3. 以下是从 Dubbo 软件中摘取的代码（https://dubbo.apache.org/zh/docs/v2.7/dev/source/service-invoking-process/）。

```
/**
 * Arthas 反编译步骤：
 * 1. 启动 Arthas
 *    java -jar arthas-boot.jar
 *
 * 2. 输入编号选择进程
 *    Arthas 启动后，会打印 Java 应用进程列表，如下：
 *    [1]: 11232 org.jetbrains.jps.cmdline.Launcher
 *    [2]: 22370 org.jetbrains.jps.cmdline.Launcher
 *    [3]: 22371 com.alibaba.dubbo.demo.consumer.Consumer
 *    [4]: 22362 com.alibaba.dubbo.demo.provider.Provider
 *    [5]: 2074 org.apache.zookeeper.server.quorum.QuorumPeerMain
 * 这里输入编号 3，让 Arthas 关联到启动类为 com.....Consumer 的 Java 进程上
 *
 * 3. 由于 Demo 项目中只有一个服务接口，因此此接口的代理类类名为 proxy0，此时
使用 sc 命令搜索这个类名。
 *    $ sc *.proxy0
 *    com.alibaba.dubbo.common.bytecode.proxy0
 *
 * 4. 使用 jad 命令反编译 com.alibaba.dubbo.common.bytecode.proxy0
 *    $ jad com.alibaba.dubbo.common.bytecode.proxy0
 *
 * 更多使用方法请参考 Arthas 官方文档：
 *   https://alibaba.github.io/arthas/quick-start.html
 */
public class proxy0 implements ClassGenerator.DC, EchoService,
DemoService {
    // 方法数组
    public static Method[] methods;
    private InvocationHandler handler;

    public proxy0(InvocationHandler invocationHandler) {
        this.handler = invocationHandler;
    }

    public proxy0() {
    }

    public String sayHello(String string) {
        // 将参数存储到 Object 数组中
        Object[] arrobject = new Object[]{string};
```

```
        // 调用 InvocationHandler 实现类的 invoke 方法得到调用结果
        Object object = this.handler.invoke(this, methods[0],
arrobject);
        // 返回调用结果
        return (String)object;
    }

    /** 回声测试方法 */
    public Object $echo(Object object) {
        Object[] arrobject = new Object[]{object};
        Object object2 = this.handler.invoke(this, methods[1],
arrobject);
        return object2;
    }
}
```

如上，代理类的逻辑比较简单。首先将运行时参数存储到数组中，然后调用 InvocationHandler 接口实现类的 invoke 方法，得到调用结果，最后将结果转型并返回给调用方。

根据透明性的分类，此种透明性应为访问透明性，即隐藏数据表示形式的不同已经资源访问方式的不同。