

Software Testing & Verification



The Foundation (2)



Instructor: Haiying SUN

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

Office: Science Build B1104

Available Time: Wednesday 8:00 a.m. – 12:00 a.m.

Exercise Time

课程网站 → 课堂练习 → 课堂练习-1

扫描二维码



下课测统计结果

第1题： 使用手机时出现的“闪退”或者“打怪”时游戏崩溃是软件Bug [单选题]

正确率: 60.29%

选项	小计	比例
对	27	<div><div></div></div> 39.71%
错 (答案)	41	<div><div></div></div> 60.29%

第2题： 你认为下面描述中哪个/些是软件Bug [多选题]

正确率: 25.00% [显示全部题目正确率](#)

选项	小计	比例
系统输出不正确的结果	49	<div><div></div></div> 72.06%
对用户需求理解错误	14	<div><div></div></div> 20.59%
代码中遗漏功能特性的实现 (答案)	51	<div><div></div></div> 75.00%
系统长时间没有响应	34	<div><div></div></div> 50.00%

Agenda



- Defect, Fault and Failure
- Test Cases

Agenda



- Defect, Fault and Failure
- Test Cases

Defect Definitions

- 软件缺陷(Software Defect)

- 存在于软件（文档、数据、程序）之中的那些不希望或不可接受的偏差
- Bug是口语化的缺陷



```
typedef int Time;
typedef int Distance;
typedef int Speed;

const Speed SAFETY_SPEED = 120;

void increase_speed (Speed s);

void check_speed (Time t, Distance d)
{
    assert(t > 0);
    Speed s = d/t;
    if (s < SAFETY_SPEED)    increase_speed (s);
}

void perform_safety_checks ()
{
    Time t = get_time ();
    Distance d = get_distance ();
    check_speed (d, t);
}
```

Exercise

```
3      import java.util.ArrayList;
4
5      public class Main {
6
7      public static void main(String[] args) {
8          // write your code here
9          ArrayList<String> bugs = new ArrayList<>();
10
11          bugs.add("Code Bug");
12          bugs.add("Requirement Bug");
13
14          for (String bug:bugs)
15              if (!(bug.equals("Logical Bug")))
16                  bugs.add("Logical Bug");
17      }
18  }
```

Defect Definitions

- 软件失效(**Software Failure**):

- 软件运行时产生的一种不希望或不可接受的外部行为结果。
- 系统崩溃，闪退，结果不正确.....

① 外部可见的软件失效是测试中推断软件中存在缺陷的基本方法

② 没有失效并不代表软件中不存在缺陷

课程网站 → 课堂练习 → 课堂练习-2-1



Defect Definitions

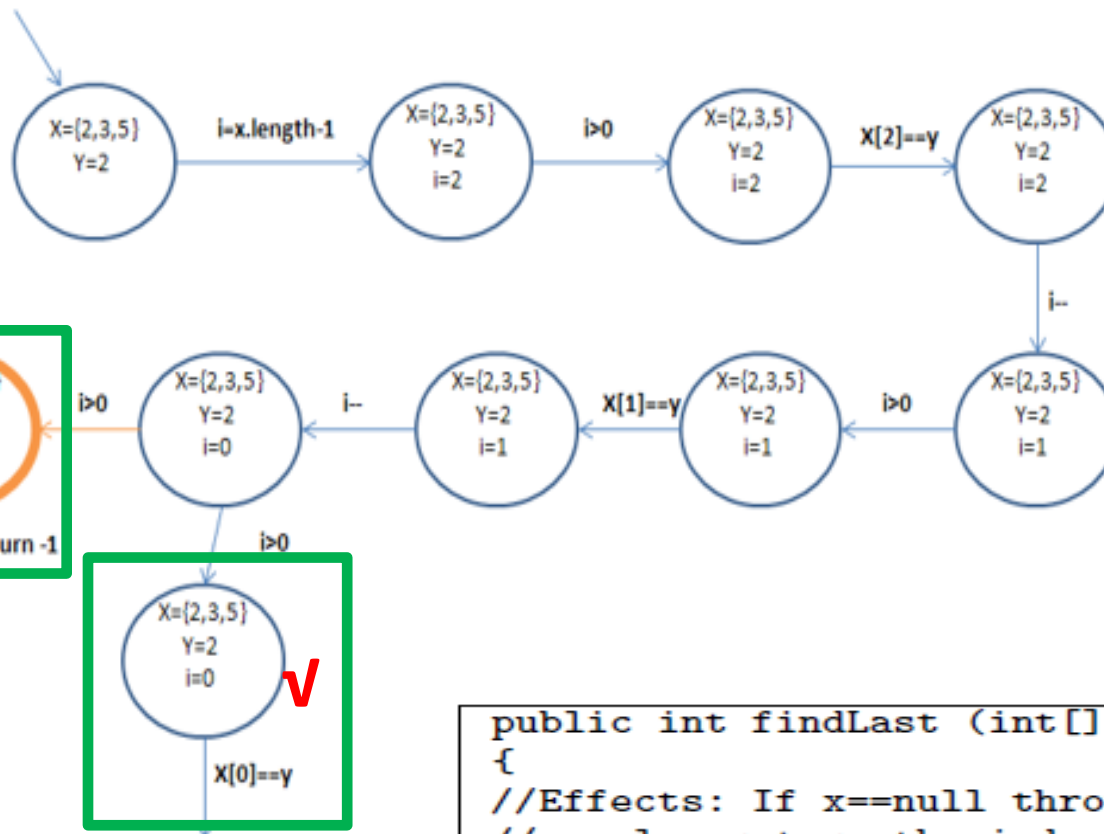
```
public int findLast (int[] x, int y)
{
    //Effects: If x==null throw NullPointerException
    //    else return the index of the last element
    //    in x that equals y.
    //    If no such element exists, return -1
    for (int i=x.length-1; i > 0, i--)
    {
        if (x[i] == y)
        {
            return i;
        }
    }
    return -1;
}

// test:  x=[2, 3, 5]; y = 2
//        Expected = 0
```

- 在输入x={2,3,5}; y=2的情况下，期望输出 是0，而实际输出 是-1，
- 实际输出-1就是一个失效。

Defect Definitions

- 软件故障(Software Fault):
 - 软件运行过程中出现的一种不希望或不可接受的内部状态
 - 内部状态：由所有程序变量的当前值和程序指针构成



在输入为(x={2,3,5}, y=2,) 下程序执行状态的变化

```

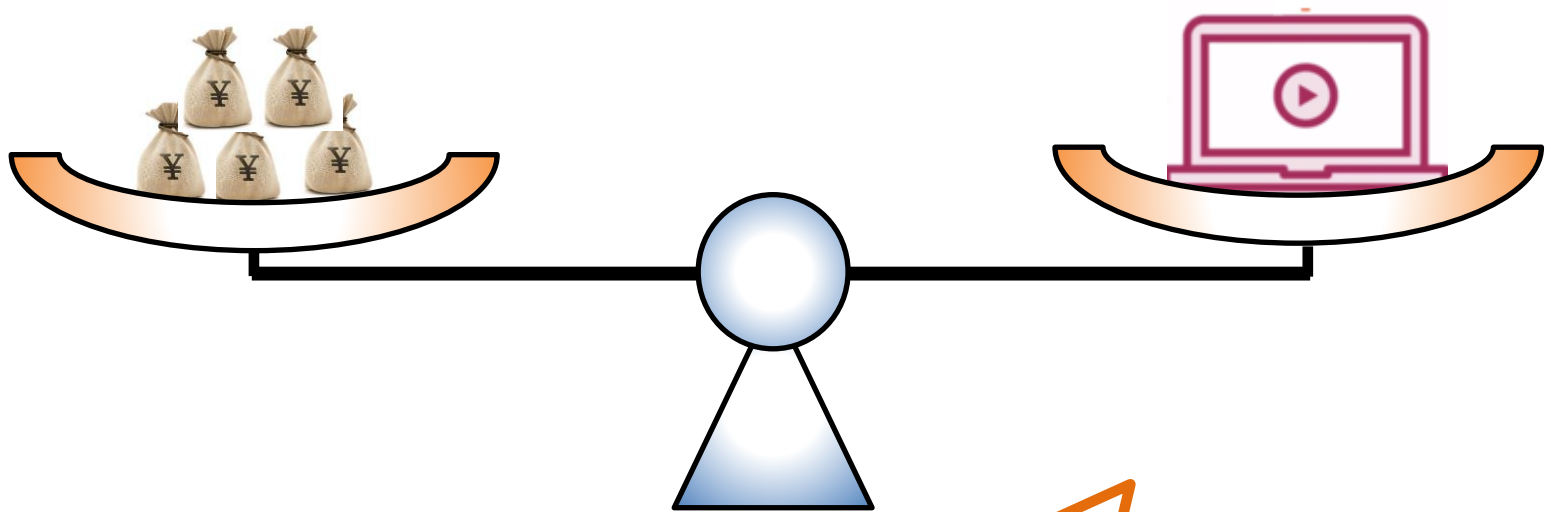
public int findLast (int[] x, int y)
{
    //Effects: If x==null throw NullPointerException
    //  else return the index of the last element
    //  in x that equals y.
    //  If no such element exists, return -1
    for (int i=x.length-1; i > 0; i--)
    {
        if (x[i] == y)
        {
            return i;
        }
    }
    return -1;
}

// test:  x=[2, 3, 5]; y = 2
//         Expected = 0
  
```

Defect Definitions

- 缺陷具有**隐蔽性**
 - 软件中一定存在缺陷
 - 有缺陷并不一定产生故障
 - 有故障也并不一定会失效
- **软件错误(Software Error):**
 - 在软件生存期内的不希望或者不可接受的**人为错误**
 - **软件缺陷**本质上是**研发人员**在软件研发过程中**所犯错误在软件中的可视化结果**

Kernel Challenge



有限的资源条件下，无法也不可能穷尽所有可能的系统输入

Defect Definitions

- RIP model

- Three conditions necessary for a failure to be observed

- I. **Reachability** : The location or locations in the program that contain the defect must be reached
- II. **Infection** : **After executing the location**, the state of the program must be incorrect
- III. **Propagation** : The infected state must propagate to cause some output of the program to be incorrect

Example

```
public int findLast (int[] x, int y)
{
    //Effects: If x==null throw NullPointerException
    //  else return the index of the last element
    //  in x that equals y.
    //  If no such element exists, return -1
    for (int i=x.length-1; i > 0; i--)
    {
        if (x[i] == y)
        {
            return i;
        }
    }
    return -1;
}

// test:  x=[2, 3, 5]; y = 2
//        Expected = 0
```

对于findLast下列输入是否可以发现代码中的缺陷

- ① x=[2,3,5], y=2
- ② x=Null, y=4
- ③ x=[]（空数组）， y=4
- ④ x=[1,2,4], y=4
- ⑤ X=[1], y=4

Example

```
public int findLast (int[] x, int y)
{
    //Effects: If x==null throw NullPointerException
    //  else return the index of the last element
    //  in x that equals y.
    //  If no such element exists, return -1
    for (int i=x.length-1; i > 0; i--)
    {
        if (x[i] == y)
        {
            return i;
        }
    }
    return -1;
}

// test:  x=[2, 3, 5]; y = 2
//        Expected = 0
```

对于findLast下列输入是否可以发现代码中的缺陷？

① $x=[2,3,5]$, $y=2$

由于期望输出是0，而实际输出是-1，因此，使用 $x=\{2,3,5\}$; $y=2$ 可能捕捉到该缺陷

② $x=Null$, $y=4$

不能执行到包含缺陷的代码，不具有可达性

Example

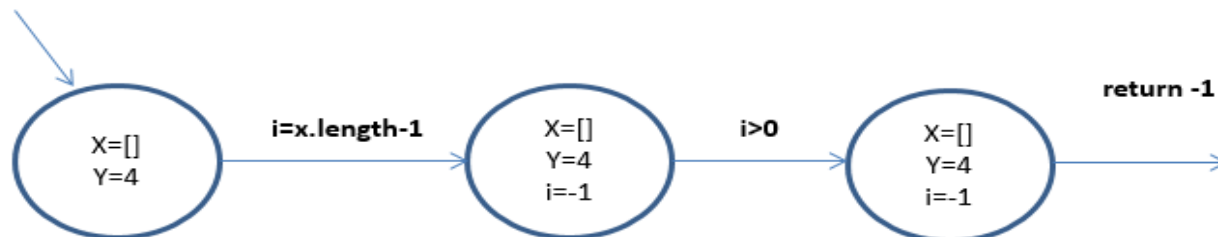
```
public int findLast (int[] x, int y)
{
    //Effects: If x==null throw NullPointerException
    //  else return the index of the last element
    //  in x that equals y.
    //  If no such element exists, return -1
    for (int i=x.length-1; i > 0; i--)
    {
        if (x[i] == y)
        {
            return i;
        }
    }
    return -1;
}

// test:  x=[2, 3, 5]; y = 2
//        Expected = 0
```

对于findLast下列输入是否可以发现代码中的缺陷

③ $x=[]$ （空数组）， $y=4$

具备可达性，不具感染性，在正确和错误代码下运行的内部状态都一样



Example

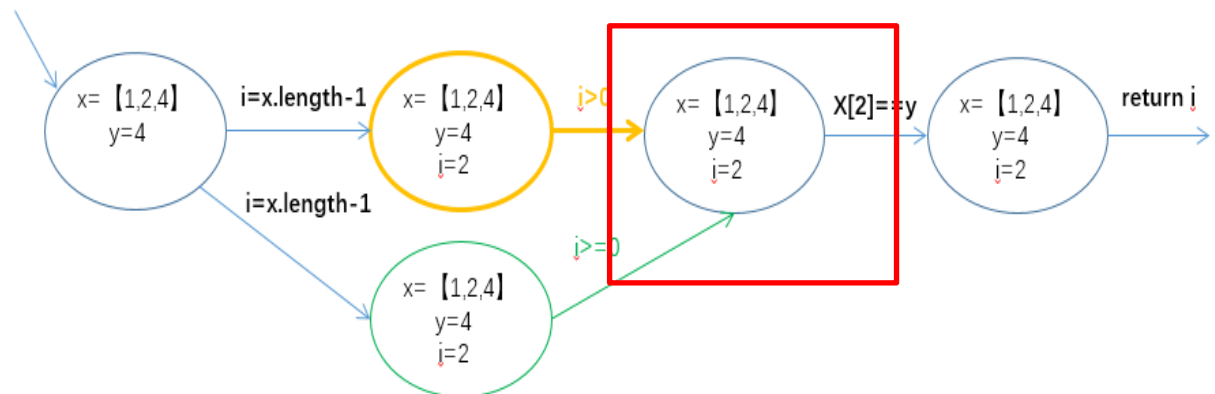
```
public int findLast (int[] x, int y)
{
    //Effects: If x==null throw NullPointerException
    // else return the index of the last element
    // in x that equals y.
    // If no such element exists, return -1
    for (int i=x.length-1; i > 0; i--)
    {
        if (x[i] == y)
        {
            return i;
        }
    }
    return -1;
}

// test:  x=[2, 3, 5]; y
//         Expected = 0
```

对于findLast下列输入是否可以发现
代码中的缺陷

④ $x=[1,2,4]$, $y=4$

与第3问是一样的情况，具备可达性
但不具备感染性，注意到执行完成
错误代码后，程序的内部状态都是一
样的，图中红色的部分



Example

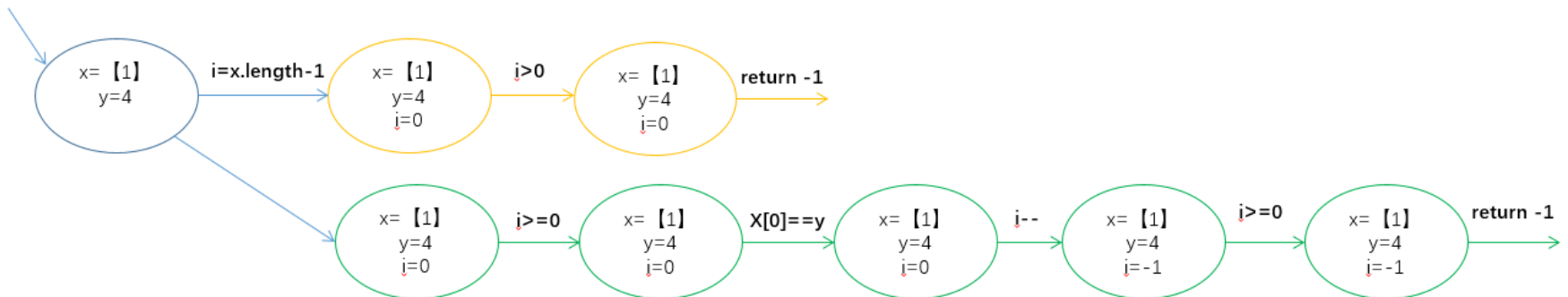
```
public int findLast (int[] x, int y)
{
    //Effects: If x==null throw NullPointerException
    //  else return the index of the last element
    //  in x that equals y.
    //  If no such element exists, return -1
    for (int i=x.length-1; i > 0; i--)
    {
        if (x[i] == y)
        {
            return i;
        }
    }
    return -1;
}

// test:  x=[2, 3, 5]; y = 2
//        Expected = 0
```

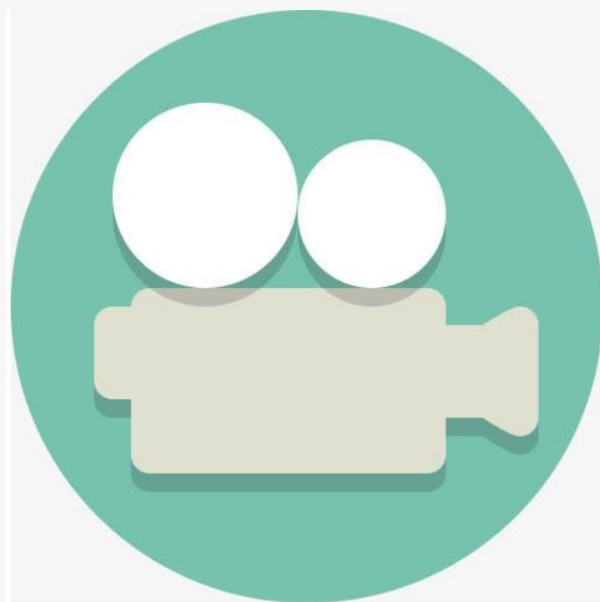
对于findLast下列输入是否可以发现
代码中的缺陷

⑤ $x=[1]$, $y=4$

具备可达性、感染性但不具备传播性，注意到虽然有故障出现（图中橘色的部分），但是错误代码和正确代码下输入的结果都是-1



Break



量子计算下测试是否仍旧存在？

Agenda



- Defect, Fault and Failure
- Test Cases

Test Case

- [Paul Amman]: A test case is composed of the **test case values**, **expected result**, prefix values, postfix values for a complete execution and evaluation of the software under test:
 - ① **Test Case Values** : The input values necessary to complete some execution of the software under test.
 - ② **Expected Results** : The result that will be produced when executing the test if the program satisfies its intended behavior
 - ③ **Prefix Values** : Any inputs necessary to put the software into the appropriate state to receive the test case values
 - ④ **Postfix Values** : Any inputs that need to be sent to the software after the test case values. There are usually two types of postfix values:
 - a) **Verification Values** : Values necessary to see the results of the test case values
 - b) **Exit Commands** : Values needed to terminate the program or otherwise return it to a stable state
- **Test Set** : A set combines with a set of Test cases

Test Case

- **Must** Ingredients
 - Test Case Inputs
 - Expected Results

Document



- Manual Test
- Excel

Code



- Automated Test
- Programming language

Manual Test Case Example

用例名称↵	生成准考证↵	↵
主要参与者↵	教务老师↵	↵
前置条件↵	教务老师登录系统。↵ 教务老师完成对报名考生的审核（资格审核通过一条）↵	↵
后置条件↵	教务老师完成准考证的生成↵	↵
基本流程↵	1、教务单击生成准考证，系统显示所有考生，同时显示资格审核状态和缴费状态↵ 2、教务老师选择生成准考证的学生，缺省是生成所有资格审核状态为通过的考生的准考证。↵ 3、系统按照规则，自动生成准考证号，准考证内容参见 req-1008，并记录考生的准考证状态为已生成。↵ 4、教务打印准考证。↵	↵
扩展流程↵	无↵	↵
约束↵	准考证的生成规则：321+两位年份+01/02+3 位顺序号↵	↵

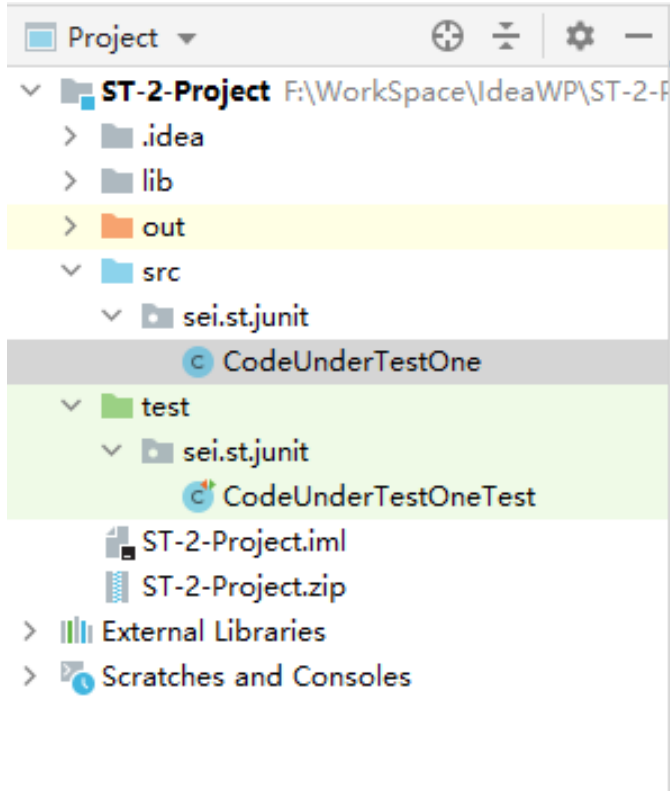
Use Case Specification Ingredients	Test Case Ingredients
Prefix Condition	Prefix Value
Inputs in Flow of Event	Test Input Value
System outputs in Flow of Event	Expected Result
Post Condition	Postfix Value
Constraints	Postfix Value

Manual Test Case Example

用例名称	生成准考证
主要参与者	教务老师
前置条件	教务老师登录系统。 教务老师完成对报名考生的审核（资格审核通过一条）
后置条件	教务老师完成准考证的生成
基本流程	1、教务单击生成准考证，系统显示所有考生，同时显示资格审核状态和缴费状态 2、教务老师选择生成准考证的学生，缺省是生成所有资格审核状态为通过的考生的准考证。 3、系统按照规则，自动生成准考证号，准考证内容参见 req-1008，并记录考生的准考证状态为已生成。 4、教务打印准考证。
扩展流程	无
约束	准考证的生成规则：321+两位年份+01/02+3 位顺序号

Prefix Values	Test Input Values	Expected Values	Postfix Values
教务的用户名，密码， 1位考生信息：孙海英 且孙海英资格审核通过， 已存在1为考生信息， 准考证号 3211701001	生成准考证命令	系统显示孙海英的信息，资格审核状态通过，缴费状态为已缴费	1、单击确定返回生成准考证 2、准考证验证脚本： 包括格式验证，用正则表达式表示为： ^321\d{2}[0][1 2]\d{3}\$； 准考证号不重复；准考证号按序增加
	选择生成孙海英准考证	系统生成准考证号为： 3211701002	
	打印准考证	页面显示“打印成功”	

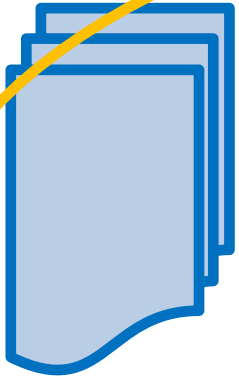
Automated Test Case Example



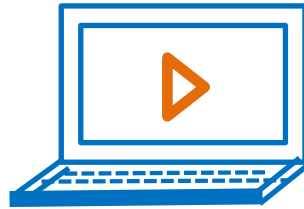
- 准备

1. 课程网站下载**ST-2-Project.zip**: 课程网站 → 课堂练习 → 课堂练习-2-2
2. 解压**ST-2-Project.zip**到指定目录（**目录中不要有中文**）
3. Idea中打开**ST-2-Project**
4. 掌握**CodeUnderTestOne**的功能，走读其实现代码（**src\CodeUnderTestOne.java**）和已有的测试代码(**test\ CodeUnderTestOneTest.java**)
5. 执行测试代码，查看结果

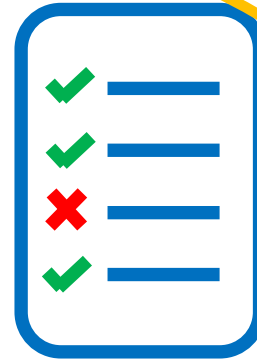
Automated Test Case



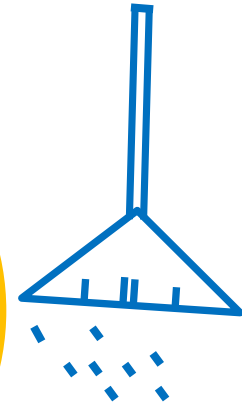
Arrange:
preparation for
the coming test



Action:
execute code
under test



Assert:
compare actual
outputs with
expected ones



Annihilation:
clean up test
environment

3A Pattern

“4A” Test Script Structure Pattern

Test Framework Tools

后端单元

5 JUnit 5



Angular – Karma & Jasmine



React – Jest & Enzyme



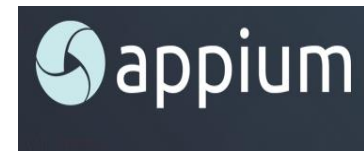
curl

Postman

Apache HttpClient Library

接口集成

Selenium WebDriver



系统功能



系统性能

Unit Test Script

```
11 import static org.mockito.Mockito.when;
12 import static org.mockito.Mockito.mock;
13
14
15
16 public class Get200withMock{
17
18
19     //public static final String BASE_ENDPOINT= "https://api.github.com";
20     //HttpClient client = HttpClientBuilder.create().build();
21
22     @Test
23     public void baseUrl_return200() throws IOException{
24
25         //HttpGet get = new HttpGet(BASE_ENDPOINT);
26         //HttpResponse response = client.execute(get);
27
28         HttpClient client = mock(HttpClient.class);
29         HttpGet get = mock(HttpGet.class);
30         HttpResponse response = mock(HttpResponse.class);
31         StatusLine statusLine = mock(StatusLine.class);
32
33         when(statusLine.getStatusCode()).thenReturn(200);
34         when(response.getStatusLine()).thenReturn(statusLine);
35         when(client.execute(get)).thenReturn(response);
36
37         int actual = response.getStatusLine().getStatusCode();
38
39         Assertions.assertEquals(actual, 200);
40
41     }
42
43 }
```

Integration Test Script

```
1  package ecnu.sei.st;
2
3  import java.io.IOException;
4
5  import org.apache.http.HttpResponse;
6  import org.apache.http.client.HttpClient;
7  import org.apache.http.client.methods.HttpGet;
8  import org.apache.http.impl.client.HttpClientBuilder;
9  import org.junit.jupiter.api.Test;
10 import org.junit.jupiter.api.Assertions;
11
12
13 public class Get200 {
14
15     public static final String BASE_ENDPOINT= "https://api.github.com";
16     HttpClient client = HttpClientBuilder.create().build();
17
18     @Test
19     public void baseurl_return200() throws IOException{
20
21         HttpGet get = new HttpGet(BASE_ENDPOINT);
22
23         HttpResponse response = client.execute(get);
24
25         int actualString = response.getStatusLine().getStatusCode();
26
27         Assertions.assertEquals(actualString, 200);
28
29     }
30
31 }
```

System Function Test Script

```
public class NoticeSearchTest{

    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;

    @BeforeEach
    public void setUp() {
        System.setProperty("webdriver.chrome.driver", "E:\\IdeaWP\\SeleniumWebDriver\\chromedriver.exe");
        driver = new ChromeDriver();

        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }

    @AfterEach
    public void tearDown() {
        driver.quit();
    }

    public String waitForWindow(int timeout) {
        try {
            Thread.sleep(timeout);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        Set<String> whNow = driver.getWindowHandles();
        Set<String> whThen = (Set<String>) vars.get("window_handles");
        if (whNow.size() > whThen.size()) {
            whNow.removeAll(whThen);
        }
        return whNow.iterator().next();
    }
}
```

System Function Test Script

```
@Test
public void noticeSearch() {
    driver.get("https://www.ecnu.edu.cn/");
    assertThat(driver.getTitle(), is( value: "华东师范大学"));
    driver.manage().window().setSize(
        new Dimension( width: 1280, height: 695));
    driver.findElement(By.linkText("在校学生")).click();
    vars.put("window_handles", driver.getWindowHandles());
    driver.findElement(By.linkText("通知公告")).click();
    vars.put("win4185", waitForWindow( timeout: 2000));
    vars.put("root", driver.getWindowHandle());
    driver.switchTo().window(vars.get("win4185").toString());

    driver.findElement(By.id("go-search")).click();
    driver.findElement(By.name("keyword")).click();
    driver.findElement(By.name("keyword")).sendKeys( ...charSequences: "四六级");
    vars.put("window_handles", driver.getWindowHandles());
    driver.findElement(By.cssSelector(".search")).click();
    vars.put("win9039", waitForWindow( timeout: 2000));
    driver.switchTo().window(vars.get("win9039").toString());
    assertThat(driver.getTitle(), is( value: "四六级 - WebplusPro 搜索"));
    vars.put("window_handles", driver.getWindowHandles());

    driver.findElement(By.cssSelector(".result_item:nth-child(2) > .item_title div")).click();
    vars.put("win9254", waitForWindow( timeout: 2000));
    driver.switchTo().window(vars.get("win9254").toString());
    assertThat(driver.getTitle(), is( value: "关于2019年下半年全国大学英语四六级考试的通知"));

    driver.close();
    driver.switchTo().window(vars.get("win9039").toString());
    driver.close();
    driver.switchTo().window(vars.get("win4185").toString());
    driver.close();
    driver.switchTo().window(vars.get("root").toString());
    driver.close();
}
```


Performance Test Case

并发数/个	响应时间要求/s	TPS要求	加压时间/min	加压方式
1	<1	N/A	3	直接加压
10	<1	N/A	10	1个用户/5s
20	<1	N/A	10	1个用户/5s
50	<1	>100笔/s	20	1个用户/5s
100	<1	>100笔/s	20	1个用户/5s
200	<1	>100笔/s	10	1个用户/1s

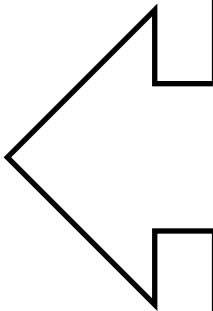
独立性能测试场景设计规划

独立性能测试场景设计实例

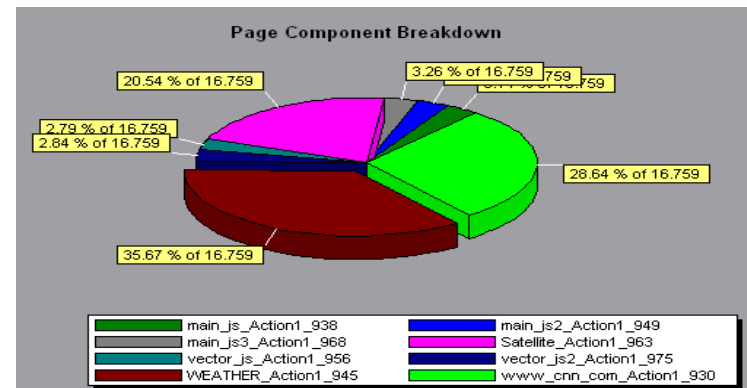
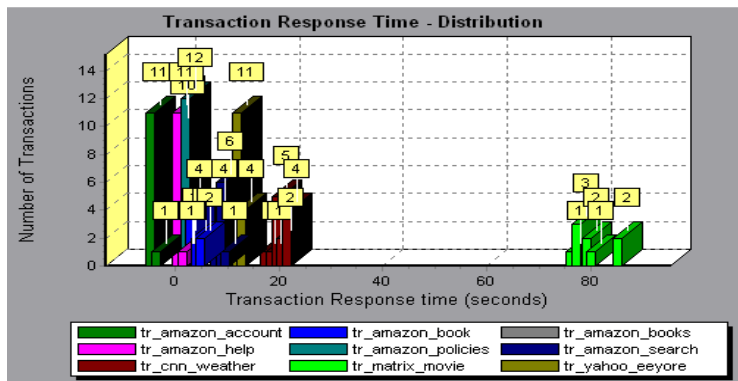
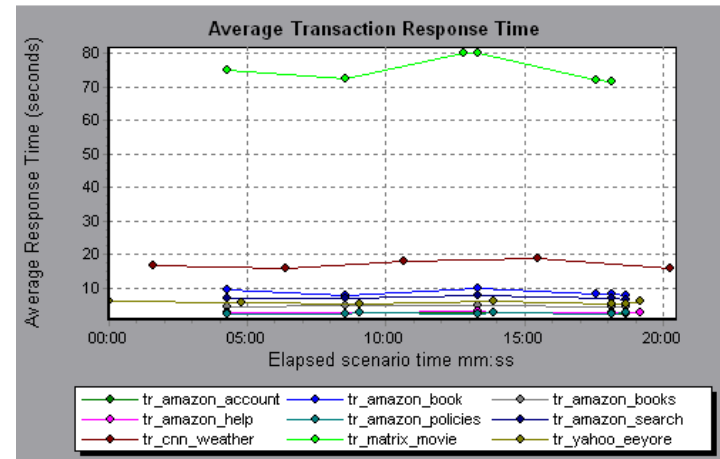
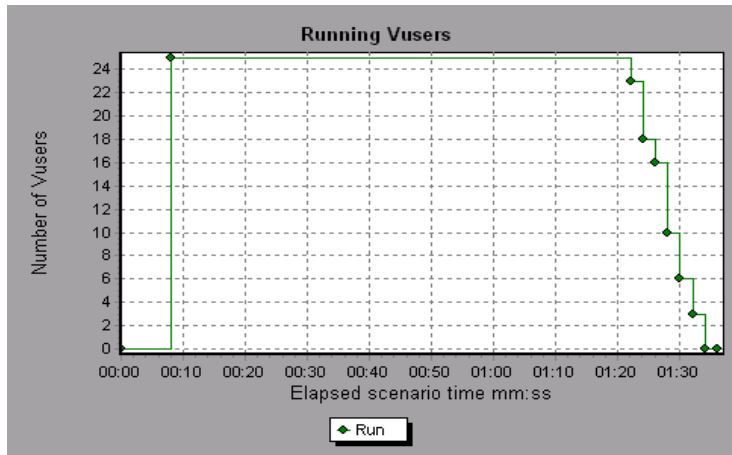
序号	用例编号	场景名称	优先级	业务/脚本名称	当前日业务量	事务个数	事物输出级别	并发用户数	加压方式			执行预估 (小时)
									加载方式	持续时间	退出方式	
1	D9AA001	账户查询	高	账户查询	168716	10	3	1/20/50/100	1个/5秒	第1-2组：10分钟 第3-4组：20分钟	1个/10秒	2
2	D9AA002	网银信息查询	高	网银信息查询	36668	6	3	1/20/50/101	1个/5秒	第1-2组：10分钟 第3-4组：21分钟	1个/10秒	2
3	D9AA003	通知存款	高	通知存款	9579	6	3	1/20/50/102	1个/5秒	第1-2组：10分钟 第3-4组：22分钟	1个/10秒	2
4	D9AA004	活期转活期	高	活期转活期	4739	4	3	1/20/50/103	1个/5秒	第1-2组：10分钟 第3-4组：23分钟	1个/10秒	2
5	D9AA005	定活互转	高	定活互转	3890	6	3	1/20/50/104	1个/5秒	第1-2组：10分钟 第3-4组：24分钟	1个/10秒	2
6	D9AA006	还款	高	还款	9579	0	3	1/20/50/105	1个/5秒	第1-2组：10分钟 第3-4组：25分钟	1个/10秒	2
7	D9AA007	本行汇款	高	本行汇款	9288	0	3	1/20/50/106	1个/5秒	第1-2组：10分钟 第3-4组：26分钟	1个/10秒	2

Performance Test Script

```
"Name=reserveFlights.x", "Value=58", ENDITEM,  
"Name=reserveFlights.y", "Value=12", ENDITEM,  
LAST);  
  
web_submit_form("reservations.pl_3",  
  "Snapshot=t6.inf",  
  ITEMDATA,  
    "Name=firstName", "Value=Joseph", ENDITEM,  
    "Name=lastName", "Value=Marshall", ENDITEM,  
    "Name=address1", "Value=234 Willow Drive", ENDITEM,  
    "Name=address2", "Value=San Jose/CA/94085", ENDITEM,  
    "Name=pass1", "Value=Joseph Marshall", ENDITEM,  
    "Name=creditCard", "Value=12345678", ENDITEM,  
    "Name=expDate", "Value=10/09", ENDITEM,  
    "Name=saveCC", "Value=<OFF>", ENDITEM,  
    "Name=buyFlights.x", "Value=55", ENDITEM,  
    "Name=buyFlights.y", "Value=8", ENDITEM,  
  LAST);  
  
lr_think_time(4);  
  
lr_start_transaction("S01_T06_CheckItinerary");  
  
web_image("Itinerary Button",  
  "Alt=Itinerary Button",  
  "Snapshot=t7.inf",  
  -----
```

- 
1. 客户端性能
测试策略
 2. 录制/回放+
脚本编辑

Performance Test Results



Test Case

- Quality Attributes
 - 有效性：发现缺陷的能力
 - 仿效性：测试用例不能发现的缺陷，其它输入发现该缺陷的概率应该非常低
 - 经济性：实现执行成本
 - 可维护性：易于修改的程度

下课测

- 任务
 - 根据功能描述，续写测试脚本
 - 如果存在，揭示代码中的缺陷，提交结果



完了你的程序出Bug了

课程网站 → 课堂练习 → 课堂练习-2-
2→缺陷提交



Summary

- 缺陷是存在于软件中的与期望不相符的静态事物
- 故障是软件运行中动态产成的事物，表现为缺陷被激活时进程产生的与期望不相符的内部状态
- 失效是软件运行中动态产成的事物，表现为缺陷被激活时进程产生的与期望不相符的外部运行结果
- 输入发现缺陷的条件：同时满足可达性、感染性和传播性
- 测试用例是检测缺陷是否存在的输入样本，输入和预期结果是必须包括的两个成份
- 测试用例即可以使用代码实现、也可以使用自然语言描述
- 测试用例的质量属性有：有效性、仿效性、经济性和可维护性

The End