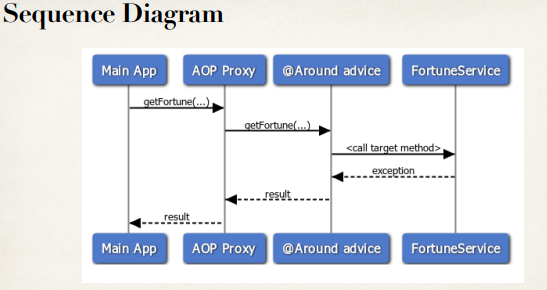
**44.3 Around Advice - Handle Exception**

**ProceedingJoinPoint – Revisited**:

* For an exception thrown from proceeding join point
  + We can handle / swallow /stop the exception
  + Or we can simply rethrow the exception
* This gives you fine-grained control over how the target method is called



**Handle Exception**:

// add a new advice for @Around

@Around("execution(\* com.ruhul.aopdemo.service.\*.getFortune(..))")

**public** Object aroundGetFortune(

ProceedingJoinPoint theProceedingJoinPoint) **throws** Throwable {

// print out which method we are advising on

String method = theProceedingJoinPoint.getSignature().toShortString();

myLogger.info("\n======> Executing @Around on method: " + method);

// get being timestamp

**long** being = System.*currentTimeMillis*();

// now, lets execute the method

Object result = **null**;

**try** {

result = theProceedingJoinPoint.proceed();

} **catch** (Exception e) {

// log the exception

myLogger.warning(e.getMessage());

// give user a message

result = "Major accident! But no worries, your private AOP

helicopter is on the way!!!";

}

// get end timestamp

**long** end = System.*currentTimeMillis*();

// compute duration and display it

**long** duration = end - being;

myLogger.info("\n======> Duration: " + duration / 1000.0 + " seconds");

**return** result;

}

44.3 Around Advice - Handle Exception