

识别类的方法

- 根据用例描述中的名词确定类的候选者

- 使用CRC分析法寻找类。

CRC是类（Class）、责任（Responsibility）和协作（Collaboration）的简称，CRC分析法根据类所要扮演的职责来确定类

- 根据边界类、控制类和实体类的划分来帮助发现系统中的类
- 对领域进行分析，或利用已有的领域分析结果得到类
- 参考分析、设计模式来确定类



我们先来玩个游戏 😊

- 我们要编写一个21点游戏的程序
- 客户交给你一份详细的问题描述，也就是21点游戏规则说明
- 下面我们开始设计这个系统
- 包括识别类、类的职责、类之间的交互行为



21点游戏（Blackjack）

- 21点游戏的目标是获得点数最接近或等于21 的手牌
- 胜者需要赢过庄家的手牌，而且不能超过21 点
- 无论桌上有几名玩家，总是只和庄家论输赢
- 21点游戏中，2-10的牌其点数就是牌面的数值
- A的点数根据玩家需要可以取1或11
- J，Q，K的点数都是10



21点游戏（Blackjack）

- 游戏开始时，玩家首先下注，通常下注的额度是有上限的
- 如果你是唯一玩家，下注后，庄家给你发两张牌，牌面向上，双方均能看到
- 庄家给自己发两张牌，一张牌面向上（明牌），一张牌面向下（底牌）
- 如果有多个玩家，则庄家给每位玩家都是先发一张明牌，庄家自己发一张底牌，再给每个玩家和庄家各发一张明牌



21点游戏（Blackjack）

- 发完两张牌之后，每位玩家自己决定是否要更多的牌，当对手上的牌数满意之后，就停手等待
- 停手之后，庄家开始加牌，不能超过21点
- 当你的手牌点数超过庄家时，你赢，当你的手牌点数不及庄家时，庄家赢
- 玩家赢：
 - 你的手牌点数超过庄家，但不超过21点
 - 庄家冒过21点
- 玩家和庄家点数持平，则不输不赢，庄家将玩家赌注归还
- 玩家输：
 - 玩家冒过 21点



CRC卡片

	Class: 类定义		Classname 类名	
	Responsibilities 类职责定义	Collaborations 交互协作关系定义		



识别21点游戏中的对象类

类对象通常对应一个命名实体，因此，用名词表达，于是我们先从名词开始

切记：别想一步到位



对象几乎无处不在

Source: Adapted from Pressman, 1994, p242

- 外部实体
 - 与建模中的系统存在交互人，设备和其他系统
- 事物
 - 建模的应用领域中存在的事物
报表，信号，文字输出
- 事件
 - 系统上下文中发生
资源传递，控制命令发出
- 角色
 - 由与系统交互的人扮演的角色
- 组织单元
 - 与应用领域相关的部分
分支机构、群组，团队
- 位置、地点
 - 建模中的问题的物理上下文
厂房、车间、货架
- 结构体
 - 定义类或者对象组合
 - 传感器、车辆、计算机

不能定义为对象的事物：

- 过程（打印、转换）
- 属性（兰颜色， 50Mb）



21点游戏（Blackjack）

- 21点游戏的目标是获得点数最接近或等于21的**手牌**
- 胜者需要赢过**庄家**的手牌，而且不能超过21点
- 无论桌上有几名**玩家**，你总是只和庄家论输赢
- 21点游戏中，2-10的**牌**其**点数**就是牌面的数值
- **A**的点数根据玩家需要可以取1或11
- **J, Q, K**的点数都是10



21点游戏（Blackjack）

- 游戏开始时，玩家首先下**注**，通常下注的额度是有**上限**的
- 如果你是唯一玩家，下注后，庄家给你发两张牌，牌面向上，双方均能看到
- 庄家给自己发两张牌，一张牌面向上（**明牌**），一张牌面向下（**底牌**）
- 如果有多个玩家，则庄家给每位玩家都是先发一张明牌，庄家自己发一张底牌，再给每个玩家和庄家各发一张明牌



21点游戏（Blackjack）

- 发完两张牌之后，每位玩家自己决定是否要更多的牌，当对手上的牌数满意之后，就停手等待
- 停手之后，庄家开始加牌，不能超过21点
- 当你的手牌点数超过庄家时，你赢，当你的手牌点数不及庄家时，庄家赢
- 玩家赢：
 - 你的手牌点数超过庄家，但不超过21点
 - 庄家冒过21点
- 玩家和庄家点数持平，则不输不赢，庄家将玩家赌注归还
- 玩家输：
 - 玩家冒过 21点



通过名词过滤识别出的对象类

- 游戏
- 21点
- 庄家
- 玩家
- 牌
- 牌堆
- 手牌
- 点数
- 花色
- 赢家
- A
- 数字牌
- K
- Q
- J
- 赌注



类筛选

- 在候选类中排除以下类：
 - 超出问题关注的范围的类；
 - 指代整个系统的类；
 - 功能重复的类；
 - 过于含糊或过于具体的类
 - 可观察到的现象是，实例对象过多过少
- Coad & Yourdon's 的筛选原则：
 - **保存对象信息**：系统需要保存对象信息吗？
 - **提供所需服务**：类对象是否对外提供修改属性值的操作？
 - **具有多个属性**：只有一个属性的类，应该建模为属性
 - **具有公共属性**：类属性是否为所有实例对象共享？
 - **具有公共操作**：类操作是否为所有实例对象共享？
 - **外部实体**：如果生产或使用对象的信息，也应考虑建模为系统类



由此识别出的对象

- 游戏
- 21点
- 庄家
- 玩家
- 牌
- 牌堆
- 手牌

- 点数
- 花色
- 赢家
- A
- 数字牌
- K
- Q
- J
- 赌注



选出的类

- 牌、牌叠、手牌、庄家、玩家、赌注

Card

Deck

Hand

Dealer

Player

Bet



类识别

- 从原始资料中识别类：
 - 找出干系人提交的问题描述中名词及短语
 - 如果他描述应用领域中的信息结构或本质，则加入模型
- 从其他来源识别类：
 - 背景信息调查
 - 用户及干系人提供
 - 分析模式
- 最好识别出尽可能多的候选类
 - 之后逐步按照其价值功用进行选择
 - 明确判断后排除一个类要比压根不考虑来的合理



识别类的功能职责

功能职责关乎行为动作，因此是问题描述中的动词...

注意：

1. 并非所有动词均将成为类职责
2. 有时多个动作合并为一个职责
3. 随着分析过程深入会发现新的职责
4. 不断修正类定义和职责定义
5. 当两个类分享职责时，为二者同时添加该职责



21点游戏（Blackjack）

- 游戏开始时，玩家首先下注，通常下注的额度是有上限的
- 如果你是唯一玩家，下注后，庄家给你发两张牌，牌面向上，双方均能看到
- 庄家给自己发两张牌，一张牌面向上（明牌），一张牌面向下（底牌）
- 如果有多个玩家，则庄家给每位玩家都是先发一张明牌，庄家自己发一张底牌，再给每个玩家和庄家各发一张明牌



21点游戏（Blackjack）

- 发完两张牌之后，每位玩家自己**决定**是否**要**更多的**牌**，当对手上的牌**点数满意**之后，就停手等待
- 停手之后，庄家开始**加牌**，不能超过21点
- 当你的手牌点数超过庄家时，你赢，当你的手牌点数不及庄家时，庄家赢
- 玩家赢：
 - 你的手牌**点数超过**庄家，切不超过21点
 - 庄家冒过21点
- 玩家和庄家点数持平，则不输不赢，庄家将玩家赌注**归还**
- 玩家输：
 - 玩家冒过 21点



候选类对应的功能职责

- 牌 (Card)
 - 读取牌面数字
 - 读取花色
 - 读取点数
 - 判断是否数字牌
 - 判断是否为A
 - 判断是否为 J、Q、K
- 赌注 (Bet)
 - 赌注类型
 - 赌注的大小
 - 剩余赌资
 - 判断赌资是否够下注
- 牌叠 (Deck)
 - 洗牌
 - 发牌
 - 查询剩余牌数
 - 是否能够新开一叠牌
- 手牌 (Hand)
 - 读取手中牌数
 - 读取手牌总点数
 - 亮牌



候选类对应的功能职责

- 庄家

- 发牌
- 洗牌
- 发一张牌给玩家
- 庄家亮牌
- 计算庄家手牌点数
- 查询庄家手牌点数
- 加一张牌
- 确定赢家
- 开始下一轮游戏

- 玩家

- 请求发牌
- 玩家亮牌
- 计算玩家手牌点数
- 查询玩家手牌点数
- 查看手牌点数是否超过21点
- 查看手牌点数是否等于21点
- 查看手牌点数是否低于21点



识别类交互协作关系：用UML用例图

识别对象及其消息交互

注意： 目的并非写出所有场景，而是对类和职责定义进行精化 ...



情景举例

庄家

- 庄家洗牌 ✓
- 庄家发第一张牌 ✓
- 庄家给自己发牌 ✓
- 手牌返回庄家手牌点数 ✓
- 庄家发第二张牌 ✓
- 庄家询问玩家是否继续要牌 ✓
- 庄家询问玩家手牌点数 ✓
- 庄家亮牌 ✓
- 庄家分配赌金 ✓

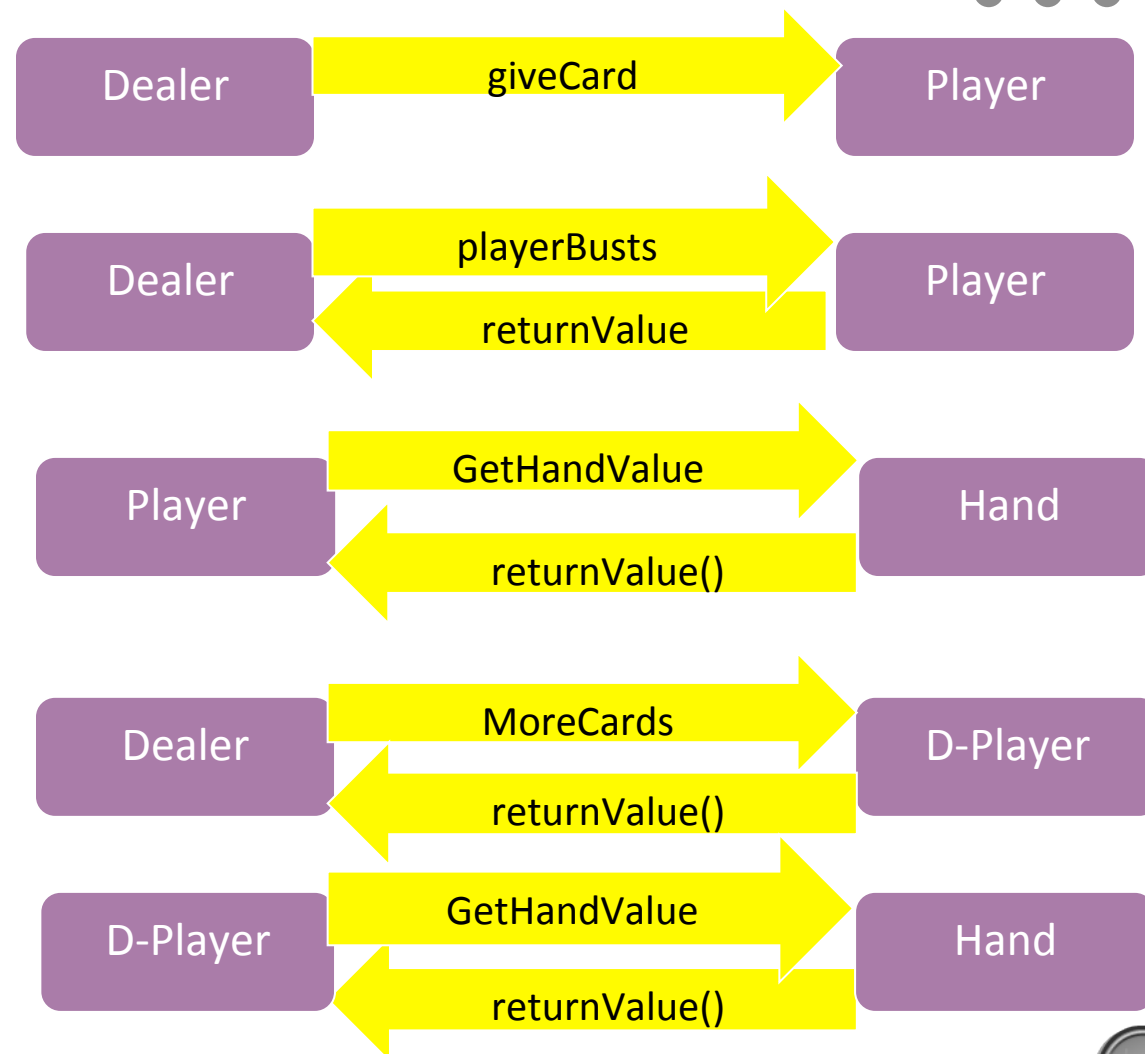
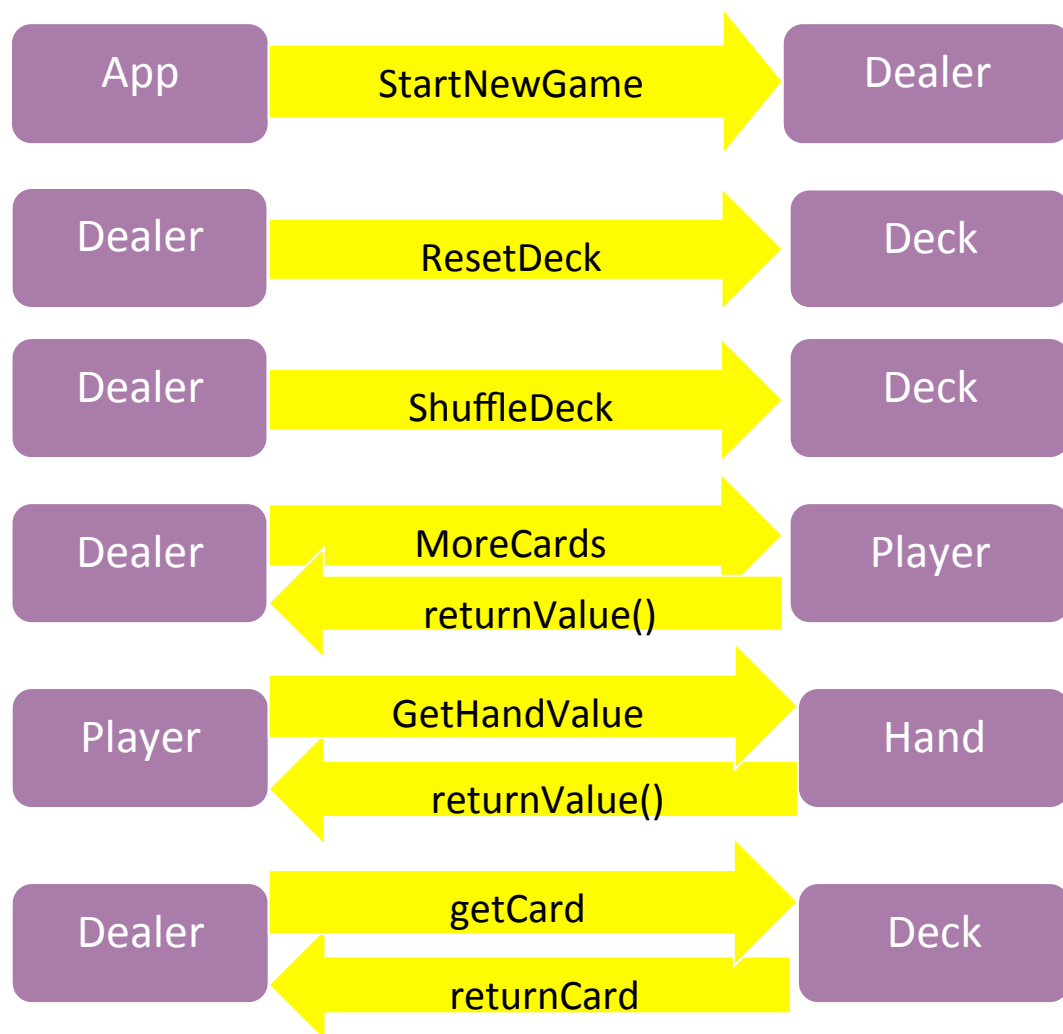


玩家

- 玩家下注 ✓
- 玩家拿牌到手里 ✓
- 手牌返回玩家手牌点数 ✓
- 玩家继续要牌 ✓
- 玩家拿牌到手里 ✓
- 手牌返回玩家手牌点数 ✓
- 玩家增加或减少赌注 ✓
- 玩家亮牌 ✓



识别交互关系





最终获得的CRC卡片



类定义：纸牌类（Card）

类名 (Class) 纸牌 Card		
	功能职责 (Responsibilities)	交互类 (Collaborations)
	GetName	Deck
	GetValue	



类定义：牌叠类（Deck）

类名（Class）

Deck

功能职责(Responsibilities)

Reset deck

Get deck size

Get next card

Shuffle Deck

Show deck

交互类（Collaborations）

Dealer

Card



类定义：庄家 (Dealer)

类名 (Class)

Dealer

功能职责 (Responsibilities)

Start a new game

Get a card

交互类 (Collaborations)

Hand

Player

Deck



类定义：玩家（Player）

类名（Class）		Player	
	功能职责(Responsibilities)		交互类（Collaborations）
	Want more cards		Dealer
	Get a card		Hand
	Show hand		
	Get value of hand		



类定义：手牌类 (Hand)

类名 (Class)		Hand	
功能职责 (Responsibilities)		交互类 (Collaborations)	
Return Value		Dealer	
Add a card		Player	
Show hand			



UML 类图

class system

