

- 创建一个吃东西接口Eat，定义eat方法
- 创建一个吃肉类EatMeat类实现Eat接口，重写eat方法
- 创建一个代理类EatProxy实现Eat接口，构造参数为Eat接口对象，注意，传入的对象可以是参数对象的实现类或继承子类
- 代理中定义before方法和after方法
- 代理类重写eat方法作为代理方法，方法中调用参数Eat接口的eat方法，在eat方法前后调用before和after方法，类似Spring的通知
- 静态代理的作用：对象只需要做好自己的事，代理类对对象的功能进行增强
- 举例：
  - EatMeat类中eat方法体为打印"吃肉"，EatProxy类的before方法体为打印"从冰箱取肉做熟"，after方法体为打印"收拾碗筷"，eat方法体为依次调用before方法、参数接口的eat方法、after方法
  - 实例化时，创建接口实现类对象，再创建代理类对象构造参数为接口实现类对象，代理类对象调用eat方法
- Thread类其实就是使用静态代理模式，它其实就是个代理类
- 它实现了Runnable接口，实例化Thread类需要传入Runnable接口实现类对象，该对象重写了run方法
- 最后Thread对象再调用start代理方法开启线程，本质上还是在执行run方法