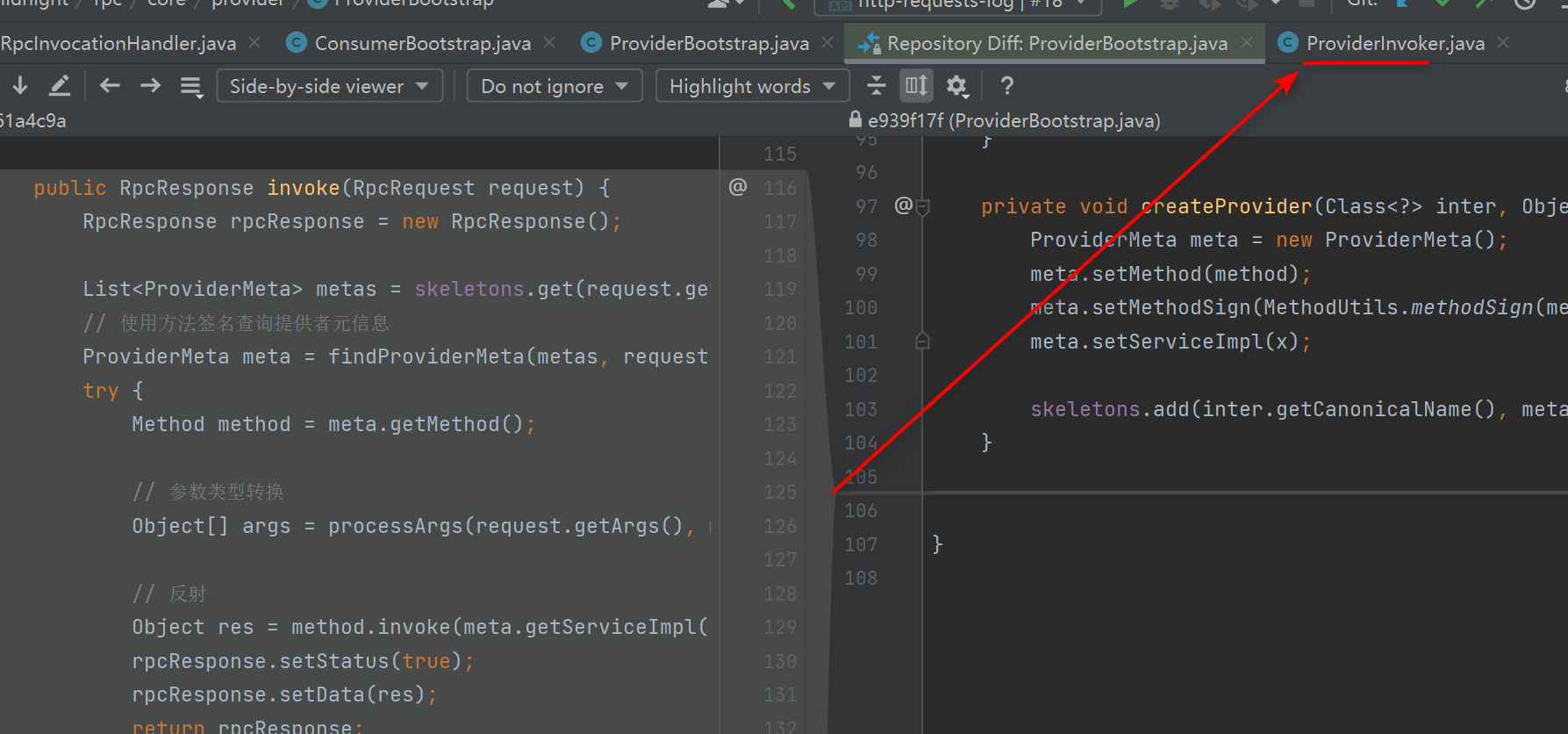


同理，使用InstanceMeta代替String类型，来表示一个实例的元数据，表达含义更加丰富，支持更多非功能性场景。



1. 封装ProviderInvoker

将服务提供者的调用逻辑抽出独立的类，职责更单一。



此次，重构的要点

1. 业务功能和非功能性逻辑分开，提高可读性和可扩展性。
2. 使用保证类型替换String类型，丰富表达含义。
3. 工具类抽取，提高复用性。
4. 基于接口做设计，提高扩展性。
5. 类职责保持单一，解耦，类聚。

源码：https://github.com/midnight2104/midnight-rpc/tree/lesson6