

第十章 软件智能化开发支撑技术

10.1.1 软件分析的任务是什么

- 软件分析是指对软件进行人工或者自动分析，以验证，确认或发现软件性质的过程或活动。其目的在于发现软件中的缺陷，分析软件中缺陷的来源分布。

10.1.2 软件分析有哪些方法。

- 语法分析
- 类型分析
- 控制流分析
- 数据流分析
- 指针分析
- 过程间分析
- 别名分析
- 逃逸分析
- 形态分析
- 静态分析
- 动态分析
- 混合分析
- 克隆分析
- 结构分析
- 程序切片
- 剖面profiling
- 运行监控
- 符号执行
- 抽象解释
- 约束求解

10.1.3 软件分析有哪些应用

- 测试用例生成
- 程序自动生成
- 故障定位
- 故障修复

10.2.1 软件演化与软件维护的区别与联系是什么？

- 演化这个术语一般是指在性质和特性方面的递增式的改变。在某种意义上来说，这种属性的改变过程导致了新特性的出现或改进。一般来说，这种改变都是诸如改变类的成员以适应环境的改变，改变会使得他们更有用或更有意义，而且在某种程度上会增加他们的价值。与此同时，演化也会去掉一些不合适的软件特性。
- 软件维护是指在软件交付使用后所进行的修改，包括对错误的修正，提高性能，或其他属性，以及使软件产品适应改变的环境。维护试图保持系统，以有效的执行各项功能。然而维护只是在原有实现的基础上简单修正错误，忽略了迅速改变的环境和需求带来的问题。

10.3.1 软件架构是什么？列举你熟悉的软件架构？

- 软件架构使一种系统草图，为软件系统提供了一个结构，行为和属性的高级抽象，由构件的描述，构件的相互作用，指导构件集成的模式以及这些模式的约束组成。
- 分层模式
- 客户端/服务器模式
- 主/从模式
- 管道/过滤器模式
- 代理模式
- 对等模式
- 事件总线模式
- MVC模式
- 黑板模式
- 解析器模式

10.4.1 根据你所熟悉或听说过的设计模式，谈谈你对设计模式的看法。

- Singleton, Builder, Factory等。
- 设计模式可以增强可复用性，但是往往通过引入额外的抽象层次来获得程序的可变和灵活使得程序变得复杂而且牺牲了一定的性能。

10.4.2 你在开发中用到过哪些设计模式？用在什么场合？

- 以制作网页为例，其主要的功能需求就是用户可以和网页的元素互动，将请求发送给服务器，服务器再回传信息至用户界面。在设计这样的系统的时候，我们主要考虑以下几点：
 - 希望服务器接受的和发送请求的相关逻辑能够与服务器处理请求的逻辑分离
 - 希望用户页面的显示逻辑和发送，接受服务器信息的逻辑分离。

为了实现以上的逻辑分离，我们可以自行设计系统，但是无法保证设计的质量和未来的可复用性。我们可以采取已有的MVC设计模式，这是在网页中经常用的一种设计模式，主要效果是将业务逻辑与数据和显示分离。

10.5.1 软件重构是什么

- 软件重构是指在不改变软件的功能和外部可见性的情况下，为了改善软件的结构，提高软件的清晰性，可扩展性和可重用性而对其进行的改造。
- 简而言之，改进已经写好的软件设计。

10.5.2 使用IDE某项重要功能，并指出该功能的优缺点

- JetBrains家的IDEA的refactor功能，可以在保持依赖的情况下，对类名变量名等进行一个全局的修改，好处是可以很快很方便的进行整体的重构，而一个个的去进行修改。缺点是统一的修改很多时候会引入不易捕捉的bug，尤其是在源代码本身不规范的情况下。

10.5.3 在手工重构时，该如何实施才能保证重构的准确性？

- 关注以下场景
 - 代码异味
 - 代码重复
 - 方法过长
 - 参数列表过长
 - 模块耦合度
- MR.Nie 忘了抄手工重构了，上面是重构关注场景。

10.6.1 软件控制论的核心是什么？

- 如何建立软件行为的控制模型，设计方法和控制理论，以实施对它们的有效，量化的控制。

10.7.1 历史上出现过哪些比较有影响的软件开发方法

- 面向对象软件工程
- 面向方面软件工程
- 面向构建软件工程
- 敏捷软件工程
- 基于模型的软件工程
- 基于搜索的软件工程