```
| Dec | Edit View | Navigate | Code | Effector | Build | Run | Dods | Git | View | springe| Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen/Collegen
```

/4

静态代理确实实现了解耦,但是由于代码都写死了,完全不具备任何的灵活性。就拿日志功能来说,将来其他地方也需要附加日志,

那还得再声明更多个静态代理类,那就产生了大量重复的代码,日志功能还是分散的,没有统一管理。

提出进一步的需求:将日志功能集中到一个代理类中,将来有任何日志需求,都通过这一个代理类来实现.

这就需要使用动态代理技术了。

*/