

阅读低代码平台Mendix之 RestServices-2 Consume REST

一起来读开源工业软件系列3

0.1 回顾上次讨论内容

- 0 本视频的观看对象和前置知识要求
- 1 阅读开源软件的流程
- 2 浏览Mendix官网
- 3 阅读Mendix GitHub代码仓库内容
- 4 在IDE里阅读Mendix之RestServices
 - 代码结构
 - 代码审查方法
 - 代码质量监测标准
 - 代码审查工具

0.2 视频的观看对象

- 想学习如何阅读软件程序
- 熟悉Java语言的基础知识
- 了解Web应用系统架构
- 熟悉Java代码规范
- 了解Java开发工具

讨论话题

1 先导知识

- 1.1 面向对象程序设计的特点
- 1.2 REST (Representational state transfer)
- 1.3 Java多线程
- 1.4 防御式编程
- 1.5 UML：对象图、类图、时序图

2 阅读RestConsume

- 应用面向对象程序设计方法
- 检查复杂度和思考可能的改善方法

3 建议学习内容

1 先导知识

1.1 面向对象设计

- 抽象
- 封装
- 多态
- 继承
- 相比面向过程，面向对象的优点是？

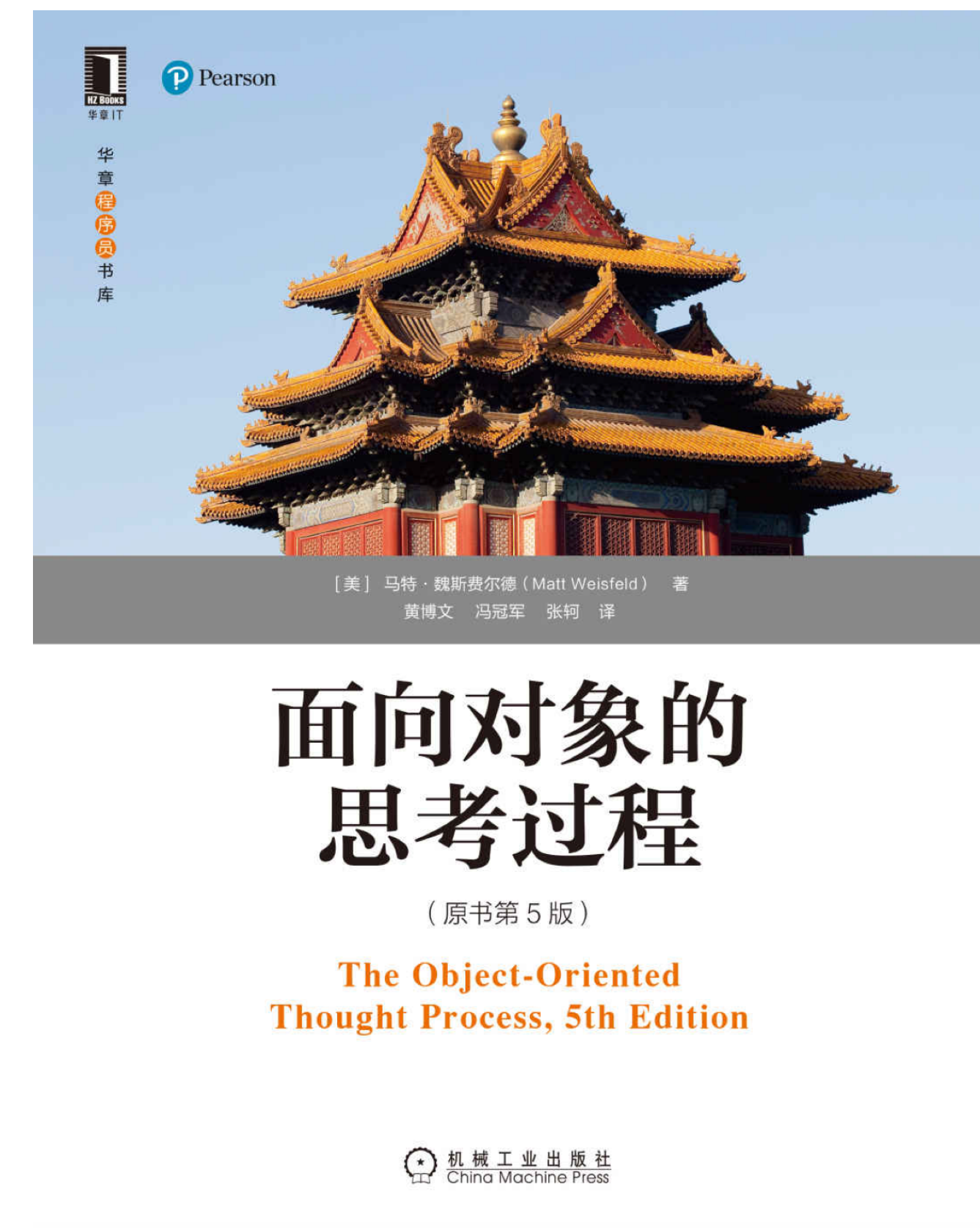
1.1 面向对象的设计方法

- 参考资料：面向对象分析和设计
- 类和对象：【面向对象分析和设计第3章】
- 面向对象开发的好处和风险：【面向对象分析和设计第7章7.9】
- 布朗大学CS0150 面向对象课程 CS0150 is one of the introductory Computer Science courses offered at Brown University. <https://cs.brown.edu/courses/cs015/>



面向对象的思维方式

- 面向对象的基本概念
 - 面向过程和面向对象编程的区别
 - 对象的属性和行为
- 如何以面向对象的方式进行思考
 - 清楚接口和实现之间的区别
 - 深入理解抽象
 - 给用户提供尽可能少的接口
- 参考【面向对象的思考过程（原书第5版）（华章程序员书库）第1章、第2章、第12章面向对象设计的SOLID原则】



1.2 REST (Representational state transfer)

- <https://docs.mendix.com/refguide/consumed-rest-services/>
- Representational state transfer 代表性状态转移 (REST) 是一种消费或暴露资源的方法。它因其简单性而受到欢迎，因为不需要大量的模式或合约来传输端点之间的数据。
 - HTTP URLs来定位资源
 - HTTP标头来验证和指定内容类型 (如XML或JSON) 。
 - HTTP方法来识别对资源的操作，如GET (检索数据) 或POST (发送数据) 。

1.3 Java多线程

- 参考书籍《Java编程思想》（中文版） <https://github.com/apacheecn/thinking-in-java-zh/blob/master/1.9.md>

1.4 防御式编程方法

- 参考资料：【代码大全第8章】 【编程匠艺1.5】
- 输入处理
- 错误处理技术
- 异常处理：【effective java 69-77】 Effective Java 3rd edition <https://github.com/echoJava/book/blob/master/Java/Effective%20Java%2C%20Third%20Edition.pdf> <https://githubhelp.com/mz157589766/Effective-Java-3rd-edition-Chinese-English-bilingual> 参考spring security代码
 - <https://github.com/apache/cn/thinking-in-java-zh/blob/master/1.8.md>
- 隔离方法：数据验证类
- 单元测试 参考资料：【构建之法第2章2.1】 参考spring security代码

防御式编程方法参考例子

- 9 Best Practices to Handle Exceptions in Java <https://stackify.com/best-practices-exceptions-java/>
- Why, When and How to Implement Custom Exceptions in Java <https://stackify.com/java-custom-exceptions/>
- Java SQL Injection Guide: Examples and Prevention <https://www.stackhawk.com/blog/java-sql-injection-guide-examples-and-prevention/>

防御式的编程技巧

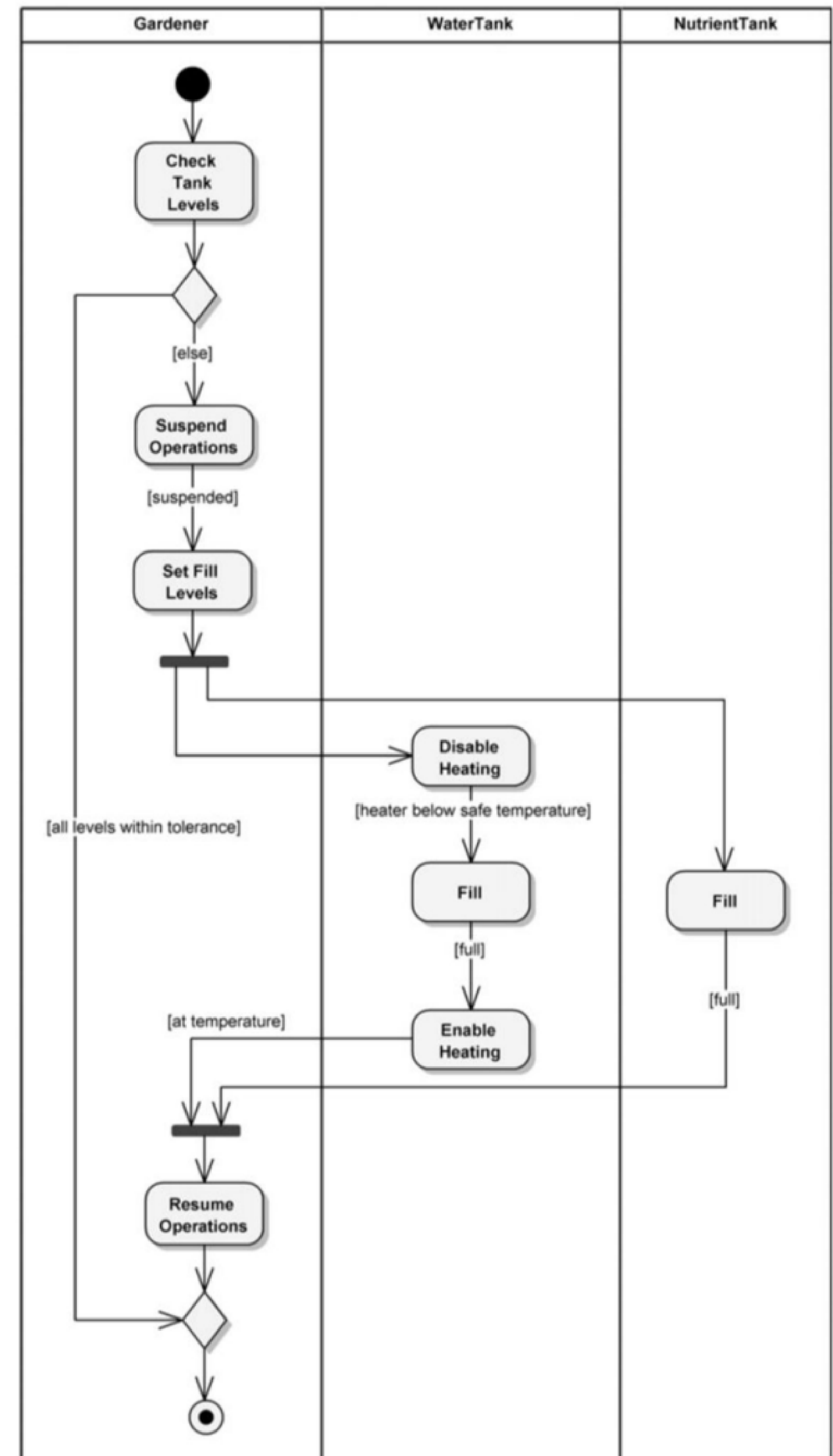
- 使用好的编码风格和合理的设计
- 不要仓促地编写代码、磨刀不误砍柴工！！！！
- 增加安全检查
- 写清晰的代码而非简洁的代码：
 - 例如：复杂的代数运算拆分为一系列简单的语句
- 审慎地处理内存和其他资源
 - 避免循环引用
- 在声明位置初始化所有变量
- 尽可能推迟一些声明变量

1.5 常用UML图

- 分析：用例图
- 设计：
 - 类图：class diagram
 - 活动图：activity diagram提供了活动流程的可视化描述
 - 序列图：sequence diagram
 - 状态机图：state machine diagram
 - 对象图（可选）object diagram

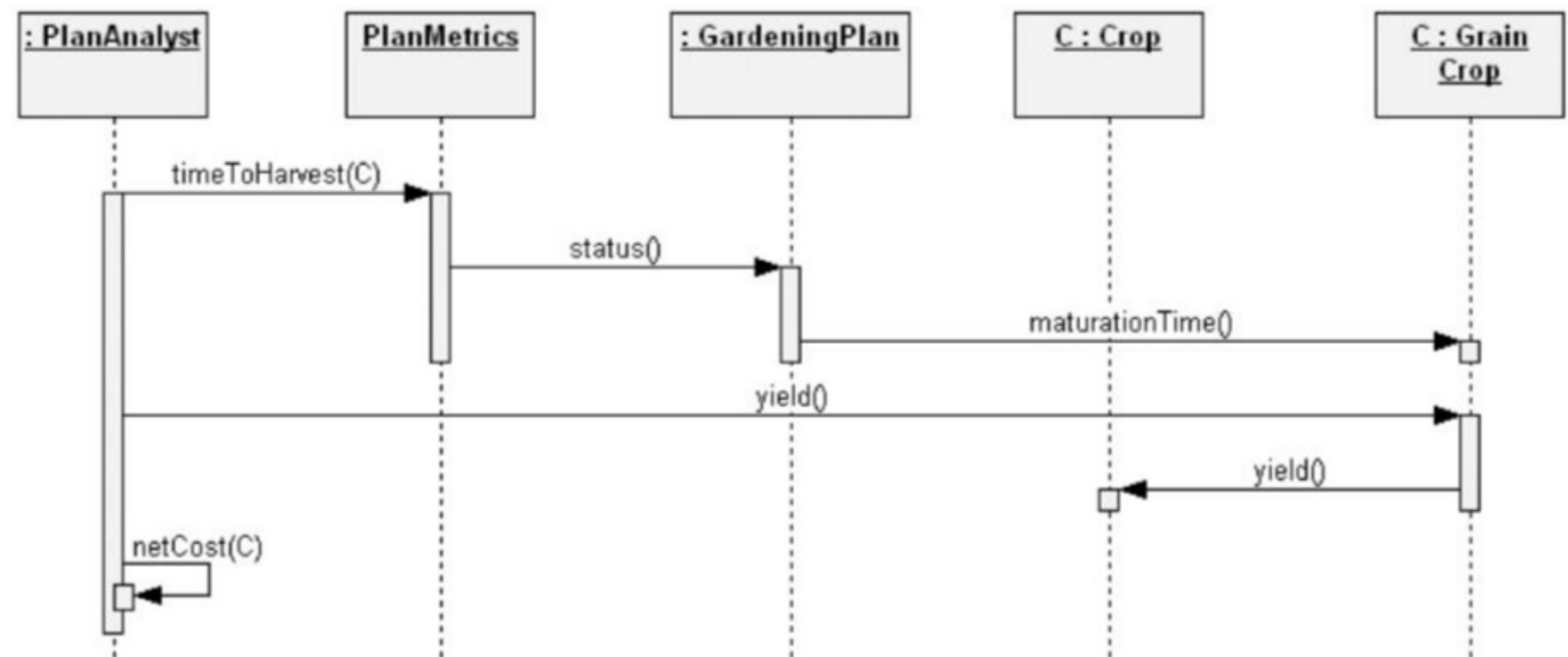
活动图

- 提供了活动流程的可视化描述，可以是在系统、业务、工作流或其他过程中。
- 关注被执行的活动以及谁（或什么）负责执行这些活动。
- 元素包括动作节点、控制节点和对象节点。
- 三种类型的控制节点：
 - 初始和终止（终止节点有两个变例：活动终止和流程终止）
 - 菱形：判断和合并（或）
 - 分叉和结合（与）
 - 粗横线使用：并发，结合



序列图

- 跟踪在同一个上下文环境中一个场景的执行
- 生命线与消息:同步异步
- 销毁事件
- 交互使用
- 控制结构



2 阅读代码

Mendix之RestServices

- REST service module for Mendix. Supports consuming and publishing REST based services and real-time data synchronization. Supports JSON, form-encoded, multipart and binary data transport. Mendix的REST服务模块。支持消费和发布基于REST的服务和实时数据同步。支持JSON、表格编码、多部分（或多段）和二进制数据传输。
- 文档链接：
 - <https://docs.mendix.com/refguide/integration/>
 - <https://docs.mendix.com/refguide/integration/rest-services/>
 - <https://docs.mendix.com/refguide/consumed-rest-services/>
 - <https://docs.mendix.com/refguide/published-rest-services/>
- 文件格式 <https://forum.mendix.com/p/questions/11540474045285303>
 - MPK: MPR + bundles files (JS, Java, Themes, data snapshot)
 - Can be used for exporting and sharing of full projects but also single Documents or Modules
 - MPR: The Mendix Model definition
 - MDA: Deployment package

RestServices代码结构

文件名	主要内容	文件格式
DIST	发布版本	.mpk mendix package包含一个Mendix项目或模块。
docs	文档	.md
images	图片	.png
javasource	communitycommons,restservices, system,tests,unittesting	.java
antisamy	用于防止Web应用程序中的跨站点脚本攻击（XSS）和其他安全漏洞	.xml，包名为resources.communitycommons
theme	主题风格	css,scss,json,ico,html

环境信息

- IntelliJ IDEA 2023.1.2 (Ultimate Edition) Build #IU-231.9011.34, built on May 16, 2023 Licensed to Shanghai XXXX Info Tech Co., Ltd. / XXXXXX Subscription is active until June 18, 2023.
- Runtime version: 17.0.6+10-b829.9 aarch64 VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o. macOS 13.4
- GC: G1 Young Generation, G1 Old Generation Memory: 4096M
- Cores: 8
- Metal Rendering is ON
- Non-Bundled Plugins:
 - plantuml-parser (0.0.5)
 - Statistic (4.2.7)
 - com.github.kisstkondoros.codemetrics (1.0.4)
 - com.caldonazzi.alessandro.uml (0.2.0)
 - com.alibaba.p3c.xenoamess (2.1.1.5x)
 - PlantUML integration (6.1.0-IJ2022.2)
 - MetricsReloaded (1.12)
 - SequenceDiagram (3.0.5)
 - com.kn.diagrams.generator.generator (2022.2.0)
- Kotlin: 231-1.8.21-IJ9011.34

阅读步骤

1. 寻找文档
2. 使用类图和序列图快速了解代码
3. 先读关键类再读辅助类
4. 检查代码复杂度和思考可能的改善方法

RestServices/Consume代码结构

类名	功能	公共方法
RestConsumer	消费REST服务的方法	readJsonObjectStream(String url, final Predicate<Object> onObject) registerCredentials(String urlBasePath, String username, String password) request(final IContext context, HttpMethod method, String url, final IMendixObject source, final IMendixObject target, final boolean asFormData) getResponseHeaderFromRequestResult(RequestResult requestResult, String headerName) getLastConsumeError(IContext context) startIdleConnectionMonitor(long interval, long maxIdleTime)
RestConsumeException	消费REST服务时的异常	getStatus() getResponseData() hasResponseData()
ChangeLogListener	变更日志侦听器	getChangesRequestUrl() follow() unfollow(String collectionUrl) getTrackingState(String collectionUrl) resetDataSyncState(String collectionUrl) fetch(String collectionUrl, String updateMicroflow, String deleteMicroflow)
IdleConnectionMonitorThread	闲置的连接监控线程	run() shutdown()

建议练习

- 用UML类图和序列图快速了解代码：
 - <https://plugins.jetbrains.com/plugin/7017-plantuml-integration>
 - <https://vanco.github.io/SequencePlugin/>
- 用代码复杂度工具检查代码复杂度
 - MetricsReloaded: <https://plugins.jetbrains.com/plugin/93-metricsreloaded>
 - CodeMetrics: <https://plugins.jetbrains.com/plugin/12159-codemetrics> 在程序中显示复杂度

总结

1 先导知识

- 1.1 面向对象程序设计的特点
- 1.2 REST
- 1.3 Java多线程
- 1.4 防御式编程
- 1.5 UML：对象图、类图、时序图

2 阅读RestConsume

- 应用面向对象程序设计方法
- 检查复杂度和思考可能的改善方法

3 建议学习内容

下一节，
阅读RestServices/Publish