Testing

Manual testing vs. automated testing

Blackbox testing 和 white box testing

黑箱 integration testing， 里面不太知道

White box 可以unit testing function

Users

Case

Bound of use

Failure condition

Test cases and how would you perform

Normal case

Extremes case like null

Special or strange case

Manual and automatic

功能，性能，压力

Exception: 异常理解<http://www.cnblogs.com/chenssy/p/3453039.html>

<http://www.runoob.com/java/java-exceptions.html>

Edge case, corner case

Junit

<http://www.tutorialspoint.com/junit/junit_interview_questions.htm>

Why not just use System.out.println() for testing?

Debugging the code using system.out.println() will lead to manual scanning of the whole output every time the program is run to ensure the code is doing the expected operations. Moreover, in the long run, it takes lesser time to code JUnit methods and test them on class files.

Why does JUnit only report the first failure in a single test?

Reporting multiple failures in a single test is generally a sign that the test does too much and it is too big a unit test. JUnit is designed to work best with a number of small tests. It executes each test within a separate instance of the test class. It reports failure on each test.

How do I test things that must be run in a J2EE container (e.g. servlets, EJBs)?

Refactoring J2EE components to delegate functionality to other objects that don't have to be run in a J2EE container will improve the design and testability of the software. Cactus is an open source JUnit extension that can be used for unit testing server-side java code.

Cactus is retired.

What are the core features of JUnit?

JUnit test framework provides following important features −

* Fixtures
* Test suites
* Test runners
* JUnit classes

Name the tools with which JUnit can be easily integrated.

JUnit Framework can be easily integrated with either of the followings −

* Eclipse
* Ant
* Maven

What are the core features of JUnit?

JUnit test framework provides following important features −

* Fixtures
* Test suites
* Test runners
* JUnit classes

What is a fixture?

Fixture is a fixed state of a set of objects used as a baseline for running tests. The purpose of a test fixture is to ensure that there is a well known and fixed environment in which tests are run so that results are repeatable. It includes following methods −

* setUp() method which runs before every test invocation.
* tearDown() method which runs after every test method.

What is a test suite?

Test suite means bundle a few unit test cases and run it together. In JUnit, both @RunWith and @Suite annotation are used to run the suite test.

What is a test runner?

Test runner is used for executing the test cases.

What are JUnit classes? List some of them.

JUnit classes are important classes which are used in writing and testing JUnits. Some of the important classes are−

* **Assert** − It contains a set of assert methods.
* **TestCase** − It contains a test case defines the fixture to run multiple tests.
* **TestResult** − It contains methods to collect the results of executing a test case.
* **TestSuite** − It is a Composite of Tests.

**Explain how you can write a simple JUnit test case?**

* Determine a subclass of TestCase
* To initialize object(s) under test, override the setup() method
* To release object(s) under test override the teardown() method

测试三步曲，

1. Define the test cases, null/ normal/ edge corner, strange input like different para type

2. Define the results

3. assertEquals

**void assertEquals(boolean expected, boolean actual)**   
检查两个变量或者等式是否平衡

import org.junit.Test;

import static org.junit.Assert.\*;

public class TestJunit1 {

@Test

public void testAdd() {

//test data

int num= 5;

String temp= null;

String str= "Junit is working fine";

//check for equality

assertEquals("Junit is working fine", str);

//check for false condition

assertFalse(num > 6);

//check for not null value

assertNotNull(str);

}

}