1. 简介

Mockito是一个强大的用于java开发的模拟测试框架，通过Mockito可以创建和配置Mock对象，进而简化有外部依赖的类的测试。

使用流程：

* 创建外部依赖的Mock对象，然后将此Mock对象注入到测试类中
* 执行测试代码
* 校验测试代码是否执行正确

使用限制（这三种类型不能被测试）：

* final classes
* anonymous classes
* primitive types

1. 基本使用
2. **依赖：**

|  |
| --- |
| testCompile "org.mockito:mockito-core:2.8.47" |
| import static org.mockito.Mockito.\*; |

1. **创建Mock对象**

方式一：

|  |
| --- |
| mockDao = mock(PersonDao.class); |

方式二：

|  |
| --- |
| @RunWith(MockitoJUnitRunner.class)  public class MockitoTest {  @Mock  private PersonDao personDao;  ……  } |

1. **配置Mock对象**

|  |
| --- |
| when(mockDao.getPerson(1)).thenReturn(new Person(1, "Person1"));  when(mockDao.update(isA(Person.class))).thenReturn(true); |

1. **校验Mock对象的方法调用**

Mockito会追踪Mock对象的方法调用和调用方法时所传递的参数，通过Mockito.verify()方法来校验指定的方法调用是否满足断言：

|  |
| --- |
| verify(mockDao, times(1)).getPerson(eq(1));  verify(mockDao, times(1)).update(isA(Person.class)); |

1. **使用spy()部分模拟对象**

Mockito.spy()方法可以包装一个真实的java对象，并返回一个包装后的新对象，若没有特别配置，对这个新对象的所有方法调用，都会委派给实际的对象：

Spy 和 Mock的区别：

Mock生成的类，所有方法都不是真实的方法，而且返回值都是NULL；

Spy生成的类，所有方法都是真实方法，返回值都是和真实方法一样的。

方式一：

|  |
| --- |
| List realList = new ArrayList();  realList.add(1);  List spyList = spy(realList);  when(spyList.size()).thenReturn(10); |

方式二：

|  |
| --- |
| @RunWith(MockitoJUnitRunner.class)  public class MockitoTest {  @Spy  private List spyList = new ArrayList();  ……  } |

1. **@InjectMocks 在Mockito中进行依赖注入**

|  |
| --- |
| @InjectMocks 与 @Mock的区别：  @Mock创建一个Mock：    @InjectMocks创建一个实例，其余用@Mock（或@Spy）注解创建的mock将被注入到该实例中： |

1. **参数捕获**

方式一：

|  |
| --- |
| List<String> list = Arrays.asList("1", "2");  List mockList = mock(List.class);  ArgumentCaptor<List> arg = ArgumentCaptor.forClass(List.class);  mockList.addAll(list);  verify(mockList).addAll(arg.capture());  System.out.println(arg.getAllValues());  System.out.println(arg.getValue()); |

方式二：

|  |
| --- |
| @RunWith(MockitoJUnitRunner.class)  public class MockitoTest {  @Captor  private ArgumentCaptor<List> arg;  ……  } |

1. **校验方法调用顺序**

|  |
| --- |
|  |

1. **参数匹配器Matchers**

|  |
| --- |
| anyInt(), anyString(), eq(), anyCollection(), anyCollectionOf(Class<T> clazz), |

1. 对返回结果进行拦截

|  |
| --- |
|  |

或者使用thenReturn()或thenThrow()