

# 软件需求工程

## 课后作业 5

徐鸣飞

# 目录

<b>第一章 开发用例</b>	<b>1</b>
1.1 使用在线书店购书 . . . . .	1
<b>第二章 需求定义</b>	<b>3</b>
2.1 对于需求分析，结构化分析和面对对象方法有何本质区别 . . . . .	3
2.2 状态图和顺序图有何异同 . . . . .	4
2.3 需求规格说明（SRS）有哪些作用 . . . . .	4
2.4 需求规格说明必须满足哪些特性 . . . . .	5
2.5 用自然语言描述需求时应注意些什么 . . . . .	5

# 第一章 开发用例

## 1.1 使用在线书店购书

**用例名称：**在线购书

**用例简要描述：**这个用例描述了用户如何通过在线书店购物平台购买图书。用户可以浏览图书，将图书加入购物车，然后进行结算。

**与用例有关的执行者：**

1. 普通用户：他们可以浏览图书，将图书加入购物车，进行结算。
2. 系统管理员：他们可以管理库存，处理用户的订单，管理用户信息等。

**前置条件：**

1. 用户已登录到在线书店平台。
2. 所需的图书在库存中可用。
3. 用户的账户余额足够支付图书费用。

**交互动作过程：**

1. 用户浏览图书列表，查看每本图书的详细信息。
2. 用户选择他们想要购买的图书，并将其添加到购物车。
3. 用户在购物车中查看已添加的图书，并可以进行编辑或删除操作。
4. 用户点击结算按钮，进入结算页面。

5. 用户输入他们的地址和支付信息，然后点击提交按钮进行支付。
6. 系统处理订单，如果支付成功，则将订单状态更新为已发货，并将图书从库存中移除。
7. 系统向用户发送订单确认邮件，包含订单的详细信息。
8. 用户可以在他们的账户页面查看已完成的订单。

**后置条件：**

1. 用户的订单已成功支付 (用户的账户余额已扣除相应的费用。)
2. 图书已从库存中移除。
3. 图书已发货。

## 第二章 需求定义

### 2.1 对于需求分析，结构化分析和面对对象方法有何本质区别

**结构化分析：**一种自顶向下的分析方法，它通过将问题分解为较小的、易于处理的模块来解决问题。它强调功能的分解和模块化设计，以数据流为中心，通过数据流图、加工说明和数据字典等工具进行系统分析。结构化方法适用于数据处理等业务领域，它注重功能和数据处理流程的描述，使得开发出的系统具有较好的可读性和可维护性。

**面对对象方法：**一种以对象为基础的软件开发方法，它将数据和操作封装在对象中，通过对象之间的关系来描述现实世界中的概念和概念之间的关系。它强调抽象、封装、继承和多态等特性，通过将问题领域中的对象抽象为软件中的类和对象，并定义类之间的关系，来实现对问题的建模和解决。面向对象方法适用于管理信息系统等业务领域，它注重对现实世界中概念的抽象和模拟，使得开发出的系统具有较强的灵活性和可重用性，能够更好地适应需求变化。

**区别：**结构化分析和面向对象方法在本质上的区别在于它们对问题的处理方式和思维模式。结构化分析方法注重功能的分解和模块化设计，以数据流为中心，通过自顶向下的方式逐步细化问题，最终实现问题的解决。而面向对象方法则以对象为基础，将数据和操作封装在对象中，通过对象之间的关系来描述现实世界中的概念和概念之间的关系，强调抽象、封装、继承和多态等特性，通过将问题领域中的对象抽象为软件中的类和对象，并定义类之间的关系，来实现对问题的建模和解决。因此，结构化分析方法更适用于数据处理等业务领域，而面向对象方法更适用于管理信息系统等业务领域。

## 2.2 状态图和顺序图有何异同

UML 中的状态图和顺序图都是用于描述系统行为的工具，但它们关注的焦点和描述的内容有所不同。

状态图是一种用于描述对象状态及状态之间的转移的工具。状态图强调的是**对象在其生命周期中的行为状态变化**，以及如何通过事件触发状态之间的转移。它可以帮助我们理解对象的行为，以及对象在生命周期中的状态变化。

顺序图则是一种强调对象间消息传递次序的交互图，也称为时序图或序列图。顺序图关注的是**对象之间的交互关系和消息传递的顺序**。它可以帮助我们理解对象之间的合作关系，以及对象之间的消息传递过程。

## 2.3 需求规格说明（SRS）有哪些作用

需求规格说明（SRS）的作用主要包括以下几点：

1. **描述软件系统的需求规格和特征：**SRS 文档包含了软件产品的所有需求规格和特征，包括功能需求、性能要求、数据要求、界面要求、可靠性要求等。
2. **提供开发、测试和维护的依据：**通过详细描述软件系统的需求，SRS 文档为开发团队提供了明确的方向和指导，同时也为测试团队提供了测试的依据，确保软件系统的质量和性能。
3. **提高开发效率：**一个清晰、准确的 SRS 文档可以帮助开发团队避免返工和重复工作，从而提高开发效率。
4. **减少后续修改的风险：**通过在 SRS 文档中详细描述软件系统的需求，可以提前发现和避免一些潜在的问题，从而减少后续修改的风险。
5. **为项目团队提供明确的目标和约束：**SRS 文档为项目团队提供了明确的目标和约束，包括交付标准、功能、性能和可靠性等级等，有助于项目团队更好地协同工作。
6. **为客户和利益相关者提供清晰的沟通：**通过编写 SRS 文档，客户和利益相关者可以更好地理解软件系统的需求和功能，从而更好地评估项目的可行性和价值。

## 2.4 需求规格说明必须满足哪些特性

1. 正确性：需求规格说明必须正确地描述软件系统的需求，符合用户的期望和要求。
2. 无歧义性：需求规格说明的描述必须清晰、明确，避免产生歧义或误解。
3. 完整性：需求规格说明必须包含软件系统的所有必要需求，不能遗漏任何关键信息。
4. 可验证性：需求规格说明中的每个需求都必须是可以验证的，即可以通过一定的手段进行验证和确认。
5. 一致性：需求规格说明中各个需求的描述必须一致，不能存在矛盾或冲突。
6. 可理解性：需求规格说明的描述必须简明易懂，避免使用过于专业或复杂的术语，以使用户和软件人员都能接受。
7. 可修改性：需求规格说明的结构风格必须在需求有必要改变时易于实现，以便进行修改和更新。
8. 可追踪性：需求规格说明中的每个需求都必须是可以追踪的，可以追踪到源头上，以便进行需求变更管理。

## 2.5 用自然语言描述需求时应注意些什么

1. 清晰明了：需求描述应该简洁、清晰，避免使用模糊词汇，以确保所有人都理解需求的具体含义。
2. 具体详细：需求描述应该具体、详细，包括具体的功能、操作流程、输入与输出、异常情况处理等，以便开发人员能够准确地实现每个需求。
3. 不涉及实现细节：需求描述应该只关注需求本身，不要涉及具体的实现细节，否则可能会限制开发人员的实现方式，影响开发效率和灵活性。
4. 避免主观性描述：需求描述应该尽可能避免主观性描述，如“用户友好”、“快速响应”等，而应该给出具体的客观标准，以便评估和衡量开发成果。

5. 考虑可测试性：需求描述应该考虑可测试性，确保每个需求都可以通过一定的测试方法进行验证和确认，以便保证软件系统的质量和性能。
6. 避免二义性：需求描述应该避免存在二义性，即每个需求都应该有一个明确的答案，避免开发人员有不同的解读和理解。