Introducing Spring Aspect Oriented Programming



Cătălin Tudose PHD IN COMPUTER SCIENCE, JAVA AND WEB TECHNOLOGIES EXPERT https://www.linkedin.com/in/catalin-tudose-847667a1





XML schema based Spring AOP

AspectJ based Spring AOP

Using the main Spring AOP annotations





AspectJ based Spring AOP

- AspectJ annotations
- @Before annotation
- @After annotation





AspectJ based Spring AOP

- @Around annotation
- AOP mechanisms
- Creation of proxy objects

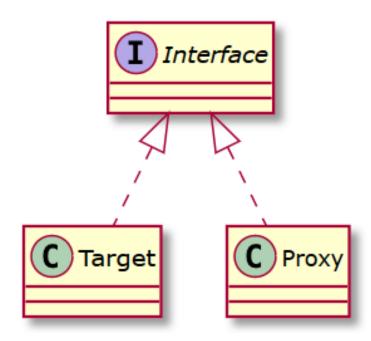


AOP Mechanisms

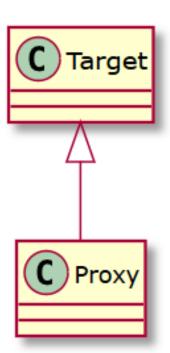
```
class PassengerDaoProxy implements PassengerDao
@Aspect
public class LoggingAspect {
                                                 public Passenger getPassenger(int id)
  @Around("execution(* *.*Passenger(..))")
                                                    Aspect logger = new LoggingAspect();
 public Object log ( ←
                                                    ProceedingJoinPoint joinpoint =
    ProceedingJoinPoint thisJoinPoint()
                                                              new ProceedingJoinPoint() {
         throws Throwable {
                                                       Object proceed() {
    String methodName =
                                                         return passengerDao.getPassenger(id);
       thisJoinPoint.getSignature().getName();
   Object[] methodArgs =
       thisJoinPoint.getArgs();
                                                    return logger.log(joinpoint);
    logger.debug("Call method " + methodName
              + " with arg " + methodArgs[0]);
   Object result = thisJoinPoint.proceed();
                                                class PassengerDaoImpl implements PassengerDao
    logger.debug("Method " + methodName
              + " returns " + result);
                                                  public Passenger getPassenger(int id) {
    return result;
```

Creating Proxy Objects

JDK dynamic proxy standard mechanism



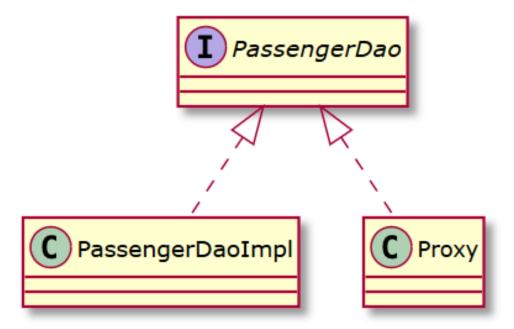
CGLIB proxy mechanism



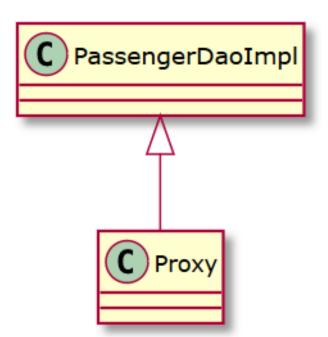


Creating Proxy Objects

JDK dynamic proxy standard mechanism



CGLIB proxy mechanism







Examples of implementing orthogonal behavior for normal execution

Examples of implementing orthogonal behavior for exceptions



Summary



XML schema based Spring AOP

AspectJ based Spring AOP

Examples of implementing orthogonal behavior for normal execution

Examples of implementing orthogonal behavior for exceptions

