

本周小结（第七周）

恭喜完成《软件测试与质量》课程第七周的学习。

本周我们讨论了课程的第三部分 测试管理与应用篇，本周我们主要讨论单元测试和代码扫描工具，展开讨论。

本周我们主要回答了如下的问题。

1 什么是单元测试？

单元测试是指对软件中的最小可测试单元或基本组成单元进行检查和验证。一般地，一个单元应具有明确的功能定义、性能定义，以及连接其他部分的接口定义等，且应可以清晰地与其他单元区分开来。因此，一个函数、一个类、一个窗口都可以看做是一个单元，在此所指的单元主要是指函数。

单元测试的目的是以分析单元的逻辑结构为基础，针对逻辑结构结合功能设计测试用例。单元测试的主要内容包括静态测试和动态测试，其中，静态测试即查看程序源代码，执行代码扫描，动态测试则需要运行程序，完成测试，主要包括对模块接口、模块边界条件、模块独立路径和错误处理进行测试。

单元测试的实施步骤包括：

- 理解设计；
- 概览源代码；
- 走读代码；
- 设计测试用例；
- 搭建测试环境；
- 执行测试用例，注意，此时可能需要设计开发驱动模块或桩模块；
- 重复进行设计测试用例和加载测试；
- 分析判断，给出结论。

2 什么是代码扫描工具 CA？

代码扫描工具 **Code Analyzer**（简称 **CA**）是通过分析或检查源程序的语法、结构、过程以及接口来检查程序的正确性和规范性，找出代码中隐藏的错误和缺陷，提高代码质量。**CA** 检查源代码的过程不需要实际执行程序。**CA** 实际是一种自动静态检查的软件工具。

CA 的功能主要包括：

- 支持基于词法规则、语法规则和语义规则的代码检查，甚至支持用户开发自己的规则包，**CA** 可基于用户自定义的规则包来检查代码；

- 支持跨 internet 实现源代码扫描的“云服务”；
- 支持主流 IDE 环境，实现对指定代码文件或整个项目源代码进行扫描检查。

CA 的架构如图 2.1 所示。

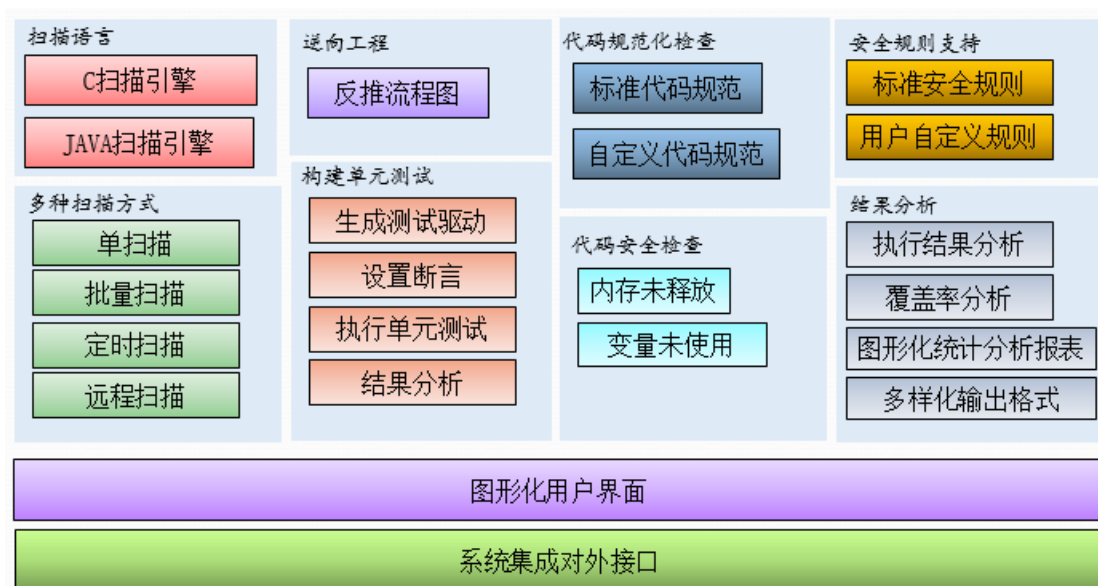


图 2.1 CA 工具的架构图

3 如何使用 CA 完成代码扫描？

CA 的代码分析流程如下：

- 代码预处理；
- 对代码进行语法分析；
- 生成代码执行树；
- 对代码进行语义分析；
- 生成分析报告。

CA 的使用非常简单，只需要导入源代码，设置好规则包，就可以启动执行代码分析，产生执行日志，对源代码的分析结果将显示在错误日志中，提供查看。

5 下周预告

下周，我们将开启本课程最后一部分，第五章 软件质量，并围绕软件质量模型、软件质量度量及其工具展开讨论。