面向对象分析方法

案例背景：开发一款手机银行APP，涉及用户场景如访问控制、转账、查询等。

面向对象分析（OOA）：按照面向对象的思想来分析问题、表达问题，并建立系统的形式模型。

面向对象: 面向对象是一种认识和模拟客观世界的方法，基本着眼点是构成客观世界的对象。

2. 面向对象的基本概念

抽象：抽取事物的共同、本质特征。

对象：现实世界中个体或事物的抽象表示，在计算机世界中是可标识的存储区域。

类：描述一组具有相同数据结构和操作的对象的集合。

封装：结合对象的属性和服务，隐蔽对象的内部细节。

继承：子类隐式使用超类的属性和操作。

多态性：隐藏接口不同实现的能力。

其他概念：信息/实现的隐藏、状态保持、对象标识、消息、一般性。

3. 面向对象分析的任务和优点

基本任务：使用面向对象方法分析问题域和系统责任，找出所需的对象，定义属性、操作及关系。

目标：建立符合问题域、满足用户需求的需求分析规格说明。

优点：加强对问题域的理解，改进交流，适应需求变化，支持软件复用，保持软件生命周期的一致性。

软件过程及过程模型

1. 软件过程和生命周期模型基础

- 软件过程：指生产软件的方式，涉及软件工程实践中的原则、概念、方法和工具。
- 软件生命周期模型：描述构建软件产品时应执行的步骤。

2. 过程模型与生命周期模型的区别

- 过程模型：以人为主体，从开发过程来看。
- 生命周期模型：从软件自身来看，侧重于软件产品的发展过程。

3. 生命周期模型的类型

- 瀑布模型（Waterfall）：线性顺序的方法，从需求到交付。
- 演化树模型（Evolutionary Tree）：展示了软件开发的迭代和增量特性。
- 增量模型（Incremental）：逐步构建软件产品的方法。

4. Winburg市公交系统案例

- 描述了一个公交系统软件开发和演化的案例，包括需求分析、设计、实现和维护等阶段。

5. 基线和迭代

- 基线：在每个开发阶段结束时，形成的一套完整的工件（artifact）。
- 迭代和增量：软件开发中的两个核心概念，迭代指重复相同的开发流程，增量指在产品功能或模块上的增加。

6. 核心工作流程

- 五个核心工作流程：需求工作流、分析工作流、设计工作流、实现工作流和测试工作流。

结构化分析方法

1. 结构化分析方法（SA）概述

- 需求分析：分析工作流的目标是分析和提取需求，以获得正确开发软件产品和易于维护它所必须的需求。
- 分析的位置：分析是从问题域向求解域迈进的第一步，需求流的输出必须能够完全被客户所理解。

2. 需求工程

- 需求的内涵/分类：包括客户需求、用户需求、变更、功能、速度和体验。
- 任务/活动：收集获取需求、现有系统调研、抽象目标系统的逻辑模型。

3. 结构化分析方法（SA）

- 基本思想：“自顶向下，逐步求精”和“抽象和分解”。
- 方法组成：结构化分析 + 结构化设计 + 结构化程序设计。

4. 结构化分析方法的核心

- 分解：将系统的复杂性降低到可以掌握的程度。
- 抽象：考虑问题最本质的属性，暂把细节略去。

5. 结构化分析方法的模型

- 系统关系图：系统与外界相关系统的输入与输出。
- 数据流程图：表达信息系统中数据的流向方式。
- 信息结构图：表达系统的顶层架构、体现分解的过程。
- 数据字典：定义数据，描述了所有的在目标系统中使用和生成的数据对象。
- 实体—关系图(ERD)：描述数据对象及数据对象之间的关系。
- 状态—迁移图(STD)：描述系统对外部事件如何响应，如何动作。
- 加工规格说明：定义数据的处理。

6. 数据流程图（DFD）

- 符号介绍：外部实体、数据存储、加工、数据流。
- 特点：可以表示任何一个系统中的数据流程，强调的是数据流程而不是控制流程。

7. 功能需求分析

- 加工说明：说明DFD中的每个加工。

4. 面向对象分析方法

Booch 方法、Rumbaugh 方法(OMT)、Jacobson 方法(OOSE)等，最终基于UML的OOA方法。

5. 面向对象分析过程

标识对象和类：识别系统保存和处理信息的能力。

标识结构：反映泛化-特化关系和整体-部分关系。

定义主题：事物的总体概貌和总体分析模型。

定义属性：描述对象或分类结构的实例。

定义服务：对象在收到消息后必须进行的处理方法。

6. OOA建模与表达

结构层：捕捉特定应用论域中的结构关系。

主题层：将对象归类到各个主题中。

7. OOA模型

功能（用例）模型：描述系统整体视图，表达详细需求。

对象（结构）模型：静态的、结构化的系统“数据”性质。

动态（行为）模型：描述系统的动态行为和对象之间的交互。

7. 通用过程框架

- 定义了五个框架活动：通信、计划、建模、构造和部署。
- 以及一系列在整个过程中应用的总体活动，如项目跟踪和控制、风险管理、质量保证、配置管理和技术审查。

8. 任务集和软件工程行动

- 任务集：定义了为实现软件工程行动目标而要完成的实际工作。
- 软件工程行动：涉及的具体任务，如需求收集、设计、编码等。

9. 规定性生命周期模型

- 代码和修复模型（Code-and-Fix）：无需求或设计，直接编写代码并多次修改以满足客户。
- 瀑布模型（Waterfall）：系统化、顺序的方法，从需求到部署。
- 快速原型模型（Rapid-Prototyping）：首先构建一个快速原型，让客户和用户与之交互，然后根据反馈制定规范文档。
- 螺旋模型（Spiral）：结合了原型和瀑布模型，通过风险分析来最小化风险。
- 描述工具：结构化语言、判定表、判定树。

8. 数据需求分析

- 数据字典（DD）：DFD中所有元素的定义的集合。
- 内容：数据流、数据流分量、数据存储。

9. 信息结构

- 分层数据结构表示法：分层框图、Warnier图。
- 实体关系图（E-R图）：用于对复杂数据的数据分析和建模。

10. 状态迁移图（STD）

- 描述：软件状态变迁及其触发事件。
- 符号表示：矩形代表系统状态，箭头代表状态转变方向，规则表达式代表事件/触发行为。

11. 结构化分析过程

- 获取软件需求：运用抽象和分解的技术，提供一组经验和规则以指导需求分析。
- 分层建立数据流图：完整、详尽、一致，用数据字典、说明作为补充。
- 描述需求
- 验证需求。

12. 结构化分析辅助工具

- MS VISIO、Statemate：建模、模型的存储、显示和检索，模型之间、数据条目之间的一致性检查。