比较有价值的技术点：

1. 日志切面

*/\*\*  
 \** ***@Description:*** *日志注解  
 \*/*@Target(ElementType.*METHOD*)   
@Retention(RetentionPolicy.*RUNTIME*)  
@Documented  
public @interface LoggerManage {  
  
 public String description();  
}

*/\*\*  
 \** ***@Description:*** *日志管理  
 \*/*@Aspect  
@Service  
public class LoggerAdvice {  
  
 protected Logger logger = LoggerFactory.*getLogger*(this.getClass());  
  
 @Before("within(com.favorites..\*) && @annotation(loggerManage)")  
 public void addBeforeLogger(JoinPoint joinPoint, LoggerManage loggerManage) {  
 logger.info("执行 " + loggerManage.description() + " 开始");  
 logger.info(joinPoint.getSignature().toString());  
 logger.info(parseParames(joinPoint.getArgs()));  
 }  
   
 @AfterReturning("within(com.favorites..\*) && @annotation(loggerManage)")  
 public void addAfterReturningLogger(JoinPoint joinPoint, LoggerManage loggerManage) {  
 logger.info("执行 " + loggerManage.description() + " 结束");  
 }  
   
 @AfterThrowing(pointcut = "within(com.favorites..\*) && @annotation(loggerManage)", throwing = "ex")  
 public void addAfterThrowingLogger(JoinPoint joinPoint, LoggerManage loggerManage, Exception ex) {  
 logger.error("执行 " + loggerManage.description() + " 异常", ex);  
 }  
  
 private String parseParames(Object[] parames) {  
 if (null == parames || parames.length <= 0 || parames.length >1024) {  
 return "";  
 }  
 StringBuffer param = new StringBuffer("传入参数[{}] ");  
 for (Object obj : parames) {  
 param.append(ToStringBuilder.*reflectionToString*(obj)).append(" ");  
 }  
 return param.toString();  
 }  
   
}

日志打印样例：

