

1. 一个对象所持的链也是对象性质。

2. 类要概括 { 已出现对象
将未出现的对象。

3. 对象 → 操作的对象。

4. 面向对象特性：面向对象核心思想

(1) 封装性。

(i) 对象

(ii) 类：视图：成员特征、性质、静态与否。

(iii) 包。

(2) 继承性

(3) 多态性

(i) 类型多态性

(ii) 行为多态性。

5. 运行时刻：对象状态 → 一组性质的值。

6. 一个类中多个特征次序无关

7. 子类不能直接访问超类 private。
但确实继承了。

8. UML 多继承 V. 一个接口 → 多个实现类。
一侯 → 多个接口。

9. 一个类的超类型 { 超类

10. 模型：局部抽象、目的特定。

11. 好模型：规范性、正确性、一致性、完备性。

12. UML 新图：包、复合结构、交互纵览、计时。

13. 用例 { 一组动作的规范

产生可观察结果

特殊用途和价值

14. 类、用例、类、关联

实例 一次执行 对象 链
(场景)

15. 语境：存在于系统外部与系统交互的事物和对象。

16. (子) → (父) 抽。

(大) - - -> (小) 抽
include

(大) ← - - (N) 抽
extend

17. 类图 → 静态结构

对象图 → 运行时刻。

序列图 消息交互

活动图 行业务操作流程。

状态图 生命周期、状态变化、解除执行。

18. UML 基类型：String, Boolean, Integer, Unlimited Natural.

19. 需 供 可作为连接器。
连接器 实现

20. 链：云组。

21. 关联：非定向关系，但可说明导向性。

22. 团体 群 个人。

23. 包的关系：包含、导入、合并。
包
import 公有导入
access 私有导入
对外可见

24. 复合结构图：内部结构、端口、协作。

25. 端口：可用时有供口需口。

26. 交互图包括序列、通信、交互纵览、计时。

27. 控制焦点 = 执行规范 (→) (←)

发生规范 (→)

28. 装图连接件的定义：从一个需口到一个供口。

1. 对象

边界和标识.
状态和行为的封装体
类的一个实例.

2. 一个对象持有的链也是该对象的性质.

3. 面向对象基本特征: 封装性, 继承性, 多态性.

4. 类的封装视图: 可见性, 静态成员特征.

5. 信息隐藏 { 设计决策局部化
优点 } 减少信息内容

6. 封装性优点 { 信息隐藏
面向对象核心思想 } 状态保持

7. 模型的一般作用: 可视化形象直观工程价值.

8. 模型特点: 局部性, 抽象, 特定目的.

9. 建模目的: 规范化设计, 可视化表达, 构建存档.

10. 建模三要素: 对象, 规范方法.

11. 好模型: 规范性, 正确性, 一致性, 完备性.

12. UML2新图: 包, 类, 对象, 类图, 类图, 类图.

结构图: 包, 类, 对象, 类图, 类图, 类图.

行为图: 用例图

交互图: 序列, 通信, 计时, 交互.

状态机图

活动图

13. 用例

系统所执行的一组动作的规范

执行产生可观察结果

对参与者产生特殊用途和价值.

14. 同一个人可以是多个参与者.

角色

15. 类

实例

用例

场景

类

对象

接口

实现类对象

关联

链 (数组)

16. 性质

单值

多值 -> 聚集

17. 对象图: 实例规范, 值规范.

{ 性质 -> 结构规范

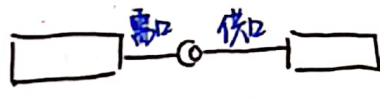
操作 -> 行为规范.

18. 开参方向

{ in 只能读无法改变

out 可改变.

19.



20.

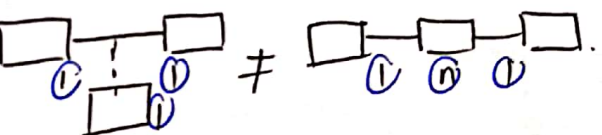
{ 约束 }



21. 关联属于非定向关系, 但可说明导向性.

聚集 { 共享式 班级没人还在. 看大的没3.11的在不在
复合式 要死一起死 (独占) -> 一个对象被多个包含

22.

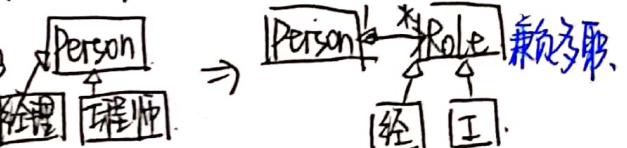


23. 限定关联



24. 泛化用途: 概念表述, 特征重用, 抽象设计.

25.

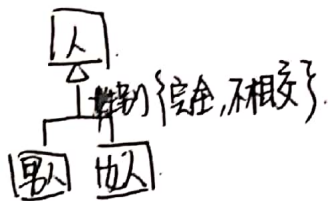


26.

关系环 { 泛化

关联.

1. 集合 { in/complete
is disjoint/overlapping



2. 依赖 { 使用=调用/创建/实例化
抽象=实现

3. 标记值 tag=value.
构造型: «JavaBean».

4. 包的关系 { 包含→封装
导入→import
合并→组合
访问→access 对外可见.

5. 复合结构图: 内部结构. 端口. 协作.
端口可同时有供、需口.

6. 协作→模板.

7. 交互标题以 sd 开头 sequence diagram

生命线→实体/实体集合.

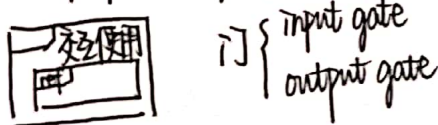
消息可以有倾斜→延时.

→ { 同步调用, 应答
--> { 异步调用 异步信号.

创建: --> <create> <delete> 删
lost found

发生规范→点. 执行规范→线.

组合片断: alt. opt. par. loop.



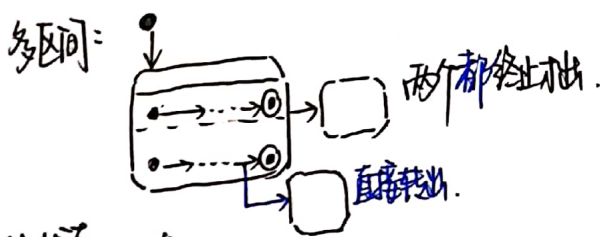
门 { input gate
output gate

8. 状态图 事件[条件]/行为.

状态 { 满足某些条件
执行特定动作
等待某些事件发生.

事件 { 调用事件 接收者被调用.
改变 轮流某布表达式
信号 接收信号
时间.

状态内部: 入口行为出口. 状态活动. (避免事件).



由状态: 始态.

分叉 水

汇合 斗 都到齐枝

接合 叉 [事件]

选择 叉

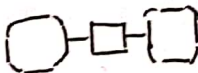
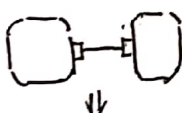
探测栈历史.

入口点&出口点

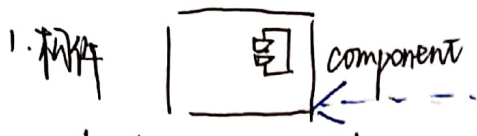
终结+终态.

终态⊙不是由状态.

9. 活动图 { 动作结束
结束 { 控制结束
调用动作
发送信号动作
接收事件动作.

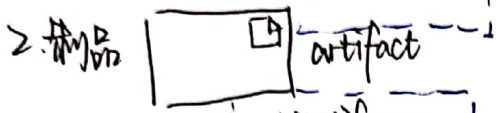


起始.
分叉与汇合 水 -- 斗
活动终止⊙ 一个到整个活动终止.
流终止⊗
活动权重 { weight=? }



- ≠ subsystem 较大规模
 - entity 持久化存储
 - process 处理事务
 - service 无状态的服务函数
- 特性: 基于接口, 自包含, 可替换.

- 构件: 物理 & 逻辑性构件.
- 类: 纯逻辑性.



软件制品可部署在结点上

- 设备 device
- 执行环境 executionEnvironment

3. 表示一个操作时, 至少应表示其基固.

4. 性质类型初始为 String