

天津科技大学人工智能学院

软件测试技术实验报告

班级： 221032

姓名： 唐晨林

学号： 22103216

实验时间：

实验地点：

实验编号：实验 10

思考题

相同之处

目的相同

两者都用于替代脚本中的硬编码值，提供动态的数据输入，从而增强脚本的灵活性和适应性。解决数据依赖性问题，避免因重复使用相同数据而导致测试失败。

实现方式相似

两者都支持从外部数据源（如 CSV 文件、Excel 文件或数据库）读取数据。通过参数化工具提供数据驱动的方式，将变量与实际的测试数据关联。

支持的数据来源相似

支持多种数据源，如文件（CSV/文本）、内存变量、内置函数（如随机数）、数据库等。

提升脚本的复用性

- 1. 通过参数化实现相同脚本适应不同的测试场景，减少脚本维护成本。

不同之处

特性	LoadRunner 参数化	QTP 参数化
用途	用于性能测试中的虚拟用户数据驱动	用于功能测试中的自动化操作数据驱动
典型场景	模拟多个用户并发测试	验证单个用户在不同场景下的功能正确性
测试对象	以服务器和系统性能为目标	以前端功能正确性为目标

特性	LoadRunner 参数化	QTP 参数化
参数化粒度	针对协议层请求（如 HTTP 请求的参数、SQL 查询等）	针对 GUI 操作（如输入框、按钮等界面的控件）
数据共享	同一个参数可以共享给多个虚拟用户	通常一个参数用于单个测试用例或流程
运行环境	通过 Load Generator 在多台机器上并发运行	在本地或单台机器上运行
动态关联	更注重关联动态数据，如会话 ID 或令牌	通常不需要处理动态关联

参数化在软件测试中的作用

功能测试中的作用

验证功能的多样性

参数化用于验证功能在多种输入情况下的正确性，例如输入边界值、特殊字符和不同格式的数据。

覆盖更多测试场景

参数化通过提供多个数据集扩展了测试覆盖面，确保软件功能在不同输入条件下均能正常工作。

降低脚本维护成本

将测试数据从脚本中抽离，便于测试数据的管理和更新。

支持数据驱动测试

实现一次编写脚本、通过不同数据集重复执行的方式，提高了测试效率。

性能测试中的作用

模拟真实用户行为

参数化允许虚拟用户使用不同的数据进行操作，模拟实际用户在生产环境中的行为（如不同的用户名、订单号等）。

避免缓存和数据冲突

不同用户使用相同的数据可能导致缓存命中或数据冲突，通过参数化可以避免这些问题，确保测试结果的真实性。

评估系统在多场景下的性能

参数化模拟不同用户输入，从而测试系统在处理多样化请求时的性能表现。

提高测试可靠性

如果所有虚拟用户都使用相同的数据，会导致系统的某些路径重复执行；参数化可以分散负载，确保性能测试更全面。