



第五部分 软件编码、测试与质量保障

- 5.1 软件编程
- 5.2 软件测试
- 5.3 白盒测试
- 5.4 黑盒测试
- 5.5 变异测试
- 5.6 性能测试



5.6 性能测试

- 性能测试的关注点
- 性能测试的指标
- 性能测试的模型
- 测试自动化的原理



性能测试基础——概念

■ 性能测试概念

■ 狭义性能测试

通过模拟生产运行的业务压力或用户使用场景来测试系统的性能是否满足生产性能的要求。

■ 广义性能测试

压力测试、负载测试、强度测试、并发测试、大量数据测试、配置测试、可靠性测试等性能相关测试的统称。



性能测试基础——概念

■ 性能测试概念

■ 压力测试:

通过对程序施加越来越大的负载，直到发现程序性能下降的拐点。

——强调压力大小变化

■ 负载测试:

不断增加压力或增加一定压力下持续一段时间，直到系统性能指标达到极限。

——强调压力增加和持续时间



性能测试基础——概念

■ 性能测试概念

■ 强度测试:

迫使系统在异常资源下运行，检查系统对异常情况的抵抗力。

——强调极端的运行条件

■ 并发测试:

多用户同时访问或操作数据时是否存在死锁或其他性能问题。

——强调对多用户操作的承受能力



性能测试基础——概念

■ 性能测试概念

■ 大数据测试:

在存储、传输、统计等业务中结合并发操作测试系统的**数据处理极限**。

——**强调数据处理能力**

■ 配置测试:

通过测试找到系统各项**资源的最优分配**原则。

——**强调资源优化配置**



性能测试基础——概念

■ 性能测试概念

■ 可靠性测试:

在给系统加载一定压力的情况下，使系统运行一段时间，以此检验系统是
否稳定。

——强调系统稳定性



性能测试基础——应用领域

■ 性能测试应用

■ 1. 系统性能瓶颈定位

最常见的应用

■ 2. 系统参数配置

寻找让系统表现最优的软、硬件配置参数

■ 3. 发现算法性能缺陷

针对多线程、同步并发算法



性能测试基础——应用领域

- 性能测试应用

- 4. 系统验收测试

- 验证预期性能指标

- 5. 系统容量规划

- 测试不同硬件环境下的性能表现

- 6. 产品评估/选型

- 全面评估产品，选择适合自己的产品类型



5.6 性能测试

- 性能测试的关注点
- 性能测试的指标
- 性能测试的模型
- 测试自动化的原理



性能测试基础——指标

■ 性能指标

■ 1. 并发用户数量

狭义并发：所有用户在同一时刻做同样的操作

广义并发：多个用户同时进行操作，但这些操作可以相同，也可以不同

虚拟用户（Vuser）：模拟真实用户操作来使用软件



性能测试基础——指标

■ 性能指标

■ 2. 响应时间

场景 (Scenario)：在一个性能测试期间依次发生并实现特定业务流程的事件总和

事务 (Transaction)：用户业务流程

思考时间 (Think Time)：用户操作过程中两个请求的间隔时间

■ 3. 吞吐量

一次测试过程中网络上传输的数据总和

吞吐量=吞吐量/传输时间



性能测试基础——指标

- 性能指标

- 4. TPS (Transaction Per Second)

- 每秒处理事务数

- 5. 点击率 (Hit Per Second)

- 每秒用户提交的HTTP请求数

- 6. 资源利用率

- CPU、磁盘、网络、数据库利用率



5.6 性能测试

- 性能测试的关注点
- 性能测试的指标
- 性能测试的模型
- 测试自动化的原理



性能测试基础——模型

■ 性能测试模型/策略

■ 1. 预期指标性能测试:

针对需求和设计阶段提出的性能指标而进行的测试

(性能测试基本要求)

■ 2. 独立业务性能测试:

核心业务模块是性能测试重点

■ 3. 组合业务性能测试

模拟多个用户，对多项业务进行组合性能测试（接近用户使用系统的真实情形）



性能测试基础——模型

■ 性能测试模型/策略

■ 4. 疲劳强度性能测试:

以一定负载长时间运行系统，测试系统长时间处理大量业务的能力和稳定性

■ 5. 大数据量性能测试:

测试运行时数据量较大或历史数据量较大时的性能情况

■ 6. 网络性能测试:

确认带宽、延迟、负载和端口变化对用户响应时间的影响

■ 7. 服务器性能测试:

确认数据库、Web服务器等瓶颈



性能测试基础——原则

■ 性能测试原则

■ 成本最优原则：

投入的测试成本能否使系统满足预先确定的性能目标

■ 策略为中心原则：

测试策略决定测试用例设计和测试实施

■ 适当裁剪原则：

对测试用例包含的内容进行合理裁剪，既满足测试需求，又节约测试成本

■ 模型完善原则：

具体测试中对模型不断调整、补充、完善，使之满足测试目的



性能测试——模型实例

Web MSN服务器端性能测试模型

在下列条件下，每台服务器内存消耗不超过20%，CPU平均值不超过30%：

1. 10万用户在线；
2. 10万用户中有1万在对话；
3. 10万用户中还有5千在改变状态或姓名；
4. 10万用户中还有5千在添加联系人



性能测试——模型实例

Web MSN客户端性能测试模型

在下列条件下，客户机内存消耗不超过10%，CPU平均值不超过5%：

1. 用户8小时在线；
2. 用户每半小时对话1次；
3. 用户每10分钟收到1个通知；
4. 用户每2小时改变1次状态；
4. 用户每4小时添加一个联系人



5.6 性能测试

- 性能测试的关注点
- 性能测试的指标
- 性能测试的模型
- 测试自动化的原理



性能测试——工具

- 主要性能测试工具

- 商业:

- Loadrunner
 - Rational Performance Tester
 - WebLoad

- 免费:

- Web Application Stress Tool
 - Application Center Test

- 开源:

- OpenSTA
 - Jmeter
 - Grinder

- 自行开发（针对某一个具体软件的一部分进行测试）



性能测试——测试自动化

- 自动化测试优点

- 1. 测试速度快
- 2. 测试结果准确
- 3. 高复用性
- 4. 永不疲劳
- 5. 可靠:

计算机不会弄虚作假

- 6. 能力:

有些手工测试做不到的地方，自动化测试可以做到。比如，在性能测试中模拟1000个用户同时访问网站



性能测试——测试自动化

■ 正确认识自动化测试

- 自动化测试有助于提高软件开发效率、缩短开发周期、节省人力资源等
- 机器执行测试用例按部就班，没有变通和创造力
- 人具有创造力，可以举一反三，从一个测试想到其他测试场景
- 资料显示，良好的自动化测试只能发现30%的问题，70%的问题还是由人工发现
- 对于复杂的逻辑判断、界面是否友好人工测试具有优势
- 自动化测试更适合于负载测试、性能测试和回归测试



性能测试——测试自动化

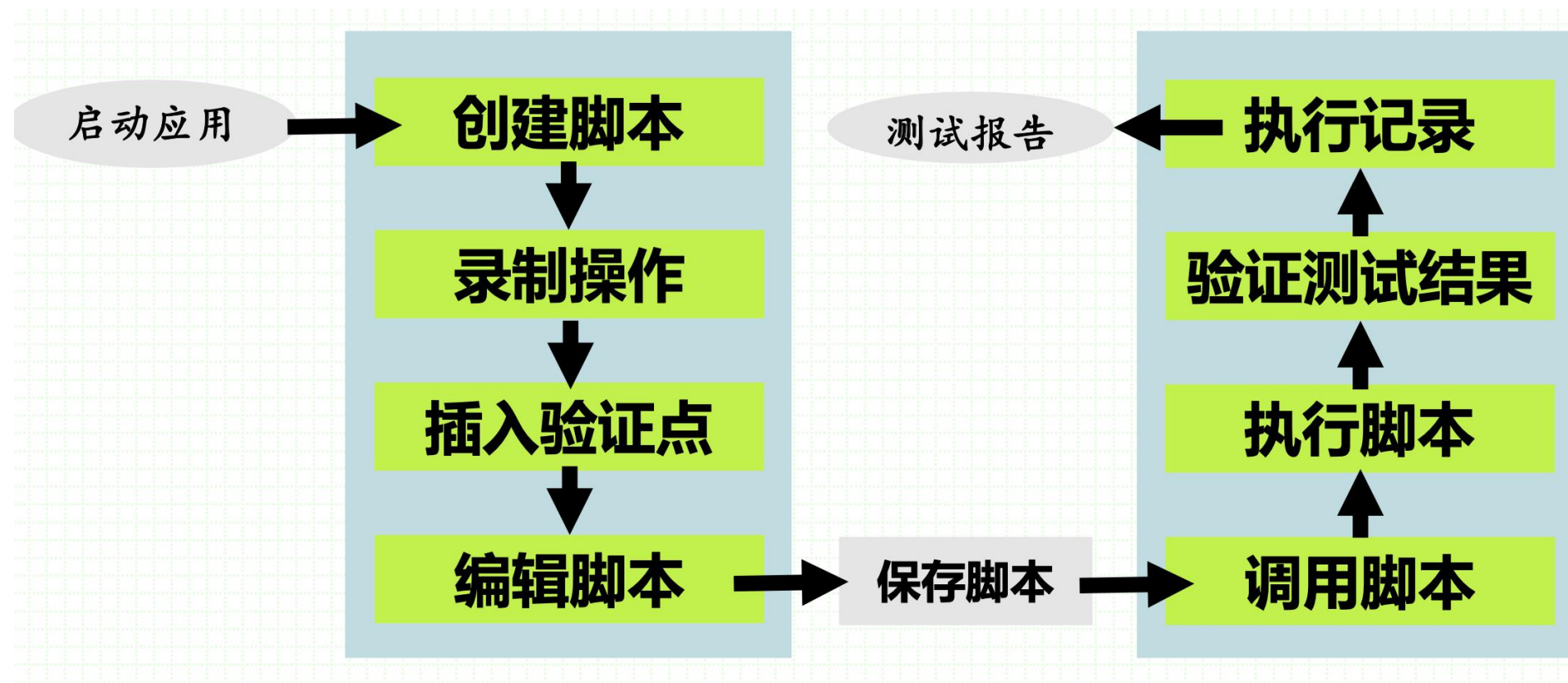
■ 不同测试阶段中的自动化测试

测试类型	自动化测试应用特点
单元测试	静态测试和动态测试均可借助工具实现一定程度的自动化
冒烟测试	自动化实现每日集成每日构建后的测试
功能测试	基本操作和数据验证可以实现自动化测试
负载测试 性能测试	一般只能由自动化测试工具完成
回归测试	自动化测试最能发挥作用的地方，应尽量实现自动化
验收测试 确认测试	自动化测试和手工测试协同完成



性能测试——测试自动化

■ 自动化测试原理





性能测试——测试自动化

- 自动化测试原理

- GUI对象识别

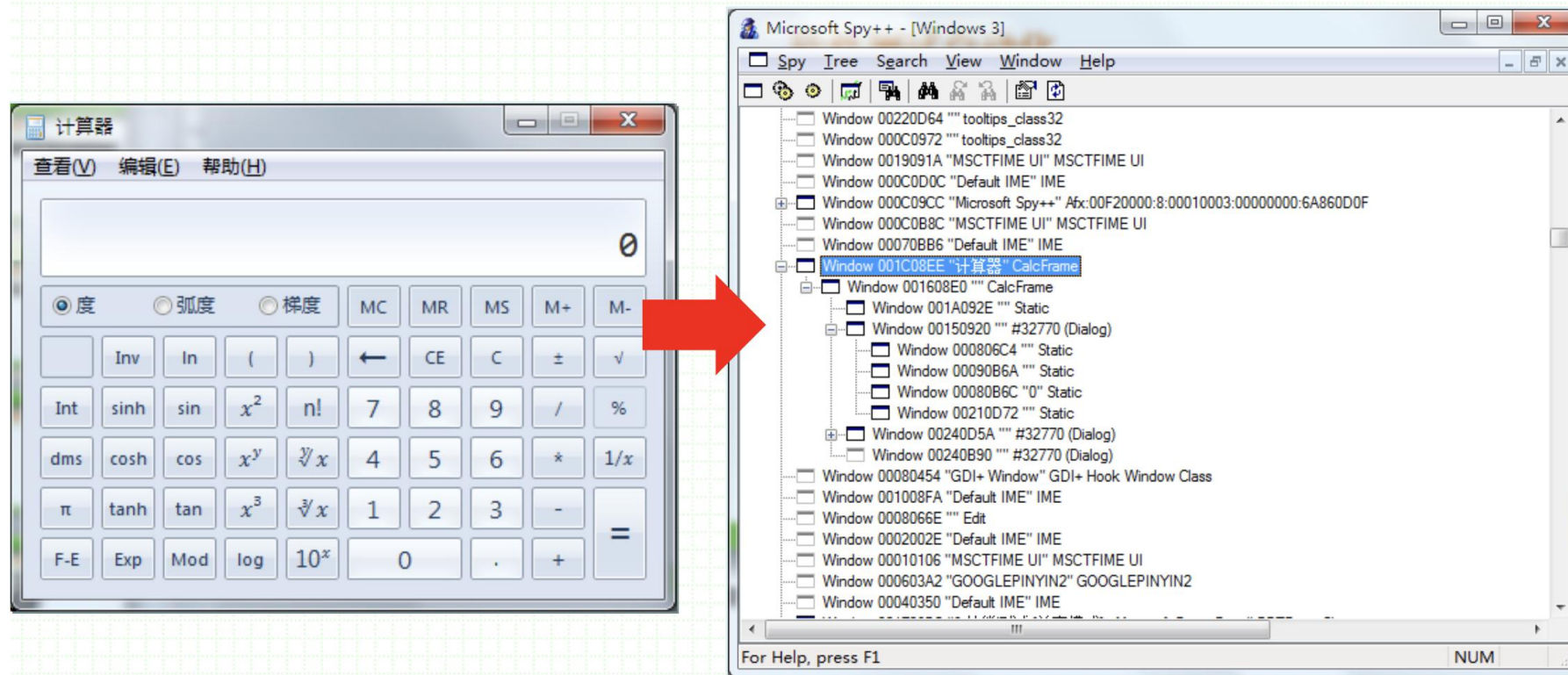
获得用户界面上对象的类别、名称、属性值

常见的GUI对象有：窗口、按钮、菜单、列表、编辑区域等

性能测试——测试自动化

Windows对象识别:

利用Windows API: FindWindow, GetWindowsRect等





性能测试——测试自动化

- 自动化测试原理

- DOM对象识别:

- DOM (Document Object Model) 文档对象模型

- 可以捕获Web页面上的各种对象

- 其他对象技术:

- C#/Java语言中的反射机制



性能测试——测试自动化

- 自动化测试原理

- 自动比较技术:

- 简单的数字、文字比较

- 复杂的图像比较

- 脚本技术:

- 脚本通过录制测试的操作而产生: 击键、移动、输入数据;

- 录制得到的脚本可被进一步编程。