

Ch2 Code Unit Testing

Write Code to Test Code(2)



Instructor: **Haiying SUN**

E-mail: hysun@sei.ecnu.edu.cn

Office: **ECNU Science Build B1104**

Available Time: **Wednesday 8:00 -12:00 a.m.**

Agenda



- Introduction to Unit Testing
- Common Code Defect Categories
- Unit Tests Design Heuristic Rules
- **Unit Tests Implementation**
 - Junit & Mockito & Qualified test scripts
- Code Test Adequacy Criteria
 - Control flow based & Jacoco
 - Data flow based
 - Mutation Based
- Code Test Generation

Readability

*Good programmers write code that humans
can understand*



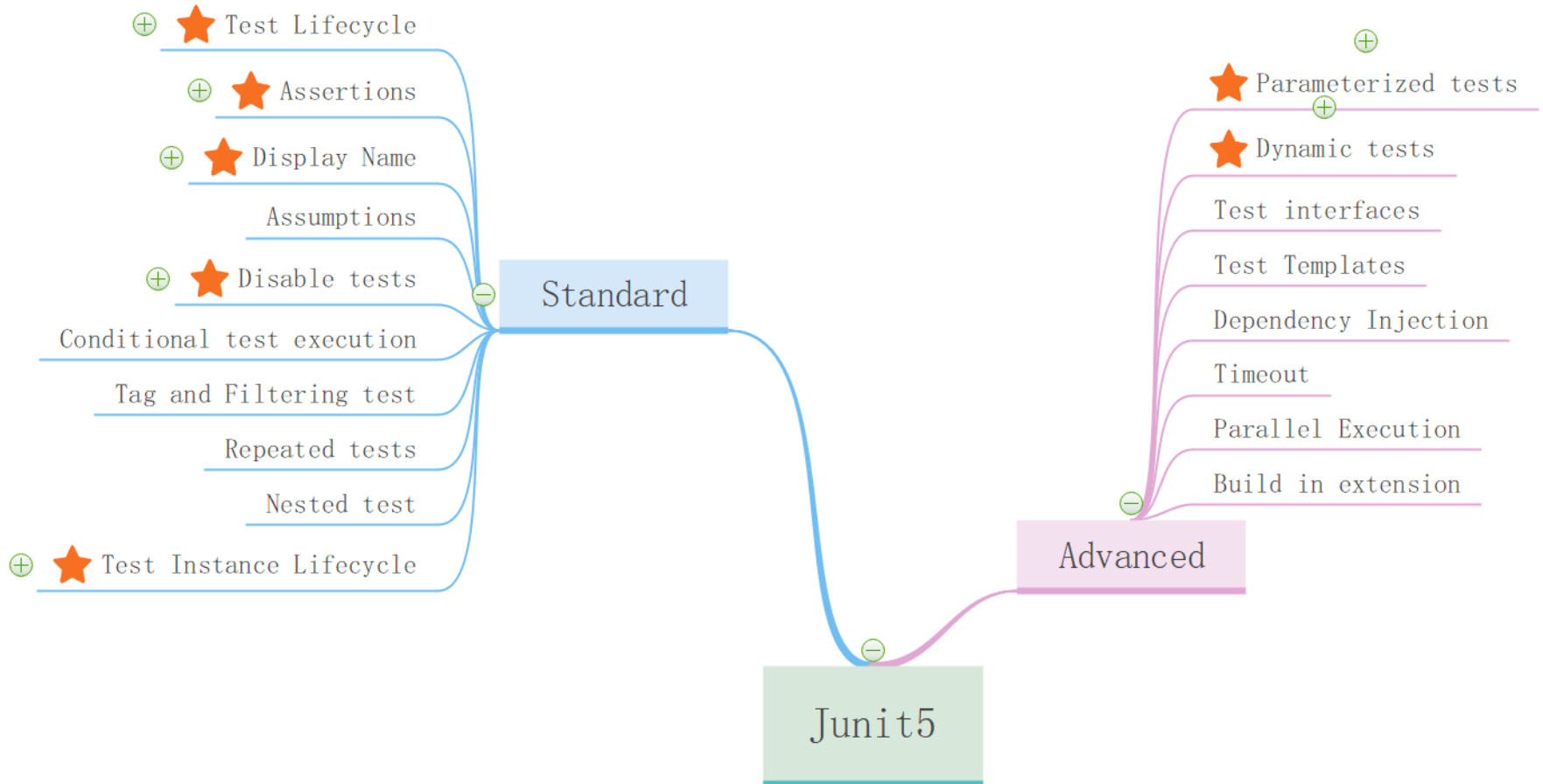
Martin Fowler

What is JUnit

JUnit is a simple, open source **framework** to **write** and **run repeated tests**

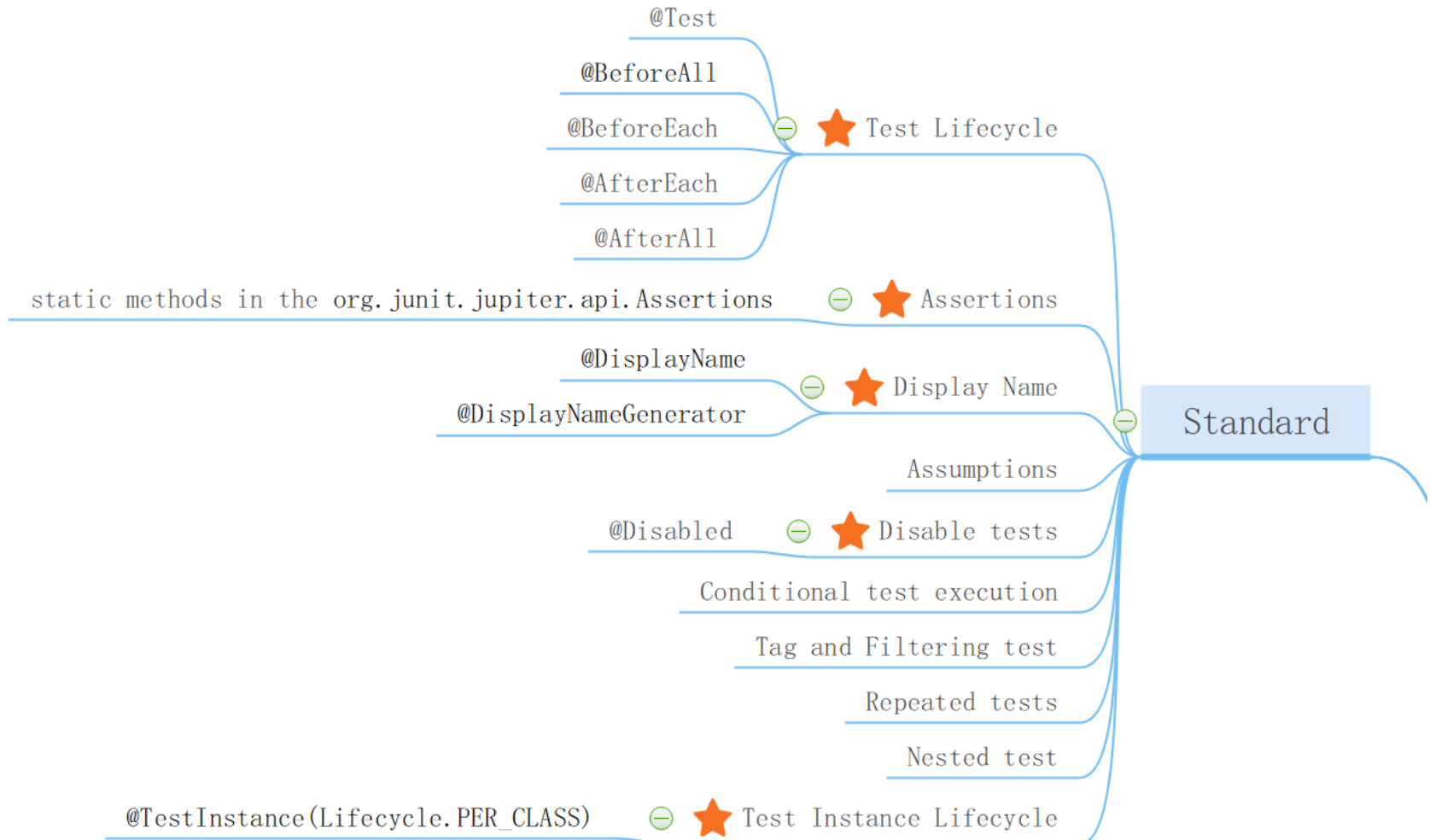
-- Org Junit

Junit5 Features

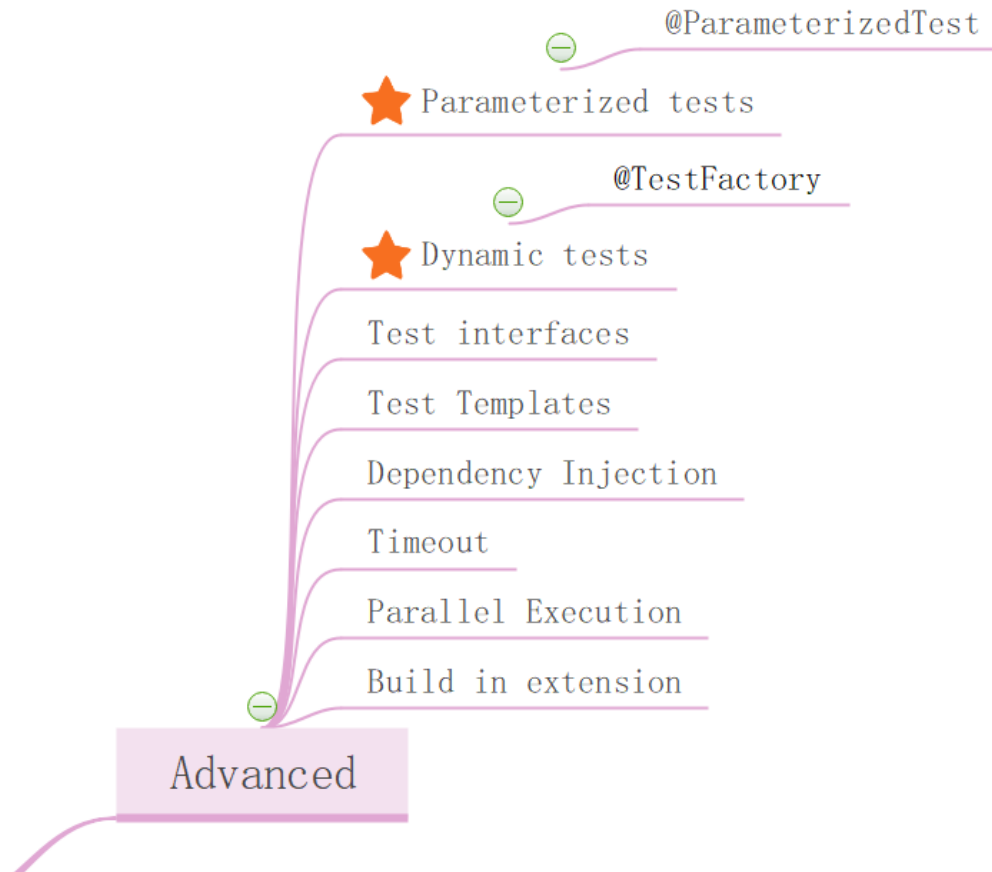


<https://junit.org/junit5/docs/current/user-guide/>

Junit5 Features

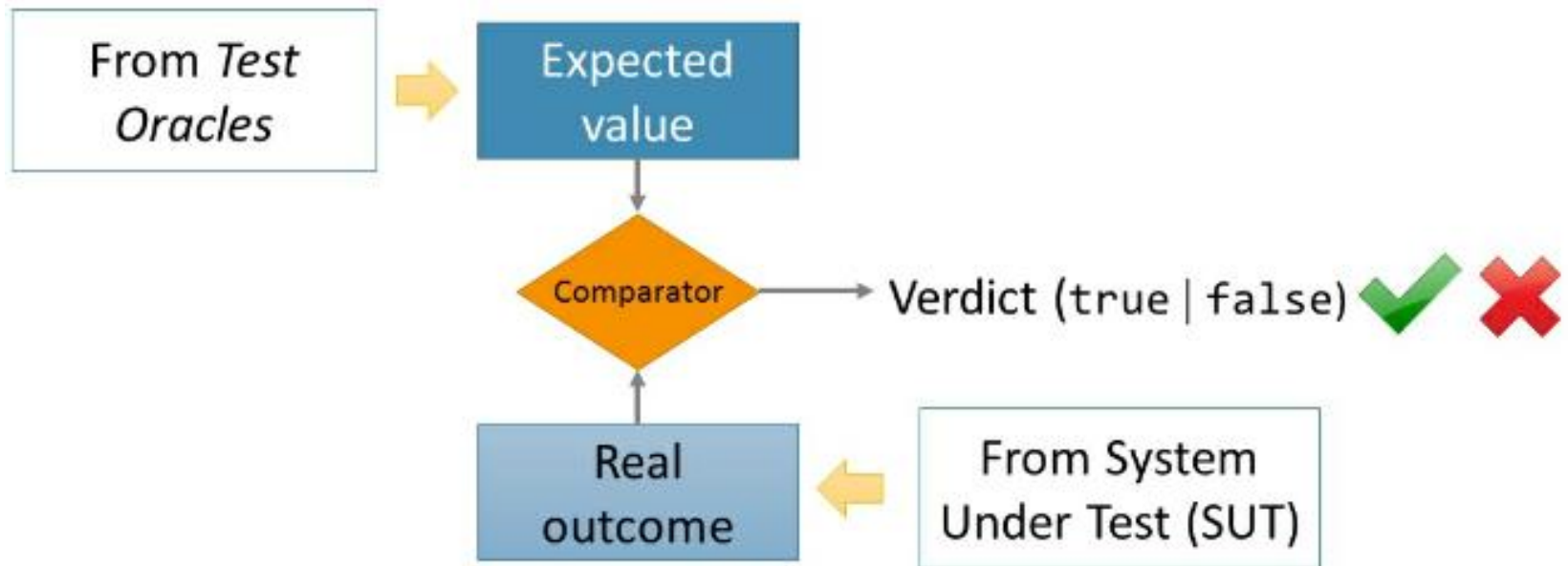


Junit5 Features

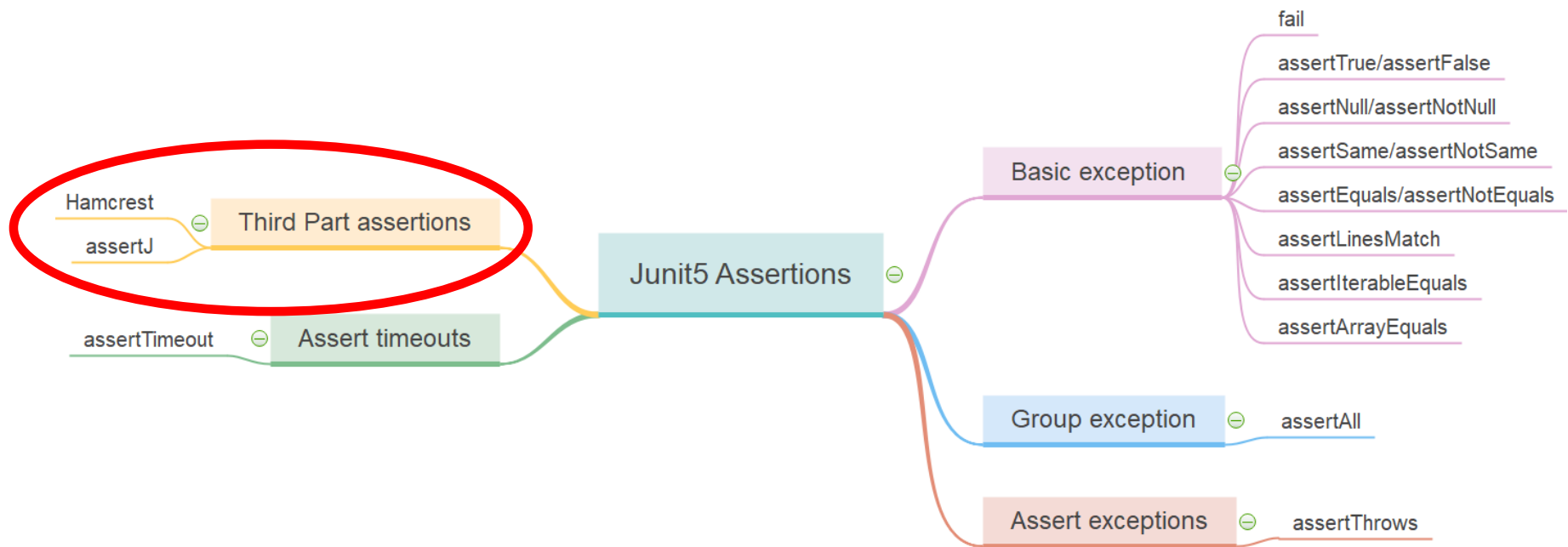


Assertions

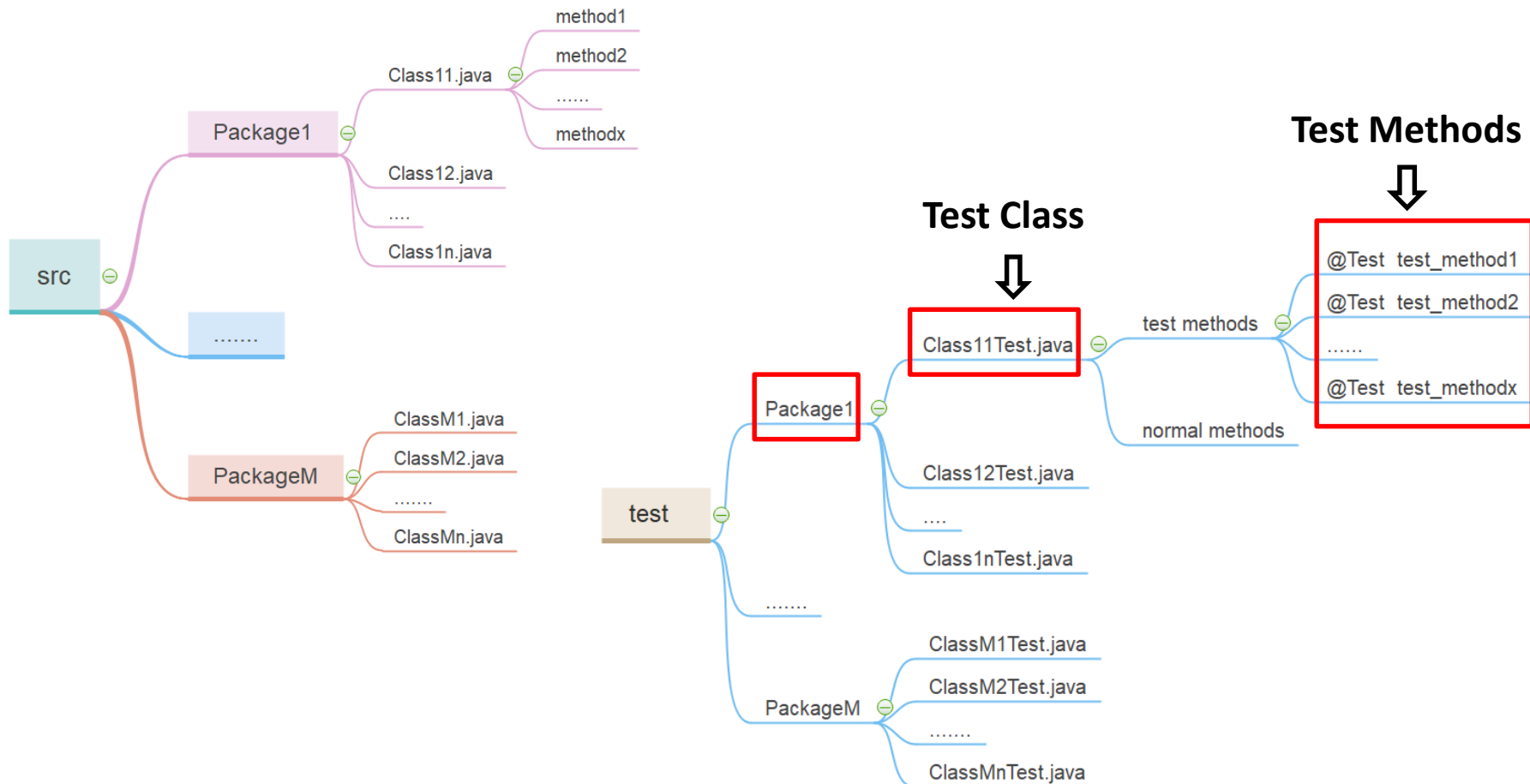
- An assertion (predicate) is a **boolean statement** typically used to reason about software correctness.



JUnit5 Assertions



JUnit Test Code Structure



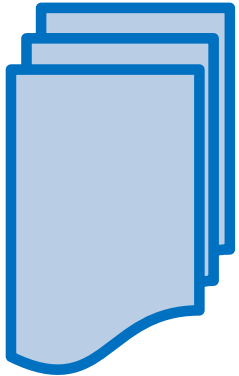
JUnit Test Code Structure

- Test Class
 - 命名方式：被测类名 + Test ， 例， MeetCalendarTest
- Test Method
 - 真正执行测试发现缺陷的地方
 - 使用@Test, @RepeatedTest, @ParameterizedTest, @TestFactory, @TestTemplate 标注的方法
 - 3A Pattern
 1. Arrange: preparation for the coming test
 2. Action: execute code under test
 3. Assertion: compare actual outputs with expected ones

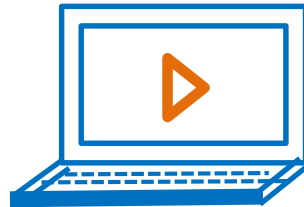
Test Lifecycle

- A test composed of 4 stages:
 1. **Setup (optional)**: the test initializes the test fixture
 2. **Exercise**: the test interacts with the SUT, getting some outcome from it as a result.
 3. **Verify**: the outcome from the system under test is compared to the expected value using one or several assertions (also known as predicates). As a result, a test verdict is created.
 4. **Teardown (optional)**: the test releases the test fixture to put the SUT back into the initial state.

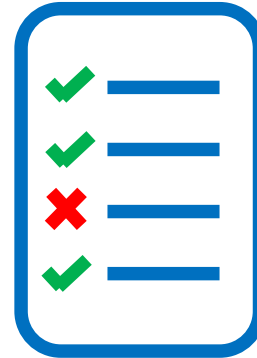
Test Lifecycle



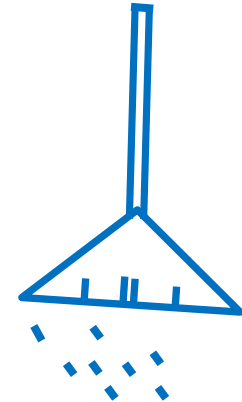
Setup:
preparation for
the coming test



Exercise:
execute code
under test



Verify:
compare actual
outputs with
expected ones

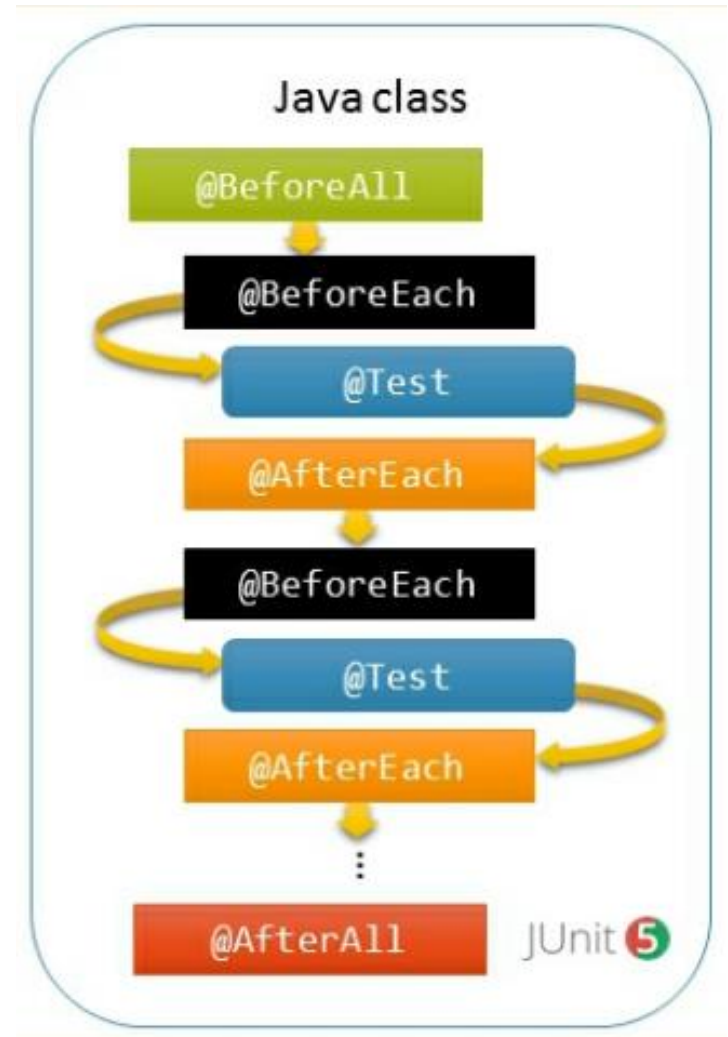


tearDown:
clean up test
environment

@BeforeAll → **@BeforeEach** → **@Test** → **@AfterEach** → **@BeforeEach** → **@Test** → **@AfterEach**
→ **@AfterAll**

Test Lifecycle

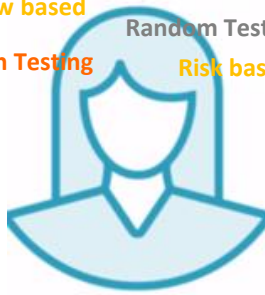
- **@BeforeAll (static method)**: executed before all @Test
- **@BeforeEach**: executed before each @Test
- **@AfterEach**: executed after each @Test
- **@AfterAll (static method)**: executed after all @Test



Deal With Repeated Test

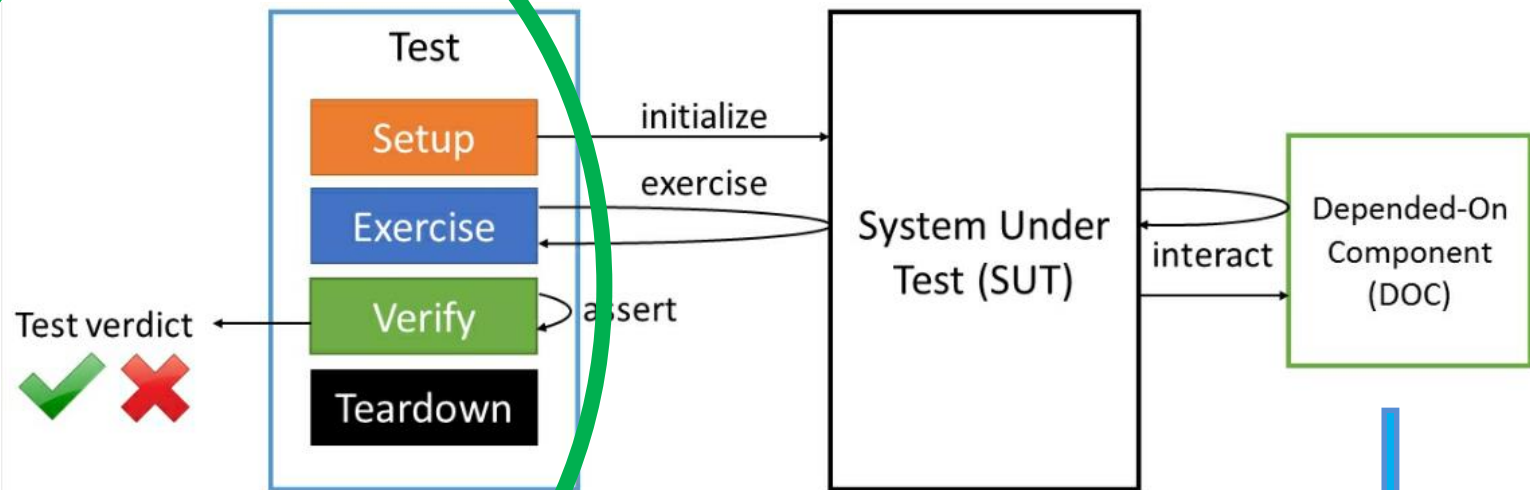
- Data Driven Test
 - Separated test data from test code
 - @ParameterizedTest, Example Code: JUnitDemo.zip
- 操作练习
 1. 课程网站下载[MeetHereMaven.zip](#): 视频资料 JUnit + mockito视频
 2. 解压[MeetHereMaven.zip](#)到当前目录（**目录中不要有中文**）
 3. Idea中打开[MeetHereMaven](#)，等待Maven更新完毕
 4. 查看test目录下SitePriceTest代码
 5. 运行SitePriceTest
 6. 查看运行测试的个数及测试执行情况

Control Flow based
Data Flow based
Mutation Testing
Logical Coverage
Random Testing
Risk based Testing



Designed Test Cases

Unit test generic structure



5 JUnit 5



mockito

Practice

黄历说，今天不宜敲代码



- 实验准备

1. 课程网站下载**MockitoDemo.zip**： 视频资料→ JUnit + mockito视频
2. 解压**MockitoDemo.zip**到当前目录（**目录中不要有中文**）
3. Idea中打开**MockitoDemo.zip**注意**JDK版本**）

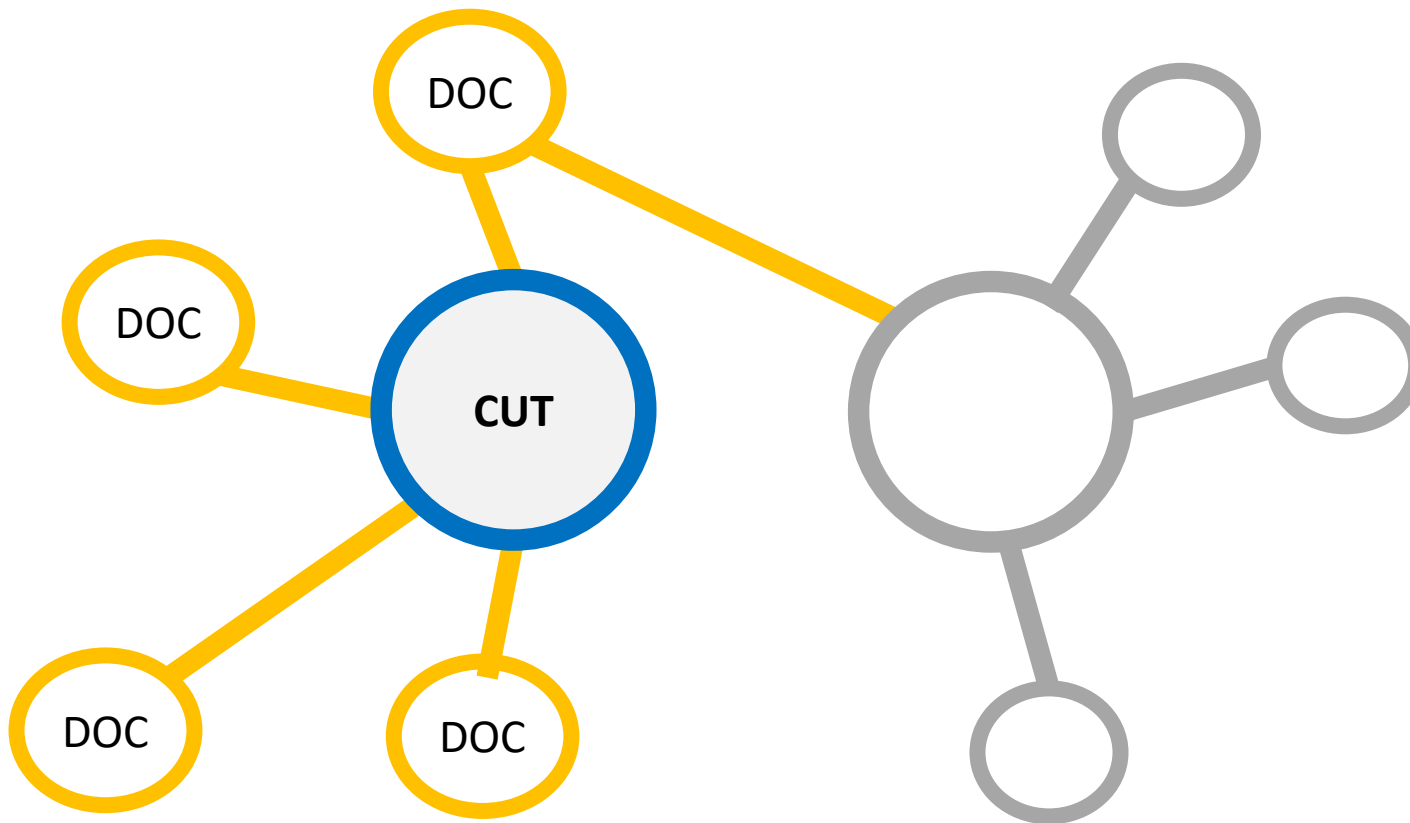
- 实验任务

- 阅读PhoneBookDAO.java， PhoneBookH2DAO.java, PhoneEntry.java
- 编写测试用例测试PhoneBookH2DAO的create（）

Difficulties

DOC Dependence On Component

CUT Code Under Test



Difficulties

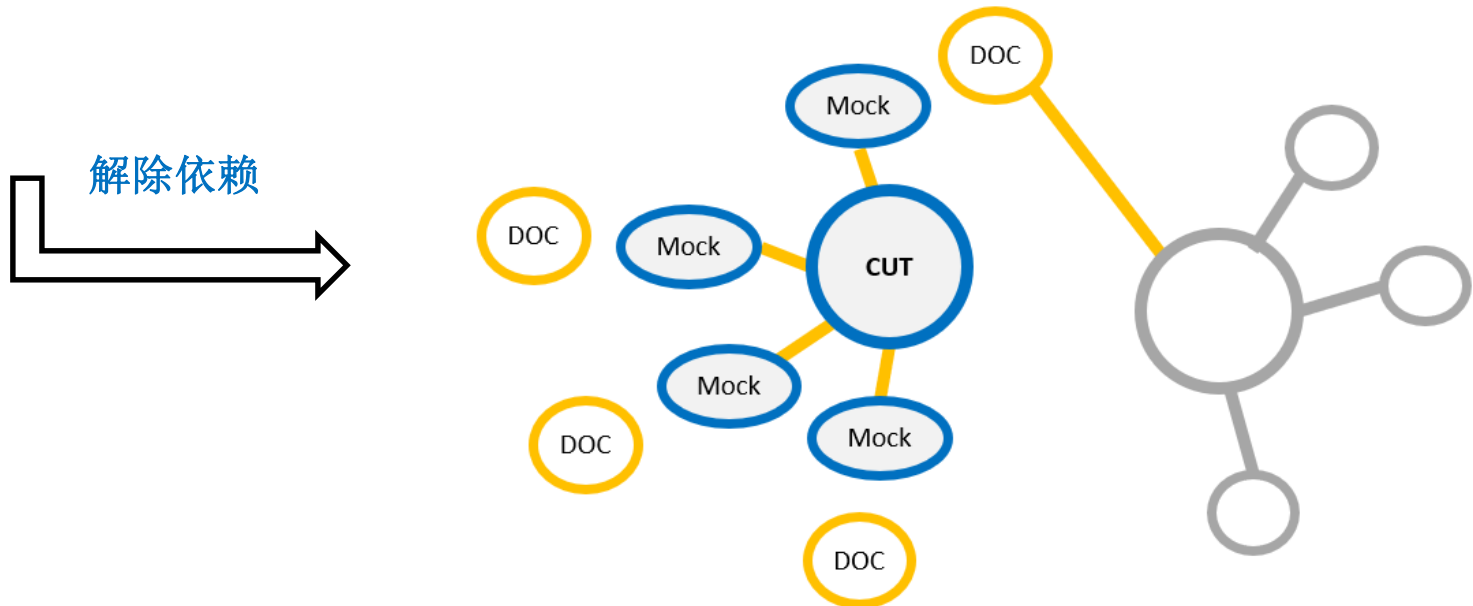
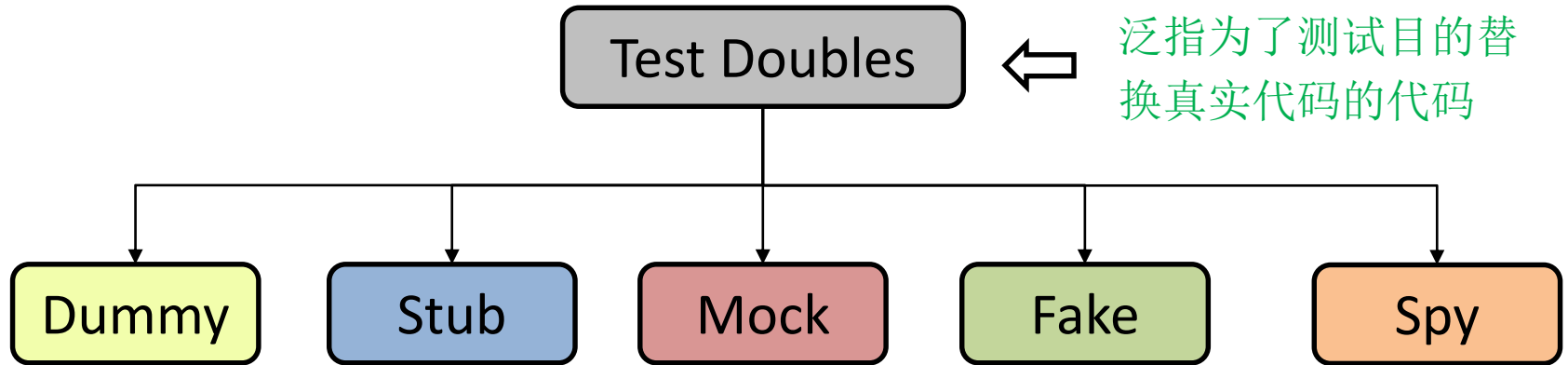
测试断言写什么？



测试断言写什么？

测试断言写什么？

Test Doubles



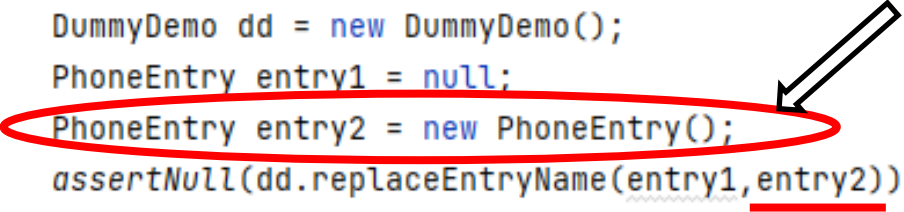
Test Doubles

- Dummy

- 在测试方法中**不使用其任何方法**的测试替身
- 一般出现在方法的参数处
- 通常为了防止NullPointerException的出现，保证测试方法可以顺利执行

```
class DummyDemoTest {  
    @Test  
    void s1_is_null_return_null() {  
        DummyDemo dd = new DummyDemo();  
        PhoneEntry entry1 = null;  
        PhoneEntry entry2 = new PhoneEntry();  
        assertNull(dd.replaceEntryName(entry1, entry2));  
    }  
}
```

A dummy



MockitoDemo项目DummyDemoTest示例

Test Double

- Fake

- 一种简化真实代码的测试替身，通常采用继承被Fake对象（真实代码），成为其子类的方法实现
- 不能作为产品代码，单纯为了测试快，不在测试中出现耗时行为
- Example: In-memory database

```
class PhoneBookH2DaoFake extends PhoneBookH2Dao {  
    protected void loadDriver() {  
    }  
  
    protected Connection getConnection() throws SQLException {  
        return connection;  
    }  
}
```

Test Doubles

- Stub
 - Provide canned answer: 为其调用者提供测试过程中需要使用
 - 通常应用响应待测系统的请求，然后返回特定的值

```
// You can mock concrete classes and interfaces
TrainSeats seats = mock(TrainSeats.class);

// stubbing appears before the actual execution
when(seats.book(Seat.near(WINDOW).in(FIRST_CLASS))).thenReturn(BOOKED);
```

Mockito中的打桩代码

Test Doubles

- spy
 - Spy are stubs that also record some information based on how they were called
 1. 使用真实代码的测试替身，返回其真实值
 2. 可以打桩
 3. 可以记录使用轨迹，便于在后续测试活动中验证是不是安排的事情按照期望发生


```
1 package edu.ecnu.sei.mockito.trading;
2
3 public class DemoClass {
4
5     public String foo() {
6
7         return "I like mock";
8     }
9 }
10
```

```
1 package edu.ecnu.sei.mockito.trading;
2
3 import static org.mockito.Mockito.spy;
4
5 import org.junit.Test;
6
7 public class FirstSpy {
8
9     DemoClass demo = spy(DemoClass.class);
10
11     @Test
12     public void what_is_a_mock() {
13
14         System.out.println(demo.foo());
15     }
16 }
```

Output: I like mock

```
1 package edu.ecnu.sei.mockito.trading;
2
3 public class DemoClass {
4
5     public String foo() {
6
7         return "I like mock";
8
9     }
10 }
```

```
1 package edu.ecnu.sei.mockito.trading;
2
3 import static org.mockito.Mockito.mock;
4
5 import org.junit.Test;
6
7 public class FirstMock {
8
9     DemoClass demo = mock(DemoClass.class);
10
11     @Test
12     public void what_is_a_mock() {
13
14         System.out.println(demo.foo());
15     }
16 }
```

Output: null

Test Double

- Mock

- 按照期望实现的用于测试方法中的行为代码
 1. 正常路径：返回正常值
 2. 异常路径：返回期望的错误/异常

- Mockito

- mock被测对象外部依赖的Java开源测试框架



Two Approaches of Test Verdict Construction

- State based testing (**Test by Result**)
 - determine whether the CUT worked correctly by examining the state of the SUT and its collaborators after the method was exercised
- Behavioral based testing (**Test by Process**)
 - determine whether the CUT worked correctly by examining its action process

Typical Scenario for Mock Object

- objects supplies nondeterministic results
 1. maximum/minimum value
 2. random result
 3. current time
- objects difficult to create or reproduce
 1. Network error
- objects not yet exist or may change behavior.
 1. want database query return same result

More about JUnit5



JUnit 4

The new major version of the programmer-friendly
testing framework for Java

User Guide

 Javadoc


Code & Issues

Q & A


Support JUnit

JUnit org: <https://junit.org/junit5/>

More About Junit5

 [About](#) [Products](#) [For Teams](#)

How are we doing? Please help us improve Stack Overflow. [Take our short survey](#)

[Home](#)
[PUBLIC](#)
[Stack Overflow](#)
[Tags](#)
[Users](#)
[FIND A JOB](#)
[Jobs](#)
[Companies](#)
[TEAMS](#) [What's this?](#)
 [Free 30 Day Trial](#)

Questions tagged [junit5]

Ask Question

Version 5 of the popular JUnit testing framework for the JVM. JUnit is a framework for writing repeatable tests. It is an instance of the xUnit architecture for unit testing frameworks.

[Learn more...](#) [Top users](#) [Synonyms](#)

1,879 questions


Newest

Active

Bountied

Unanswered

More ▾

 Filter

0 votes

1 answer


30 views

Running JUnit5 from Command line with dependencies

I have a Java project with the following files: C:\MyProject\myPackage\MyTests.class C:\MyProject\lib\junit-platform-console-standalone-1.5.2.jar C:\MyProject\lib\other-library.jar The MyTests file ...

[java](#) [cmd](#) [junit5](#)

asked yesterday

 [Kel Varnsen](#)
156 • 7

Questions on stackoverflow : <https://stackoverflow.com/questions/tagged/junit5>

More About Junit5



GITTER

Where communities thrive

JOIN OVER 1.5M+ PEOPLE
JOIN OVER 100K+ COMMUNITIES
FREE WITHOUT LIMITS
CREATE YOUR OWN COMMUNITY

EXPLORE MORE COMMUNITIES

junit-team/junit5 The new major version of the programmer-friendly testing framework for Java

```
args("--include-engine", "cucumber") args("--reports-dir",  
reportsDir) }
```

task consoleLauncherTest(type:JavaExec) {
dependsOn(testClasses)
def reportsDir = file("\$buildDir/test-results")
outputs.dir(reportsDir)
classpath = sourceSets["test"].runtimeClasspath
main = "org.junit.platform.console.ConsoleLauncher"
args("--scan-classpath")
args("--include-engine", "cucumber")
args("--reports-dir", reportsDir)
}

I have following config for Cucumber

```
@Cucumber  
@CucumberOptions(  
glue = {"stepdefs"},  
plugin = {"html:build/cucumber-reports/report.html"},  
features = {"src/test/resources/features"})  
public class CucumberRunnerTest {  
  
}
```

PEOPLE **REPO INFO**

SEE ALL (491 PEOPLE)

codecov[bot] commented #2446 05:56
codecov[bot] commented #2446 05:51
codecov[bot] commented #2446 05:51
sormuras synchronize #2446 05:50
sormuras on java15 05:50

Junit Gitter: an instant chat room which can talk directly with the Junit5 members

<https://gitter.im/junit-team/junit5>

Summary

- Junit is the most popular unit test frame for Java which sharpens the unit test techniques
- Mockito is a mock library used for which helps running code under test isolatedly

The End