



链滴

使用 Mockito 进行测试

作者: [seanlee](#)

原文链接: <https://ld246.com/article/1500687511046>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

关于单元测试的一些问题

当我们Javaweb项目中编写单元测试的时候，通常会面临一个普遍的问题：需要测试的类会有很多依赖，而这些依赖的类或者对又会有很多别的依赖，导致我们在写单元测试的时候几乎需要把完整的业务体系代码编写出来，而在单元测试中将这这些个依赖完整的构建出来是一件很困难的事情，通常这个时候，我们会想到把里面一需要的依赖“Mock”出来。

Mock及使用Mockito

Mock,从单词层面来讲师仿和模拟，在软件开发和测试中通常是指模拟对象。

简单地说就是对类的类所依赖的其他类和对象，进行mock - 构建它们的一个假的对象，定义这些假对象上的行为，然提供给被测试对象使用。被测试对象像使用真的对象一样使用它们。用这种方式，我们可以把测试的标限定于被测试对象本身，就如同在被测试对象周围做了一个划断，形成了一个尽量小的被测试目标

通过这样，我们可以在不写大量依赖对象的前提下使用少量的代码即可完成整个业务流程的单元测试。具体的使用可以参考下一个业务中的简单例子。

通过这样，我们可以在不写大量依赖对象的前提下使用少量的代码即可完成整个业务流程的单元测试。具体的使用可以参考下一个业务中的简单例子。

Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol'; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: start; text-indent: 0px; text-transform: none; white-space: normal; widows: 1; word-spacing: 0px; -webkit-text-stroke-width: 0px;" data-source-line="10"><strong style="border: 1px solid #ccc; padding: 5px; font-weight: 600;">Mockito的框架有很多，最为知名的一个是Mockito，这是一个开源项目，使用广泛。官网：http://site.mockito.org/。</p>

<h4 id="使用实例" style="border: 1px solid #ccc; margin-top: 24px; margin-bottom: 16px; font-weight: 600; line-height: 1.25; font-size: 14px; color: #333333; font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Helvetica, Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol'; font-style: normal; font-variant: normal; letter-spacing: normal; orphans: auto; text-align: start; text-indent: 0px; text-transform: none; white-space: normal; widows: 1; word-spacing: 0px; -webkit-text-stroke-width: 0px;" data-source-line="12">使用实例</h4>

<ul style="border: 1px solid #ccc; margin-top: 0px; margin-bottom: 16px; padding-left: 2em; color: #333333; font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Helvetica, Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol'; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 2px; orphans: auto; text-align: start; text-indent: 0px; text-transform: none; white-space: normal; widows: 1; word-spacing: 0px; -webkit-text-stroke-width: 0px;" data-source-line="14">

<li style="border: 1px solid #ccc;">创建项目时引入单元测试包和mockito的包

<pre data-source-line="15"><code class="hljs" style="border: 1px solid #ccc; display: inline-block; overflow: visible; padding: 0px; color: #333333; font-family: Consolas, 'Liberation Mono', Menlo, Courier, monospace; margin: 0px; font-size: 11.9px; border-radius: 3px; word-break: normal; white-space: pre; border: 0px; line-height: inherit; word-wrap: normal; background: 0px 0px transparent;"><dependency>

<groupId>>junit</groupId>

<artifactId>>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

```

<span class="hljs-tag" style="box-sizing: border-box; color: navy; font-weight: 400;">&lt;
pan class="hljs-title" style="box-sizing: border-box; color: navy; font-weight: 400;">depende
cy</span>&gt;</span>
  <span class="hljs-tag" style="box-sizing: border-box; color: navy; font-weight: 400;">&lt;
<span class="hljs-title" style="box-sizing: border-box; color: navy; font-weight: 400;">group1
</span>&gt;</span>org.mockito<span class="hljs-tag" style="box-sizing: border-box; color:
navy; font-weight: 400;">&lt;/<span class="hljs-title" style="box-sizing: border-box; color: na
y; font-weight: 400;">group1d</span>&gt;</span>
  <span class="hljs-tag" style="box-sizing: border-box; color: navy; font-weight: 400;">&lt;
<span class="hljs-title" style="box-sizing: border-box; color: navy; font-weight: 400;">artifac
d</span>&gt;</span>mockito-all<span class="hljs-tag" style="box-sizing: border-box; color
navy; font-weight: 400;">&lt;/<span class="hljs-title" style="box-sizing: border-box; color: n
vy; font-weight: 400;">artifact1d</span>&gt;</span>
  <span class="hljs-tag" style="box-sizing: border-box; color: navy; font-weight: 400;">&lt;
<span class="hljs-title" style="box-sizing: border-box; color: navy; font-weight: 400;">versio
</span>&gt;</span>2.0.2-beta<span class="hljs-tag" style="box-sizing: border-box; color: n
vy; font-weight: 400;">&lt;/<span class="hljs-title" style="box-sizing: border-box; color: navy;
ont-weight: 400;">version</span>&gt;</span>
  <span class="hljs-tag" style="box-sizing: border-box; color: navy; font-weight: 400;">&lt;
<span class="hljs-title" style="box-sizing: border-box; color: navy; font-weight: 400;">scope<
span>&gt;</span>test<span class="hljs-tag" style="box-sizing: border-box; color: navy; font
weight: 400;">&lt;/<span class="hljs-title" style="box-sizing: border-box; color: navy; font-we
ght: 400;">scope</span>&gt;</span>
  <span class="hljs-tag" style="box-sizing: border-box; color: navy; font-weight: 400;">&lt;/
span class="hljs-title" style="box-sizing: border-box; color: navy; font-weight: 400;">depende
cy</span>&gt;</span></code></pre>

```

```

<ul style="box-sizing: border-box; margin-top: 0px; margin-bottom: 16px; padding-left: 2em;
color: #333333; font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Helvetic
, Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol'; font-size: 14px; fon
-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 2
px; orphans: auto; text-align: start; text-indent: 0px; text-transform: none; white-space: normal
widows: 1; word-spacing: 0px; -webkit-text-stroke-width: 0px;" data-source-line="31">
<li style="box-sizing: border-box;">新建一个账户对象类 Account</li>
</ul>

```

```

<pre data-source-line="32"><code class="hljs" style="box-sizing: border-box; display: inline;
overflow: visible; padding: 0px; color: #333333; font-family: Consolas, 'Liberation Mono', Men
o, Courier, monospace; margin: 0px; font-size: 11.9px; border-radius: 3px; word-break: normal;
white-space: pre; border: 0px; line-height: inherit; word-wrap: normal; background: 0px 0px tr
nsparent;">public class Account {
  /*
  * 账户ID
  */
  private String accountId;
  /**
  * 收支
  */
  private long balance;

  /**
  * 初始化账户ID和收支
  * @param accountId
  * @param initialBalance

```

```

*/
public Account(String accountId, long initialBalance){
    this.accountId = accountId;
    this.balance = initialBalance;
}

public void debit(long amount){
    this.balance -= amount;
}

public void credit(long amount){
    this.balance += amount;
}

public long getBalance(){
    return this.balance;
}
}

```

- 创建一个账号管理接口，里面包含了一个查找账户对象的方法
一个更新账户信息的方法

```


```

/**
 * 根据userId找到对应账号对象
 * @param userId
 * @return
 */
Account findAccountForUser(String userId);

/**
 * 更新账户信息
 * @param account
 */
void updateAccount(Account account);
}

```


```

- 创建一个账号管理接口，里面包含了一个查找账户对象的方法
一个更新账户信息的方法


```
widows: 1; word-spacing: 0px; -webkit-text-stroke-width: 0px;" data-source-line="89">
<li style="box-sizing: border-box;">创建账号管理接口的实现类MockAccountManager，并在其
新增一个账号管理的方法</li>
```

```
</ul>
```

```
<pre data-source-line="90"><code class="hljs" style="box-sizing: border-box; display: inline;
overflow: visible; padding: 0px; color: #333333; font-family: Consolas, 'Liberation Mono', Men
o, Courier, monospace; margin: 0px; font-size: 11.9px; border-radius: 3px; word-break: normal;
white-space: pre; border: 0px; line-height: inherit; word-wrap: normal; background: 0px 0px tr
ansparent;"><span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-
eight: bold;">public</span> <span class="hljs-class" style="box-sizing: border-box;"><span
lass="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">class
/span> <span class="hljs-title" style="box-sizing: border-box; color: #445588; font-weight: bo
d;">MockAccountManager</span> <span class="hljs-keyword" style="box-sizing: border-bo
; color: #333333; font-weight: bold;">implements</span> <span class="hljs-title" style="box
sizing: border-box; color: #445588; font-weight: bold;">AccountManager</span> </span>{
    <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bo
d;">private</span> Hashtable accounts = <span class="hljs-keyword" style="box-sizing: bor
er-box; color: #333333; font-weight: bold;">new</span> Hashtable();
```

```
<span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic;"
/**
```

```
* 新增一个添加账户的方法
```

```
* <span class="hljs-doctag" style="box-sizing: border-box; color: #dd1144;">@param</spa
> userId
```

```
* <span class="hljs-doctag" style="box-sizing: border-box; color: #dd1144;">@param</spa
> account
```

```
*/</span>
```

```
<span class="hljs-function" style="box-sizing: border-box;"><span class="hljs-keyword" styl
="box-sizing: border-box; color: #333333; font-weight: bold;">public</span> <span class="hl
s-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">void</span>
span class="hljs-title" style="box-sizing: border-box; color: #990000; font-weight: bold;">ad
Account</span><span class="hljs-params" style="box-sizing: border-box;">(String userId, A
count account)</span> </span>{
```

```
    <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bo
d;">this</span>.accounts.put(userId, account);
```

```
}
```

```
<span class="hljs-function" style="box-sizing: border-box;"><span class="hljs-keyword" styl
="box-sizing: border-box; color: #333333; font-weight: bold;">public</span> Account <span
class="hljs-title" style="box-sizing: border-box; color: #990000; font-weight: bold;">findAcco
untForUser</span><span class="hljs-params" style="box-sizing: border-box;">(String userId)
/span> </span>{
```

```
    <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bo
d;">return</span> (Account) <span class="hljs-keyword" style="box-sizing: border-box; colo
: #333333; font-weight: bold;">this</span>.accounts.get(userId);
```

```
}
```

```
<span class="hljs-function" style="box-sizing: border-box;"><span class="hljs-keyword" styl
="box-sizing: border-box; color: #333333; font-weight: bold;">public</span> <span class="hl
s-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">void</span>
span class="hljs-title" style="box-sizing: border-box; color: #990000; font-weight: bold;">upd
teAccount</span><span class="hljs-params" style="box-sizing: border-box;">(Account acco
nt)</span> </span>{
```

```
<span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic"
">// do nothing</span>
}
}</code></pre>
```

```
<ul style="box-sizing: border-box; margin-top: 0px; margin-bottom: 16px; padding-left: 2em;
color: #333333; font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Helvetic
, Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol'; font-size: 14px; fon
-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 2
px; orphans: auto; text-align: start; text-indent: 0px; text-transform: none; white-space: normal
widows: 1; word-spacing: 0px; -webkit-text-stroke-width: 0px;" data-source-line="114">
<li style="box-sizing: border-box;">创建一个服务类AccountService，通过该类来调用接口中的
法</li>
</ul>
```

```
<pre data-source-line="115"><code class="hljs" style="box-sizing: border-box; display: inline
overflow: visible; padding: 0px; color: #333333; font-family: Consolas, 'Liberation Mono', Men
o, Courier, monospace; margin: 0px; font-size: 11.9px; border-radius: 3px; word-break: normal;
white-space: pre; border: 0px; line-height: inherit; word-wrap: normal; background: 0px 0px tr
nsparent;"><span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-
eight: bold;">public</span> <span class="hljs-class" style="box-sizing: border-box;"><span
lass="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">class
/span> <span class="hljs-title" style="box-sizing: border-box; color: #445588; font-weight: bo
d;">AccountService</span> </span>{
  <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bo
d;">private</span> AccountManager accountManager;
  <span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic;"
// private MockAccountManager mockAccountManager;</span>
```

```
<span class="hljs-function" style="box-sizing: border-box;"><span class="hljs-keyword" styl
="box-sizing: border-box; color: #333333; font-weight: bold;">public</span> <span class="hl
s-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">void</span>
span class="hljs-title" style="box-sizing: border-box; color: #990000; font-weight: bold;">set
ccountManager</span> <span class="hljs-params" style="box-sizing: border-box;">(MockAc
ountManager manager)</span> </span>{
  accountManager = manager;
}
```

```
<span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic;"
/**
 *
 * <span class="hljs-doctag" style="box-sizing: border-box; color: #dd1144;">@param</spa
> senderId 转账人的ID
 * <span class="hljs-doctag" style="box-sizing: border-box; color: #dd1144;">@param</spa
> beneficiaryId 收款人ID
 * <span class="hljs-doctag" style="box-sizing: border-box; color: #dd1144;">@param</spa
> amount 转账金额
*/</span>
<span class="hljs-function" style="box-sizing: border-box;"><span class="hljs-keyword" styl
="box-sizing: border-box; color: #333333; font-weight: bold;">public</span> <span class="hl
s-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">void</span>
span class="hljs-title" style="box-sizing: border-box; color: #990000; font-weight: bold;">tran
fer</span> <span class="hljs-params" style="box-sizing: border-box;">(String senderId, Strin
```

```

beneficiaryId, <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">long</span> amount)</span> </span>{
    Account sender = accountManager.findAccountForUser(senderId);
    Account beneficiary = accountManager.findAccountForUser(beneficiaryId);

    <span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic"
">//转账账户减少</span>
    sender.debit(amount);
    <span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic"
">//收款账户增加</span>
    beneficiary.credit(amount);
    <span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic"
">//更新转账账户信息</span>
    <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">this</span>.accountManager.updateAccount(sender);
    <span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic"
">//更新收款账户信息</span>
    <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">this</span>.accountManager.updateAccount(beneficiary);
}
}
</code> </pre>

```

- 创建一个普通的单元测试类来测试这个业务流程

```

<pre data-source-line="147"><code class="hljs" style="box-sizing: border-box; display: inline-block; overflow: visible; padding: 0px; color: #333333; font-family: Consolas, 'Liberation Mono', Meno, Courier, monospace; margin: 0px; font-size: 11.9px; border-radius: 3px; word-break: normal; white-space: pre; border: 0px; line-height: inherit; word-wrap: normal; background: 0px 0px transparent;"><span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">import</span> org.junit.<span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">Test</span>;
<span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">import</span> junit.framework.<span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">TestCase</span>;

```

```

public <span class="hljs-class" style="box-sizing: border-box;"><span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">class</span> <span class="hljs-title" style="box-sizing: border-box; color: #445588; font-weight: bold;">TestAccountService</span> <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">extends</span> <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">extends</span> <span class="hljs-title" style="box-sizing: border-box; color: #445588; font-weight: bold;">TestCase</span>{</span>

```

```

<span class="hljs-annotation" style="box-sizing: border-box;">@Test</span>
public void testTransferOk(){

```



```

    <span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic"
">使用MockAccountManager</span>
    <span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;"
MockAccountManager</span> mockAccountManager = <span class="hljs-keyword" style="
ox-sizing: border-box; color: #333333; font-weight: bold;">new</span> <span class="hljs-ty
e" style="box-sizing: border-box; color: #445588; font-weight: bold;">MockAccountManager
/span>();

    <span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic"
">初始化两个账户</span>
    <span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;"
Account</span> senderAccount = <span class="hljs-keyword" style="box-sizing: border-box;
color: #333333; font-weight: bold;">new</span> <span class="hljs-type" style="box-sizing:
order-box; color: #445588; font-weight: bold;">Account</span>(<span class="hljs-string" sty
e="box-sizing: border-box; color: #dd1144;">"1"</span>, <span class="hljs-number" style="
ox-sizing: border-box; color: teal;">200</span>);
    <span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;"
Account</span> beneficiaryAccount = <span class="hljs-keyword" style="box-sizing: border
box; color: #333333; font-weight: bold;">new</span> <span class="hljs-type" style="box-siz
ng: border-box; color: #445588; font-weight: bold;">Account</span>(<span class="hljs-strin
" style="box-sizing: border-box; color: #dd1144;">"2"</span>, <span class="hljs-number" sty
e="box-sizing: border-box; color: teal;">100</span>);
    mockAccountManager.addAccount(<span class="hljs-string" style="box-sizing: border-box
color: #dd1144;">"1"</span>, senderAccount);
    mockAccountManager.addAccount(<span class="hljs-string" style="box-sizing: border-box
color: #dd1144;">"2"</span>, beneficiaryAccount);

    <span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic"
">初始化AccountService, 将MockManager对象传入</span>
    <span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;"
AccountService</span> accountService = <span class="hljs-keyword" style="box-sizing: bor
er-box; color: #333333; font-weight: bold;">new</span> <span class="hljs-type" style="box-
izing: border-box; color: #445588; font-weight: bold;">AccountService</span>();
    accountService.setAccountManager(mockAccountManager);

    <span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic"
">转帐操作</span>
    accountService.transfer(<span class="hljs-string" style="box-sizing: border-box; color: #dd
144;">"1"</span>, <span class="hljs-string" style="box-sizing: border-box; color: #dd1144;"
"2"</span>, <span class="hljs-number" style="box-sizing: border-box; color: teal;">50</spa
>);

    <span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic"
">验证</span>
    assertEquals(<span class="hljs-number" style="box-sizing: border-box; color: teal;">150</
pan>, senderAccount.getBalance());
    assertEquals(<span class="hljs-number" style="box-sizing: border-box; color: teal;">150</
pan>, beneficiaryAccount.getBalance());
}
} </code> </pre>

```

<ul style="box-sizing: border-box; margin-top: 0px; margin-bottom: 16px; padding-left: 2em; color: #333333; font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Helvetic

```
, Arial, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol'; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 2px; orphans: auto; text-align: start; text-indent: 0px; text-transform: none; white-space: normal; widows: 1; word-spacing: 0px; -webkit-text-stroke-width: 0px;" data-source-line="181">
<li style="box-sizing: border-box;">创建一个用Mockito进行测试的类</li>
</ul>
```

```
<pre data-source-line="182"><code class="hljs" style="box-sizing: border-box; display: inline overflow: visible; padding: 0px; color: #333333; font-family: Consolas, 'Liberation Mono', Meno, Courier, monospace; margin: 0px; font-size: 11.9px; border-radius: 3px; word-break: normal; white-space: pre; border: 0px; line-height: inherit; word-wrap: normal; background: 0px 0px transparent;">public <span class="hljs-class" style="box-sizing: border-box;"><span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">class</span> <span class="hljs-title" style="box-sizing: border-box; color: #445588; font-weight: bold;">TestMockAccountService</span> <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">extends</span></span> <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">extends</span></span> <span class="hljs-title" style="box-sizing: border-box; color: #445588; font-weight: bold;">TestCase</span> {</span>
```

```
<span class="hljs-annotation" style="box-sizing: border-box;">@Test</span>
public void testTransferOk() {
    <span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">Account</span> senderAccount = <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">new</span> <span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">Account</span>(<span class="hljs-string" style="box-sizing: border-box; color: #dd1144;">"1"</span>, <span class="hljs-number" style="box-sizing: border-box; color: teal;">200</span>);
    <span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">Account</span> beneficiaryAccount = <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">new</span> <span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">Account</span>(<span class="hljs-string" style="box-sizing: border-box; color: #dd1144;">"2"</span>, <span class="hljs-number" style="box-sizing: border-box; color: teal;">100</span>);
```

```
<span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic">>// 使用mockito创建一个模拟对象</span>
```

```
<span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">MockAccountManager</span> mockAccountManager = <span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">Mockito</span>.mock(<span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">MockAccountManager</span>.<span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">class</span>);
```

```
<span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic">>//mockito自带的控制流写法</span>
```

```
when(mockAccountManager.findAccountForUser(<span class="hljs-string" style="box-sizing: border-box; color: #dd1144;">"1"</span>)).thenReturn(senderAccount);
when(mockAccountManager.findAccountForUser(<span class="hljs-string" style="box-sizing: border-box; color: #dd1144;">"2"</span>)).thenReturn(beneficiaryAccount);
```

```
<span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">AccountService</span> accountService = <span class="hljs-keyword" style="box-sizing: border-box; color: #333333; font-weight: bold;">new</span> <span class="hljs-type" style="box-sizing: border-box; color: #445588; font-weight: bold;">AccountService</span>();
```

```
accountService.setAccountManager(mockAccountManager);
```

```
<span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic" > // 转帐操作 </span>  
    accountService.transfer(<span class="hljs-string" style="box-sizing: border-box; color: #dd1144;" > "1" </span>, <span class="hljs-string" style="box-sizing: border-box; color: #dd1144;" > "2" </span>, <span class="hljs-number" style="box-sizing: border-box; color: teal;" > 50 </span> </pre>
```

```
<span class="hljs-comment" style="box-sizing: border-box; color: #999988; font-style: italic" > // 验证 </span>  
    assertEquals(<span class="hljs-number" style="box-sizing: border-box; color: teal;" > 150 </span>, senderAccount.getBalance());  
    assertEquals(<span class="hljs-number" style="box-sizing: border-box; color: teal;" > 150 </span>, beneficiaryAccount.getBalance());  
}
```

```
</code><br /><br /><br /><br />  
上。 </pre>
```