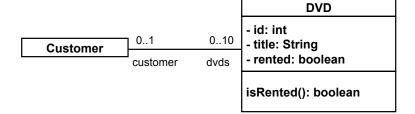
Modeling Exercise: DVD Rental

- Imagine a classical DVD rental shop.
- Each customer can rent up to 10 DVDs.

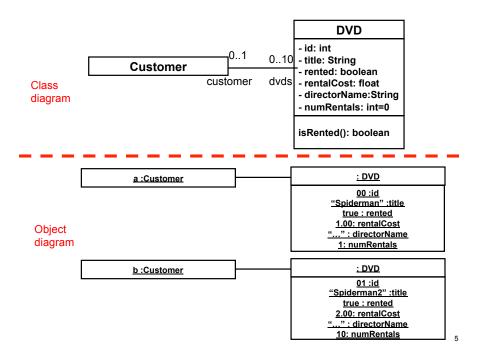
- DVD rental
 - Each customer can rent up to 10 DVDs.



DVD - id: int Class 0..10 title: String Customer diagram - rented: boolean customer dvds isRented(): boolean : DVD a:Customer 00 :id "Spiderman" :title Object true: rented diagram : DVD 10 :id "Spiderman2" :title true : rented : DVD b:Customer 01 :id "Spiderman" :title true : rented

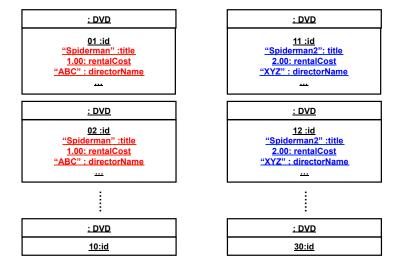
What if...

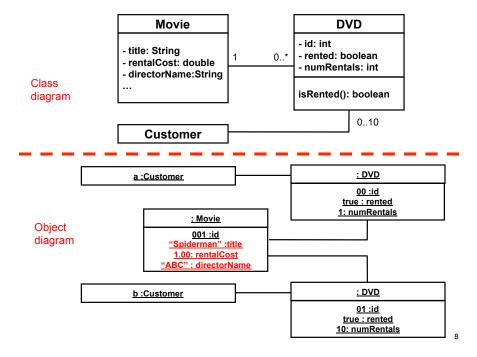
- you need to maintain individual DVD's properties?
 - Rental fee, total # of rentals in the past, production company, distribution company, distribution year, director's name, rating (Rrated, PG13, etc.), etc., etc.



- Why is redundancy bad?
 - It leads to a fragile design for changes.
 - A single change can impact many places in a system.
 - What if rentalCost changes...
 - Those places may fail to be in sync as software size/complexity increases.
- · How can we remove the redundancy?
 - Localize, modularize or minimize the impacts of a change
- Separate things from their types/kinds.
 - Separate individual/physical DVD media (containers) and movies (contents).

- How can we remove redundancy?
 - The total # of instances = # of physical DVD media = 30
 - 6 * 30 properties in total.
 - 10 instances share the same set of property values, except for ID.
 - 20 instances share the same set of property values, except for ID.





Exercise: Insurance Products

- Each customer (insured) can buy multiple insurance products.
 - e.g., home owner's/renter's insurance, automotive insurance, boat insurance, life insurance...
 - The properties of each insurance product:
 - Product name, insurer's name, deductible, annual premium the amount of year-to-date payments, policy number, agent's name, policy term (e.g., 09/01/07 to 08/31/08)
 - Let's take an insurance agency's viewpoint.

Insured productName: String - products insured - insurerName: String - policyNum: int deductible: float Class - annualPremium: float diagram - vtdPavment: float agentName: String : InsuranceProduct a: Insured "AutomotiveXYZ" : productName "XYZ": insurerName 001: policyNum 500: deductible Object 1000: annualPremium diagram "ABC": agentName : InsuranceProduct b: Insured "AutomotiveXYZ" :productName

Two instances share the same set of properties.

Think of separating products from their types/kinds.

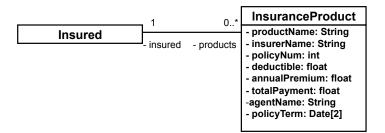
InsuranceProduct

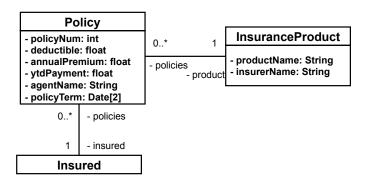
"XYZ" : insurerName

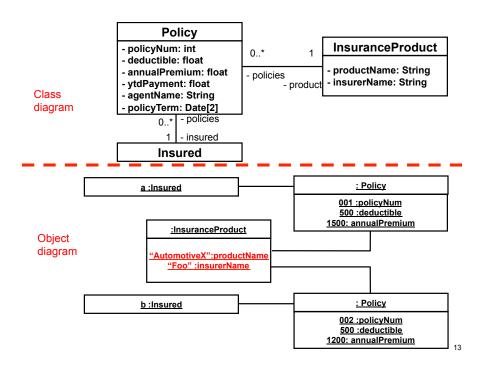
002 : policyNum

750: deductible 800: annualPremium "DEF" : agentName

- Each customer (insured) can buy multiple insurance products.
 - e.g., home owner's/renter's insurance, automotive insurance, boat insurance, life insurance...
 - The properties of each insurance product:
 - Product name, insurer's name, deductible, annual premium, the total (e.g., year-to-date) amount of payments, policy number, agent's name, policy term (e.g., 09/01/07 to 08/31/08)
 - · Let's take an insurance agency's viewpoint.





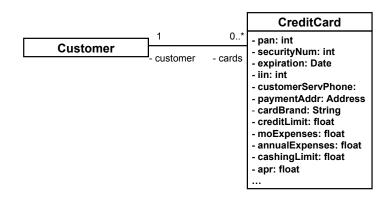


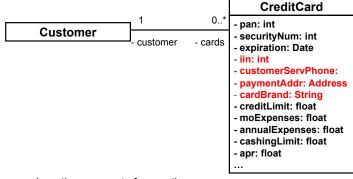
Exercise: Credit Cards

- Each customer can have multiple credit cards.
 - The properties of each card:
 - Issuer (bank; e.g., Chase, Citibank, etc.), brand (Visa, MasterCard, Amex, etc.), card number (primary account number: PAN), issuer identification number (IIN), security number, expiration date, monthly credit limit, monthly cashing limit, month-to-date expenses, year-todate expenses, payment address, customer service phone number, finance charge APR, etc. etc.

14

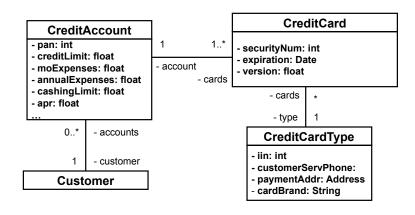
• Let's take an issuer's (bank's) viewpoint.

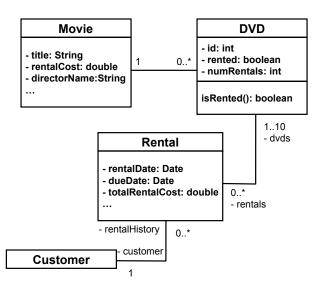




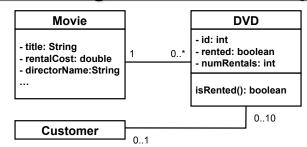
Instances share the same set of properties.

Think of separating credit cards from their types/kinds.





Revisiting the DVD Example



- How can you keep each customer's rental history?
 - 2 DVDs rented 2 weeks ago, and returned last week
 - 5 DVDs rented last week, and not returned yet
 - 3 DVDs rented today

17

By the way...

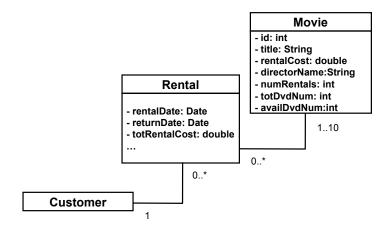
- Do you really need to model physical DVD media?
 - Maybe
 - Rigorous property management to keep track of individual DVD media
 - Who is renting which DVD media?
 - When was each DVD media purchased/created/disposed?
 - · Quality management for DVD media
 - Disposing DVD media when it has been rented certain times.
 - Sharing DVD media among different DVD stores
 - cf. car rental, car dealer, book rental (library) and credit cards you need rigorous property management.
 - Maybe not
 - Simpler property management
 - c.f. Netflix
 - Particularly for sales businesses rather than rental businesses.
 - » Book stores (e.g., Amazon.com)

Σ Δmazon com)

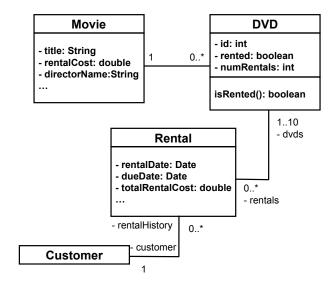
18

20

• Simpler DVD media management

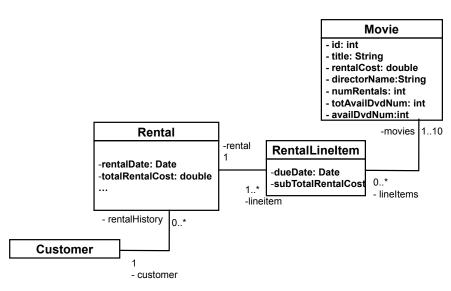


21

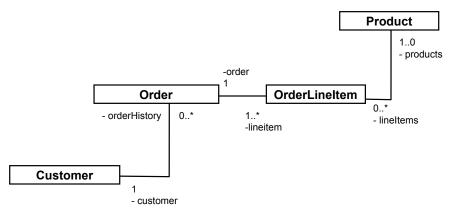


• What if different DVD media have different due dates?

DVD Movie id: int title: String - rented: boolean rentalCost: double - numRentals: int · directorName:String isRented(): boolean 1..10 - dvds Rental -rental RentalLineItem -rentalDate: Date -dueDate: Date -totalRentalCost: double -subTotalRentalCost - lineltems -lineitem - rentalHistory Customer - customer



Product Sales



HW5-1

- Choose the insurance or credit card example and add more details to it based on your own insurance policies or credit cards.
 - Make the example model more detailed/realistic

26

- Draw a class diagram and an object diagram
- Write code for it with a simple test code.

