1.依赖倒置原则:

1、高层模块不应该依赖底层模块，二者都应该依赖抽象。

2、抽象不应该依赖细节，细节应该依赖抽象。

3、依赖倒置的中心思想是面向接口编程。

4、依赖倒置原则是基于这样的设计理念：相对于细节的多变性，抽象的东西要稳定的多。以抽象为基础搭建的架构比以细节为基础搭建的架构要稳定的多。

5、使用接口或抽象类的目的是指定好规范，而不涉及任何具体的操作，把展现细节的任务交给他们的实现类来完成。

接触过的一些

比如spi，容器初始化：Map<String,String> map = new HashMap<String,String>();工厂模式，spring依赖注入，不注入具体对象，注入接口或者抽象类

好莱坞原则:

"别调用我们，我们会调用你"：高层组件对待低层组件的方式是“别调用我们，我们会调用你”

作用:

防止“依赖腐败”

应用：

模板方法模式就应用到了这个原则,模板方法类对它的子类说: 不要调用我们，我们会调用你

两者区别:

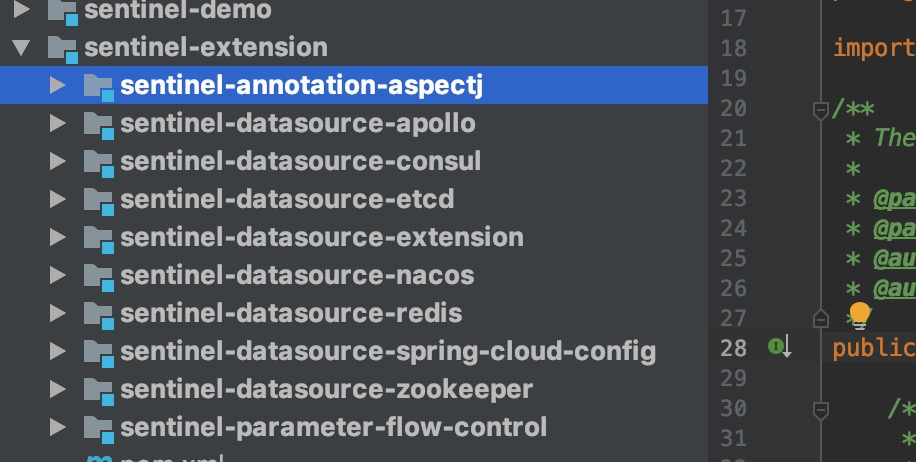
依赖倒置原则 面向接口或者抽象类编程

好莱坞原则 子类的(底层)尽量不要调用超类(父类)的方法

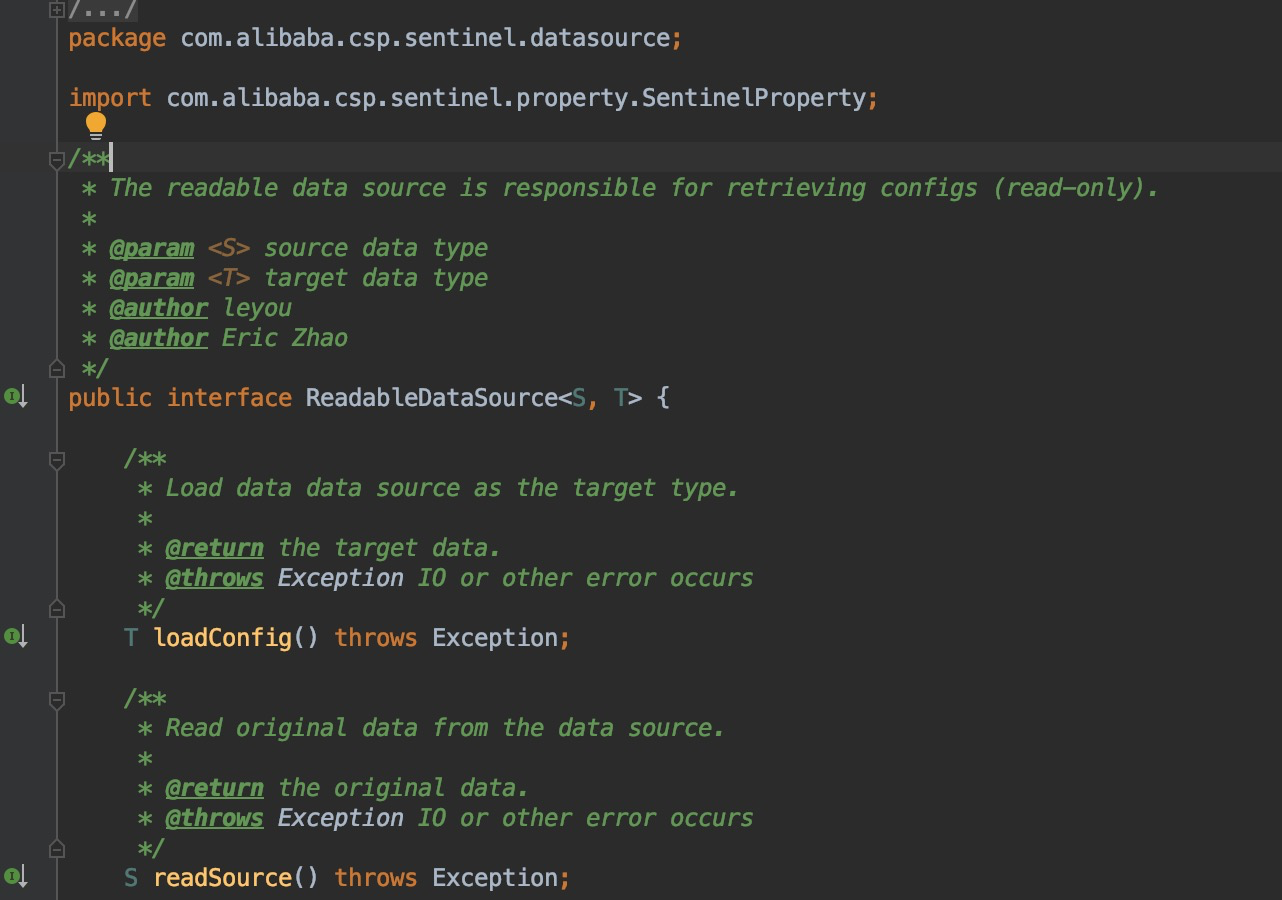
2.依赖倒置原则:

比如sentinel（最近一直看的阿里的一个框架）：

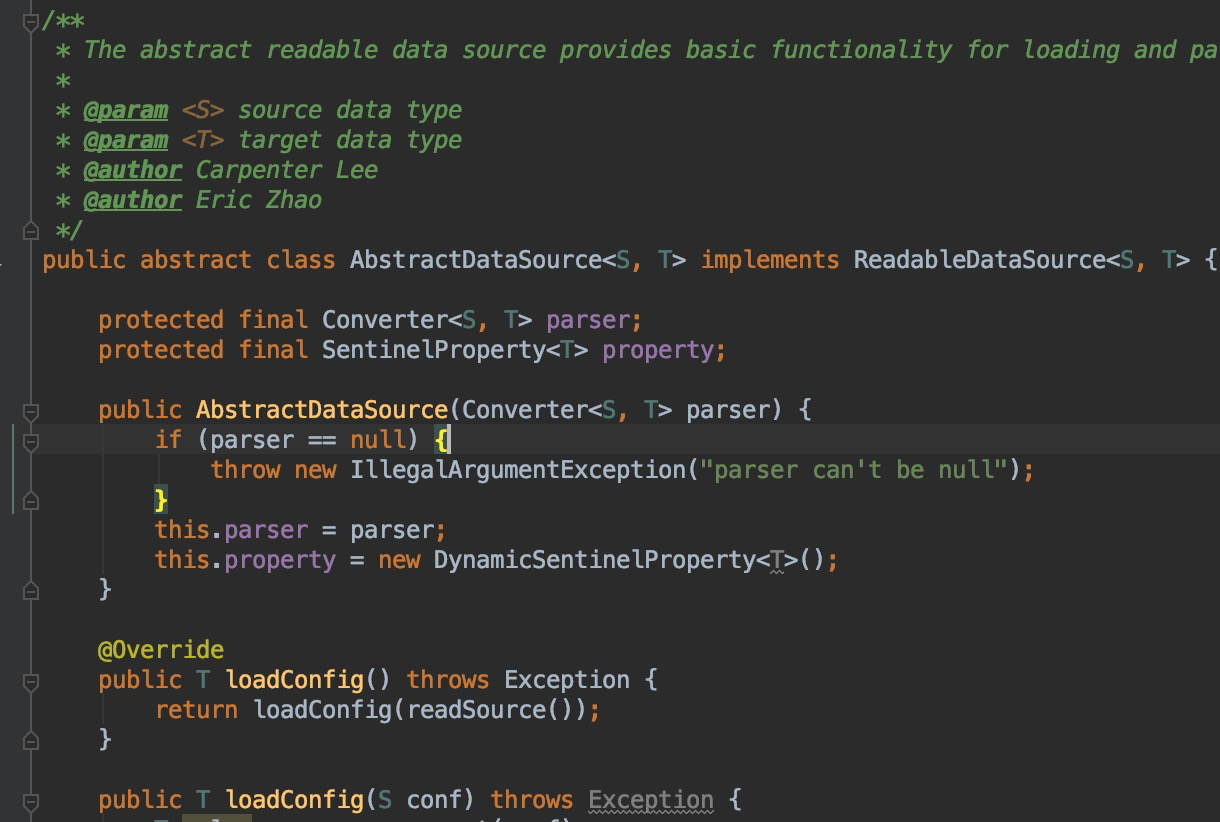
比如数据源的扩展:



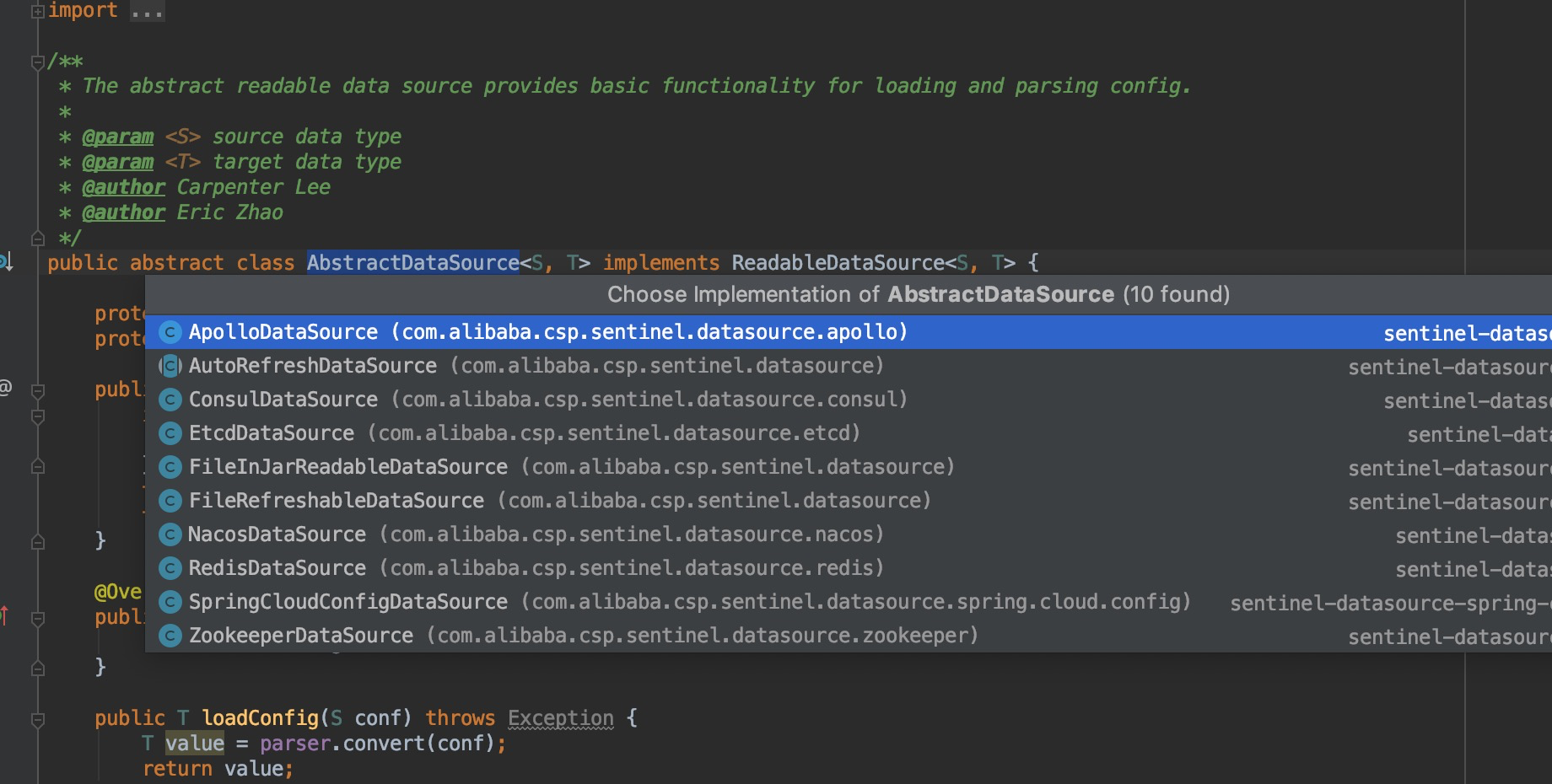
接口类:



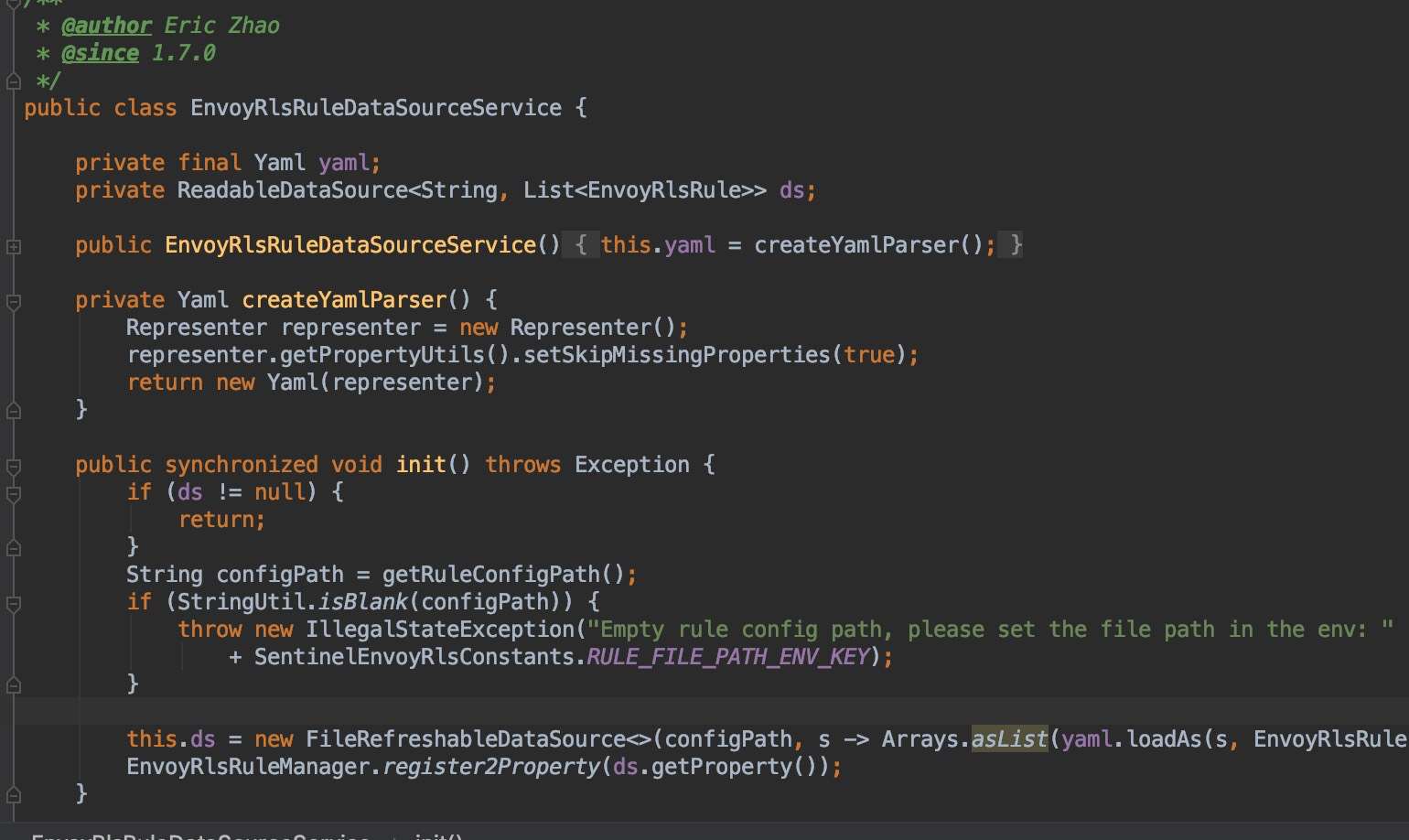
基于接口的抽象类:



所有的实现：



使用方式:



没有依赖具体的实现类