Jedis测试操作

88322511@qq.com
bigdatalyn
2018/04/08

准备工作：

++

下载Eclipse+Maven并配置

1.Eclipse+Java下载解压即可

2.maven配置参考如下文章配置

Eclipse+Maven整合开发Java项目（一）➢Maven基础环境配置

https://www.cnblogs.com/xibei666/p/6706512.html

配置环境变量：

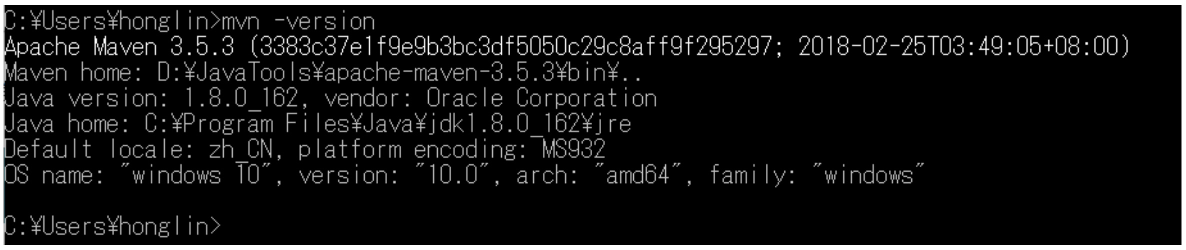
MAVEN_HOME

D:\JavaTools\apache-maven-3.5.3

PATH

%MAVEN_HOME%\bin

如下版本信息：

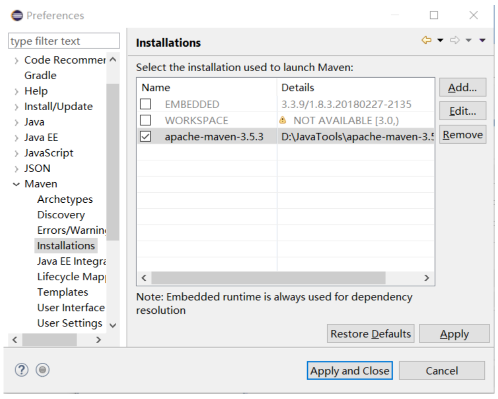


```
C:¥Users¥honglin>mvn -version
Apache Maven 3.5.3 (3383c37e1f9e9b3bc3df5050c29c8aff9f295297; 2018-02-25T03:49:05+08:00)
Maven home: D:¥JavaTools¥apache-maven-3.5.3¥bin¥..
Java version: 1.8.0_162, vendor: Oracle Corporation
Java home: C:¥Program Files¥Java¥jdk1.8.0_162¥jre
Default locale: zh_CN, platform encoding: MS932
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:¥Users¥honglin>
```

利用Eclipse+maven编译Jedis源码成jar包和源码jar包

http://www.mamicode.com/info-detail-505779.html

Eclipse：windows/preferences

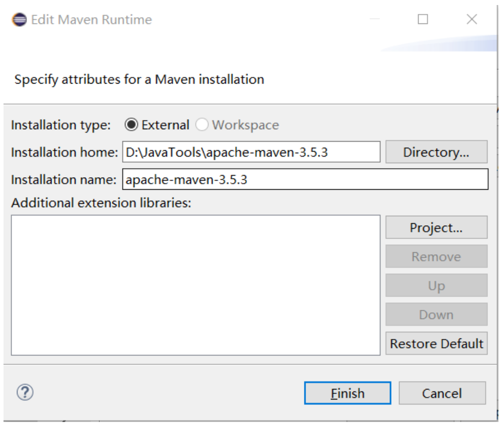


Installations

Select the installation used to launch Maven:

Name	Details
☐ EMBEDDED	3.3.9/1.8.3.20180227-2135
☐ WORKSPACE	NOT AVAILABLE [3.0,)
☑ apache-maven-3.5.3	D:\JavaTools\apache-maven-3.5

Note: Embedded runtime is always used for dependency resolution



Specify attributes for a Maven installation

Installation type: [✓ External] Workspace

Installation home: D:\JavaTools\apache-maven-3.5.3

Installation name: apache-maven-3.5.3

Additional extension libraries:

Java开发-Redis客户端Jedis

https://blog.csdn.net/fjssharpsword/article/details/50637061

关于Maven项目build时出现No compiler is provided in this environment的处理

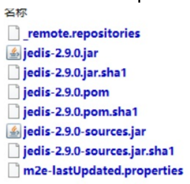
https://blog.csdn.net/lslk9898/article/details/73836745

下载源码 maven install生成jar包

https://github.com/xetorthio/jedis

生成了redis jar包

D:\JavaTools\apache-maven-salf\Restory\redis\clients\jedis\2.9.0\



名称
_remote.repositories
jedis-2.9.0.jar
jedis-2.9.0.jar.sha1
jedis-2.9.0.pom
jedis-2.9.0.pom.sha1
jedis-2.9.0-sources.jar
jedis-2.9.0-sources.jar.sha1
m2e-lastUpdated.properties

+++

测试工作：

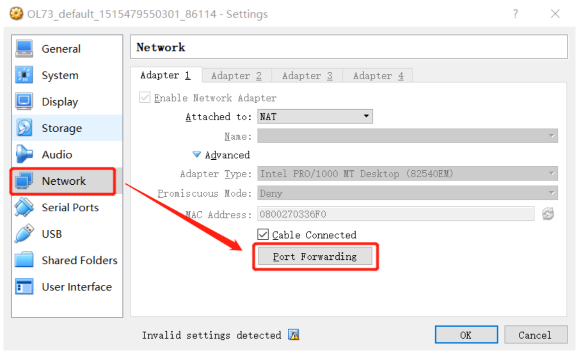
启动Reids服务器

+++

```
root@oraclelinux7:~# ps -ef | grep redis
root      9879     1  0 00:38 ?        00:00:03 /root/redis-3.0.6/src/redis-server *:6380
root     13941  8188  0 00:56 pts/0    00:00:00 grep --color=auto redis
root@oraclelinux7:~#
```

+++

virtualbox设置端口跳转（我的是NAT网络模式）



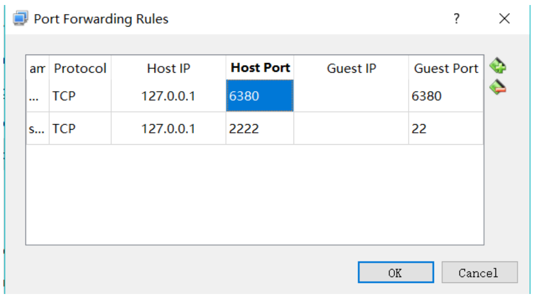
OL73_default_1515479550301_86114 - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

Enable Network Adapter
Attached to: NAT
Name:
Advanced
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Deny
MAC Address: 0800270336F0
Cable Connected
Port Forwarding

Invalid settings detected



Port Forwarding Rules

arr	Protocol	Host IP	Host Port	Guest IP	Guest Port
...	TCP	127.0.0.1	6380		6380
s...	TCP	127.0.0.1	2222		22

6380端口开通

另外主要linux服务器的防火墙需要关闭

```
root@oraclelinux7:~# systemctl status firewalld
仟 firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead) -> 保证是inactive（dead）模式
     Docs: man:firewalld(1)
```

如果启动的话通过systemctl stop firewalld关闭进行测试

+++

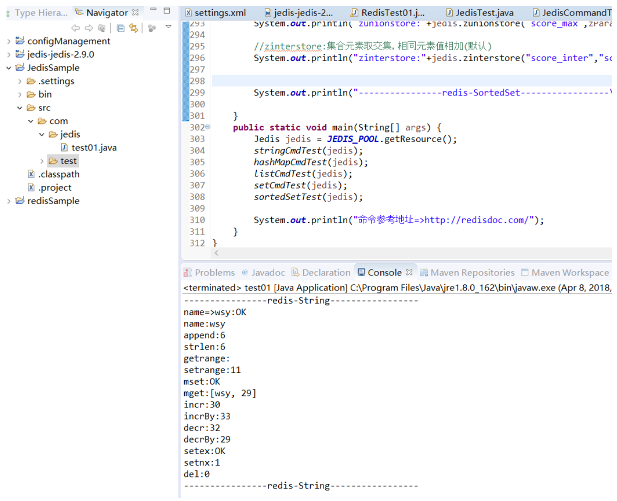
测试代码：

```
package com.jedis;

import redis.clients.jedis.*;

import java.util.HashMap;
import java.util.Map;
```

```
public class test01 {
    private static final JedisPool JEDIS_POOL;

    static {
        JedisPoolConfig config = new JedisPoolConfig();//redis连接池配置对象
        config.setMaxTotal(32);//最大连接数
        config.setMaxIdle(6);//闲置最大连接数
        config.setMinIdle(0);//闲置最小连接数
        config.setMaxWaitMillis(15000);//到达最大连接数后，调用者阻塞时间
        config.setMinEvictableIdleTimeMillis(300000);//连接空闲的最小时间，可能被移除
        config.setSoftMinEvictableIdleTimeMillis(-1);//连接空闲的最小时间，多余最小闲置连接的将被移除
        config.setNumTestsPerEvictionRun(3);//设置每次检查闲置的个数
        config.setTestOnBorrow(false);//申请连接时，是否检查连接有效
        config.setTestOnReturn(false);//返回连接时，是否检查连接有效
        config.setTestWhileIdle(false);//空闲超时,是否执行检查有效
        config.setTimeBetweenEvictionRunsMillis(60000);//空闲检查时间
        config.setBlockWhenExhausted(true);//当连接数耗尽，是否阻塞

        //连接池配置对象+ip+port+timeout+password+dbname
        JEDIS_POOL = new JedisPool(config,"127.0.0.1",6380,10000);//,null,1);
    }

    /**
     * redis中String类型操作：字符串+数字+bit
     */
    public static void stringCmdTest(Jedis jedis){
        System.out.println("----------------redis-String-----------------");
        //set:返回操作结果
        System.out.println("name=>wsy:"+jedis.set("name","wsy"));

        //get:value
        System.out.println("name:"+jedis.get("name"));

        //append:字符串长度
        System.out.println("append:"+jedis.append("name","_ss"));

        //strlen:字符串长度
        System.out.println("strlen:"+jedis.strlen("name"));

        //getrange:返回不包括起始坐标的值
        System.out.println("getrange:"+jedis.getrange("name", 10, 13));

        //setrange:从起始坐标考试替换，未替换的保持
        System.out.println("setrange:"+jedis.setrange("name", 10, "#"));

        //mset:批量设置，返回批量设置结果
        System.out.println("mset:"+jedis.mset("name","wsy","age", "29"));

        //mget:返回数组
        System.out.println("mget:"+jedis.mget("name","age"));

        //incr:value自增1后，返回value
        System.out.println("incr:"+jedis.incr("age"));

        //incr:value自增传参值后，返回value
        System.out.println("incrBy:"+jedis.incrBy("age",3));
        //decr:value自减1，返回value
        System.out.println("decr:"+jedis.decr("age"));

        //decrBy:value自减入参值，返回value
        System.out.println("decrBy:"+jedis.decrBy("age",3));

        //setex:设置key值+有效时间，如果key存在则覆盖value
        System.out.println("setex:"+jedis.setex("phone",10,"13600000001"));

        //setnx:当key不存在时，设置才成功
        System.out.println("setnx:"+jedis.setnx("address","china"));

        //del:删除对应key
        System.out.println("del:"+jedis.del("address1"));

        System.out.println("----------------redis-String-----------------\n");

    }

    /**
     * redis中hash类型常用操作
     * @param jedis
     */
    public static void hashMapCmdTest(Jedis jedis){
        System.out.println("----------------redis-HashMap-----------------");
        //hset:返回值为key为新返回1，为旧覆盖旧值返回0
        System.out.println("hset:"+jedis.hset("user","name","wangshaoyi"));

        Map map = new HashMap();
        map.put("name","wsy");
        map.put("age","29");
        //hmset:map对象
        System.out.println("hmset:" + jedis.hmset("user", map));

        //hexists:判断hashmap中key是否存在
        System.out.println("hexists:"+jedis.hexists("user","age"));

        //hget:获取map中key对应的value
        System.out.println("hget:"+jedis.hget("user","name"));

        //hgetAll:获取map中所有对象
        System.out.println("hgetAll:"+jedis.hgetAll("user"));

        //hkeys:获取map中所有key
        System.out.println("hkeys:"+jedis.hkeys("user"));

        //hvals:获取map中所有value
        System.out.println("hvals:"+jedis.hvals("user"));



        //hmget:批量获取keys的对象，返回List
        System.out.println("hmget:"+jedis.hmget("user","age","name"));

        //hlen:map的大小
        System.out.println("hlen:"+jedis.hlen("user"));

        //hdel:删除map中对应key,正确删除返回1
        System.out.println("hdel:"+jedis.hdel("user","age0"));
        System.out.println("----------------redis-HashMap-----------------\n");

    }

    public static void listCmdTest(Jedis jedis){
        System.out.println("----------------redis-List-----------------");
        //
        jedis.del("contacts");
        jedis.del("contacts_old");


        //lpush:批量头部插入，返回List的size
        System.out.println("lpush:"+jedis.lpush("contacts","xx","yy","zz"));

        //lpushx:单个头部插入，返回List的size
        System.out.println("lpushx:"+jedis.lpushx("contacts","aa"));

        //linsert:指定对象位置(前or后)插入
        System.out.println("linsert:"+jedis.linsert("contacts", BinaryClient.LIST_POSITION.BEFORE,"zz","bb"));

        //lset:将指定的位置设置值（替换旧值）
        System.out.println("lset:"+jedis.lset("contacts",2,"cc"));

        //lpop:链表头的对象
        System.out.println("lpop:"+jedis.lpop("contacts"));

        //lrange:获取list指定start、end位置value
        System.out.println("lrange:"+jedis.lrange("contacts",1,3));

        //ltrim:只剩start\end中list值，其余删除
        System.out.println("ltrim:"+jedis.ltrim("contacts",1,3));

        //lrem:删除list指定值（次数指定），返回删除个数
        System.out.println("lrem:"+jedis.lrem("contacts",2,"yy"));

        //rpoplpush:将源list尾部对象移到目标list对象头部
        System.out.println("rpoplpush:"+jedis.rpoplpush("contacts","contacts_old"));

        //rpush:在list尾部对象添加值
        System.out.println("rpush:"+jedis.rpush("contacts","aa","bb"));

        //rpop:移除在list尾部值，返回移除的对象
        System.out.println("rpop:"+jedis.rpop("contacts"));

        //brpop:阻塞尾部对象抛出，指定超时时间，返回抛出值
        System.out.println("brpop:"+jedis.brpop(1,"contacts"));

        System.out.println("blpop:"+jedis.blpop(1, "contacts"));

        System.out.println("blpop（阻塞1秒返回）:"+jedis.blpop(1, "contacts"));

        System.out.println("----------------redis-List-----------------\n");


    }

    public static void setCmdTest(Jedis jedis){
        System.out.println("----------------redis-Set-----------------");
        jedis.del("phones");
        jedis.del("phones_old");
        jedis.del("phones_old_1");
        jedis.del("phones_new");


        //sadd:集合添加元素,返回添加成功后数据
        System.out.println("sadd:"+jedis.sadd("phones","13600000001","13300000001"));
        System.out.println("sadd:"+jedis.sadd("phones","13600000002","13300000002"));

        //scard:返回集合中元素数
        System.out.println("scard:"+jedis.scard("phones"));

        jedis.sadd("phones_old","13600000002");
        jedis.sadd("phones_old_1","13300000001");

        //sdiff:首set与其他set之间的差集，返回差集值
        System.out.println("sdiff:"+jedis.sdiff("phones","phones_old","phones_old_1"));

        //sdiffstore:首set与其他set之间的差集保存至新set，返回差集数
        System.out.println("sdiffstore:"+jedis.sdiffstore("phones_new","phones","phones_old"));

        //sinter:返回集合的交集
        System.out.println("sinter:"+jedis.sinter("phones","phones_new"));

        //sismember:判断value是否为set的值
        System.out.println("sismember:"+jedis.sismember("phones","13600000001"));

        //smembers:返回集合中成员
        System.out.println("smembers:"+jedis.smembers("phones"));

        //smove:将首源set中元素移动目标set，返回移动数
        System.out.println("smove:"+jedis.smove("phones","phones_new","13600000002"));

        //spop:随机移除set的一元素，返回移除元素
        System.out.println("spop:"+jedis.spop("phones"));

        //srandmember:随机取出集合中一个元素
        System.out.println("srandmember:"+jedis.srandmember("phones_new"));

        //srem:删除集合中指定元素
        System.out.println("srem:"+jedis.srem("phones_new","13600000002"));

        //sunion:集合中并集
        System.out.println("sunion:"+jedis.sunion("phones","phones_new","phones_old"));

        System.out.println("----------------redis-Set-----------------\n");

    }

    public static void sortedSetTest(Jedis jedis){

        System.out.println("----------------redis-SortedSet-----------------");
        jedis.del("scores");
        jedis.del("scores_1");
        jedis.del("scores_total");
        jedis.del("score_inter");
        jedis.del("score_max");



        //zadd:sortedSet添加元素
        System.out.println("zadd:"+jedis.zadd("scores", 610.5, "xx"));
        jedis.zadd("scores", 630, "yy");

        //zcard:返回sortedset中元素数
        System.out.println("zcard:"+jedis.zcard("scores"));

        //zcount:返回指定分值（包括）的元素数
        System.out.println("zcount:"+jedis.zcount("scores",610,620));

        //zincrby:将指定值分数加分，返回加后的分数
        System.out.println("zincrby:"+jedis.zincrby("scores",10,"xx"));

        //zrange:返回指定坐标的值
        System.out.println("zrange:"+jedis.zrange("scores",0,1));

        //zrangeByScore:返回指定分数范围内的对象
        System.out.println("zrangeByScore:"+jedis.zrangeByScore("scores",600,700));

        //zrank:返回指定值的位置（分数低->高，0开始）
        System.out.println("zrank:"+jedis.zrank("scores","yy"));

        //zrevrank:返回指定值的位置（分数高->低，0开始）
        System.out.println("zrevrank:"+jedis.zrevrank("scores", "yy"));


        //zrem:删除，其中还有zremrangeByRank\zremrangeByScore
        System.out.println("zrem:"+jedis.zrem("scores", "yy"));

        jedis.zadd("scores", 630, "yy");
        jedis.zadd("scores", 640, "zz");
        //zrevrange：获取指定位置数据（分数从高->低）
        System.out.println(":"+jedis.zrevrange("scores",0,1));


        System.out.println("zrangeByScoreWithScores:"+jedis.zrangeByScoreWithScores("scores",600,700));

        //zscore:获取指定分数
        System.out.println("zscore:"+jedis.zscore("scores", "xx"));
        jedis.zadd("scores_1", 630.5, "xx");
        jedis.zadd("scores_1",610.5,"bb");
        jedis.zadd("scores_1",622.5,"cc");

        //zunionstore:sortedset集合的并集并保存,如果集合中元素相同，则分数相加
        System.out.println("zunionstore:"+jedis.zunionstore("score_total","scores","scores_1"));


        ZParams zParams = new ZParams();
        zParams.aggregate(ZParams.Aggregate.MAX);//指定分数操作：+，最小，最大
        zParams.weightsByDouble(1,0.1);//分数中的乘法因子
        System.out.println("zunionstore:"+jedis.zunionstore("score_max",zParams,"scores","scores_1"));

        //zinterstore:集合元素取交集，相同元素值相加(默认)
        System.out.println("zinterstore:"+jedis.zinterstore("score_inter","scores","scores_1"));
```

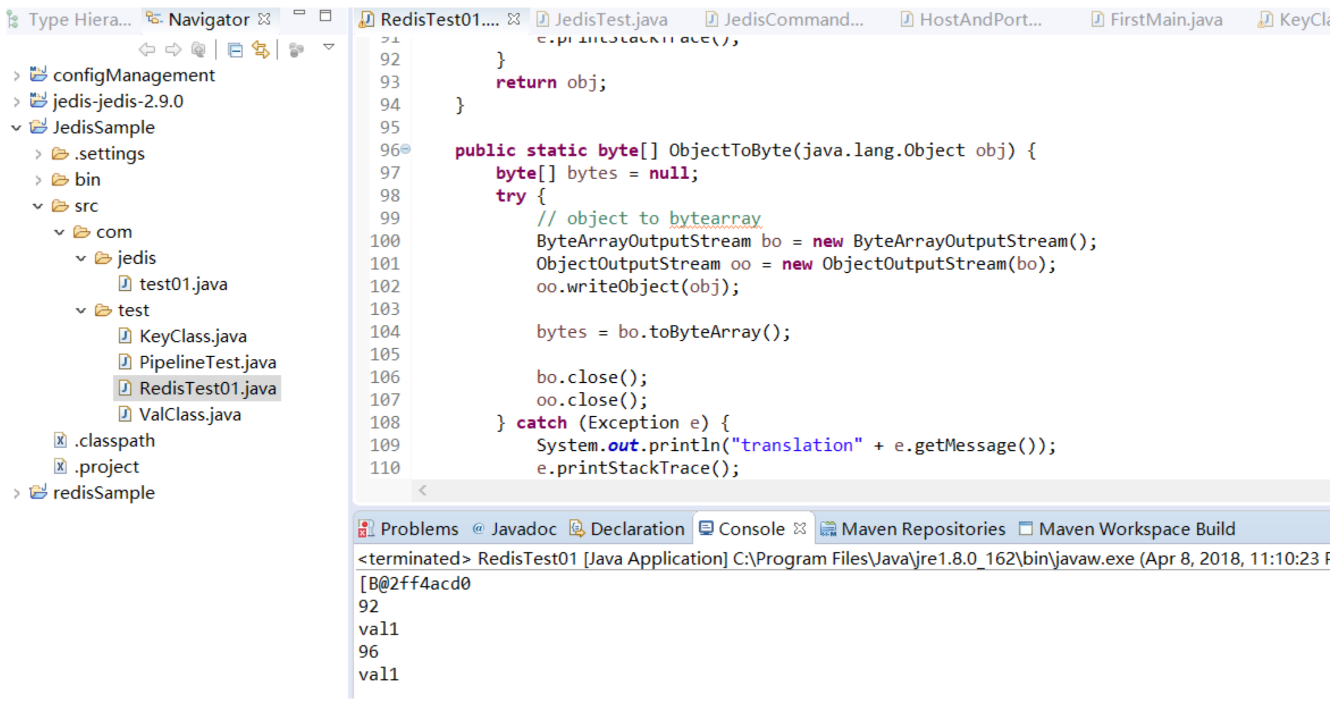
```
        System.out.println("---------------redis-SortedSet----------------\n");

    }
    public static void main(String[] args) {
        Jedis jedis = JEDIS_POOL.getResource();
        stringCmdTest(jedis);
        hashMapCmdTest(jedis);
        listCmdTest(jedis);
        setCmdTest(jedis);
        sortedSetTest(jedis);
    }
}
```

代码结果如下：



```
        System.out.println("zunionstore:"+jedis.zunionstore("score_max",zPara

        //zinterstore:集合元素取交集，相同元素值相加(默认)
        System.out.println("zinterstore:"+jedis.zinterstore("score_inter","sc


        System.out.println("---------------redis-SortedSet---------------\

    }
    public static void main(String[] args) {
        Jedis jedis = JEDIS_POOL.getResource();
        stringCmdTest(jedis);
        hashMapCmdTest(jedis);
        listCmdTest(jedis);
        setCmdTest(jedis);
        sortedSetTest(jedis);

        System.out.println("命令参考地址=>http://redisdoc.com/");
    }
}
```

```
<terminated> test01 [Java Application] C:\Program Files\Java\jre1.8.0_162\bin\javaw.exe (Apr 8, 2018,
---------------redis-String----------------
name=>wsy:OK
name:wsy
append:6
strlen:6
getrange:
setrange:11
mset:OK
mget:[wsy, 29]
incr:30
incrBy:33
decr:32
decrBy:29
setex:OK
setnx:1
del:0
---------------redis-String----------------
```



```
            e.printStackTrace();
        }
        return obj;
    }

    public static byte[] ObjectToByte(java.lang.Object obj) {
        byte[] bytes = null;
        try {
            // object to bytearray
            ByteArrayOutputStream bo = new ByteArrayOutputStream();
            ObjectOutputStream oo = new ObjectOutputStream(bo);
            oo.writeObject(obj);

            bytes = bo.toByteArray();

            bo.close();
            oo.close();
        } catch (Exception e) {
            System.out.println("translation" + e.getMessage());
            e.printStackTrace();
```

```
<terminated> RedisTest01 [Java Application] C:\Program Files\Java\jre1.8.0_162\bin\javaw.exe (Apr 8, 2018, 11:10:23 
[B@2ff4acd0
92
val1
96
val1
```

完工