



南京大学

复习和总结

Object-Oriented Design: Review and Conclusion Remarks

Review

- OO基础(30')
 - 从结构化到面向对象
 - 抽象数据类型
 - 契约式设计 with 异常
- OO设计原则(10')
- OO设计模式(60')

OO基础

- 软件质量
 - 外部质量因素(正确性，鲁棒性，完整性...)
- 应对软件复杂性的“元”方法
 - 分解(分而治之)，抽象(抓本质，抓重点)，层次化(应对大系统，纲举目张)
- 软件模块
 - 高内聚、低耦合
- 软件复用
 - 传统技术 vs. OO技术

抽象数据类型

- 过程抽象 vs. 数据抽象
 - 过程抽象：指任何一个明确定义功能的操作都可以被使用者看作单个的实体，尽管这个操作实际上可能由一系列更低级的操作完成
 - 数据抽象：定义了数据类型和施加于该类型对象上的操作，并限定了对象的值只能通过使用这些操作修改和观察。包含了2个概念：模块封装和信息隐蔽
- 抽象数据类型：用**数学方法**定义对象集合和运算集合，仅通过运算的性质刻画数据对象，而独立于计算机中可能的表示方法
- ADT规约方法：代数方法

抽象数据类型

- 语法部分

- **ADT**名

- 运算(函数)的定义域和值域

- 公理部分

- 给出一组刻画各运算之间相互关系的方程来定义各运算的含义
 - 语义正确性：相应代数满足规约中公理部分的所有公理。

ADT specification of stacks

TYPES

- $STACK [G]$

FUNCTIONS

- $put: STACK [G] \times G \rightarrow STACK [G]$
- $remove: STACK [G] \rightarrow STACK [G]$
- $item: STACK [G] \rightarrow G$
- $empty: STACK [G] \rightarrow BOOLEAN$
- $new: STACK [G]$

AXIOMS

For any $x: G, s: STACK [G]$

- A1 • $item (put (s, x)) = x$
- A2 • $remove (put (s, x)) = s$
- A3 • $empty (new)$
- A4 • $not\ empty (put (s, x))$

PRECONDITIONS

- $remove (s: STACK [G])$ require not empty (s)
- $item (s: STACK [G])$ require not empty (s)

- 基本思想
 - 借鉴“契约”原理，界定模块之间的权利义务，规范软件的开发，提高软件质量。
- 具体措施
 - **Preconditions, Postconditions, Invariants**
- 契约如何应用于继承？
- DbC和防御性编程的比较

- “异常” 的分类
 - Checked exception; unchecked exception; error
- 对 “异常” 的理解
 - 依据**DbC**，对**Java**异常的分类的理解
 - **Java**子类若重定义父类中的方法，其**throws**的异常有何限制

	权利	义务
Client		知晓可能发现的异常，并作相应处理
Supplier	允许抛出一定的异常	

Review

- OO基础
 - 从结构化到面向对象
 - 抽象数据类型
 - 契约式设计 with 异常
- OO设计原则
- OO设计模式

- 各个原则的含义
 - SRP：单一职责原则
 - OCP：开放-封闭原则
 - LSP：Liskov替换原则
 - DIP：依赖倒置原则
 - ISP：接口隔离原则
 - CARP：合成/聚合复用原则
 - LoD：迪米特法则
- 应用
 - 违反原则
 - 改正

Review

- OO基础
 - 从结构化到面向对象
 - 抽象数据类型
 - 契约式设计 with 异常
- OO设计原则
- OO设计模式

- GoF设计模式
 - 名，目的
 - 问题 - 解决方案(典型实现)
 - 与其它模式的关系
 - 考虑例子!
- 分类
 - 创建型
 - 结构型
 - 行为型

OO设计模式

- 重点

- Singleton：多线程安全的方案；如何实现多例模式
- Prototype：深浅拷贝实现方式
- Adapter：类适配，对象适配
- Proxy：静态代理，动态代理以及其它几种常见的代理模式
- Command：命令对列，请求日志，撤销和恢复操作
- Iterator：内部类实现方式
- Observer：观察者模式与MVC之间的关系
- Template Method：反向控制结构
- Visitor：双分派

- 时间：2021年6月22日8:00-10:00
- 地点：待定
- 方式：闭卷