Association Class

Java Collections

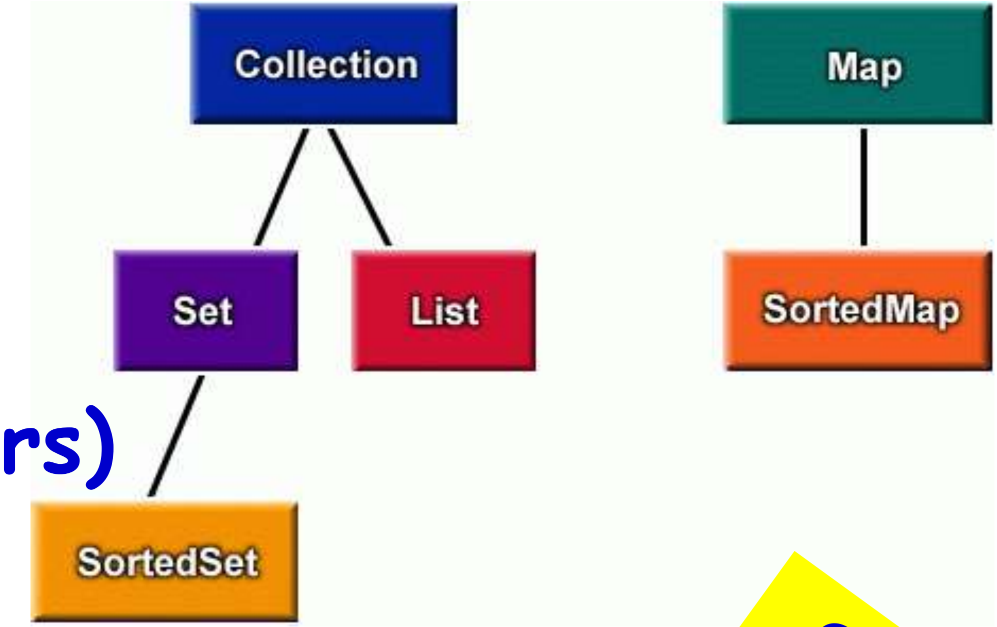
- When association-end multiplicities is *, need to use Collections.

- Collections were added to Java as part of JDK 1.2

- Operations supported:
  - **Add**
  - **Remove**
  - **Access individual objects (Iterators)**

Java Collections

- Java collections are of three basic types:
  - **List (Ordered)**
  - **Set (Unordered)**
  - **Map (Key-value pairs)**

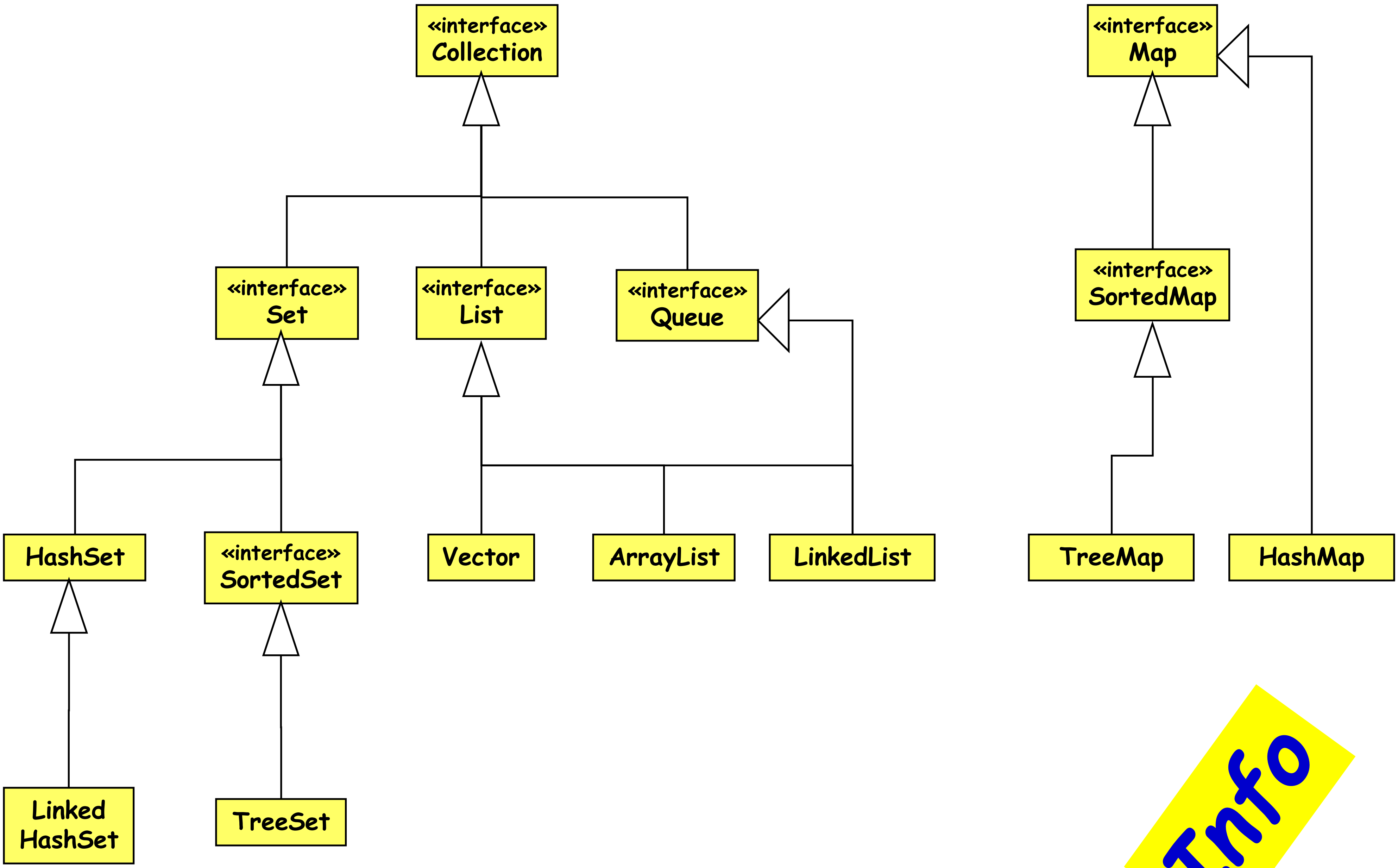


Info

Bit of History…

Info

- Pre Java SDK1.2, Java provided a handful of data structures:
  - **Hashtable**
  - **Vector**
  - **Bitset**
- These were for the most part good and easy to use:
  - But were not organized into a general framework.
- **These legacy data structures retrofitted to new model in SDK1.2 .**



«interface»
Collection

«interface»
Set

«interface»
List

«interface»
Queue

HashSet

«interface»
SortedSet

Vector

ArrayList

LinkedList

Linked
HashSet

TreeSet

«interface»
Map

«interface»
SortedMap

TreeMap

HashMap

Info

Java Collections

		Implementations			
		Hash Table	Resizable Array	Balanced Tree	Linked List
Interfaces	Set	HashSet		TreeSet	
	List		ArrayList		LinkedList
	Map	HashMap		TreeMap	

Info

Three Basic Java Collections

- **List : Sequences**
  - Ordering is implicit
  - it is legitimate to ask questions like "what is the first object in the sequence?"
  - Add book as first book etc.
- **Sets: Unordered collection**
- **Maps: Qualified associations**
  - Each entry involves a pair of objects.
  - **A map is also called as a dictionary.**
  - it is legitimate to ask questions like "what value - if any - is associated with the following key?" or
  - "does this map contain the following key?".
- For all types of Collections:
  - Can create an Iterator object to access each item in the collection once.

Info

Which List to Use?

- **LinkedList:**
  - Good if the list changes size (grows or shrinks) frequently
  - Good for accessing either end of the list, but slower when accessing items in the middle of the list
- **ArrayList:**
  - Good if accessing elements by specific position, but slow for adds and removes.

Info

Vector class vs ArrayList

Info

- Vector similar to an ArrayList, but synchronized for multithreaded programming.
  - **ArrayList is faster since it is non-synchronized, while vector is thread-safe**
- Vectors retained mainly for backward-compatibility with old java.
- Used as base class for *Stack* implementation.

Which Set to Use?

- **HashSet:**
  - Good efficiency in most cases

Info

- **TreeSet:**
  - Useful when an iterator will access the elements of the set in a specific order based on their value (e.g. Strings would be kept in alphabetical order.)

Which Map to Use?

- **HashMap:**
  - Efficient in most cases

Info

- **TreeMap:**
  - An iterator obtained from the key set will access the elements of the map in key order.

Collection: Basic operations

```
ArrayList<Integer> numbers = new ArrayList<>();
ListIterator<Integer> iterate =  numbers.listIterator();
    while(iterate.hasNext()) {
      System.out.print(iterate.next() + ", ");
    }
```

```
int size( );

boolean isEmpty( );

boolean contains(Object element);

boolean add(Object element); // Optional

boolean remove(Object element); // Optional

Iterator iterator( );
```

Info

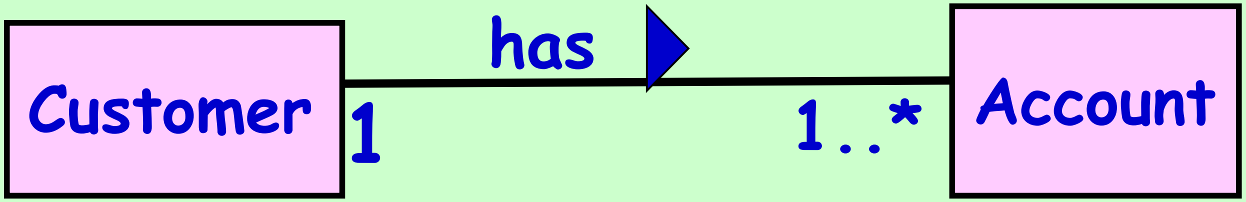
Collection: Iterator

```
public interface Iterator {

    boolean hasNext( );
      // true if there is another element

  Object next( );
     // returns the next element (advances the
     iterator)

    void remove( );
      // removes the element returned by next
}
```

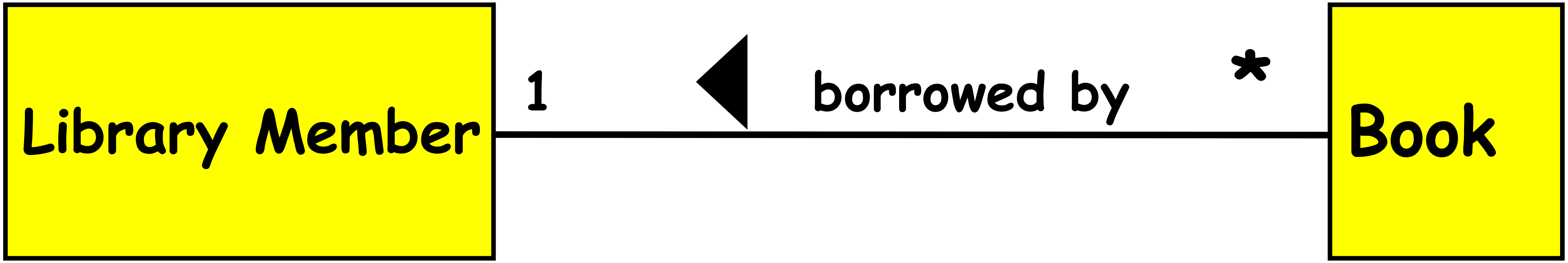
Info

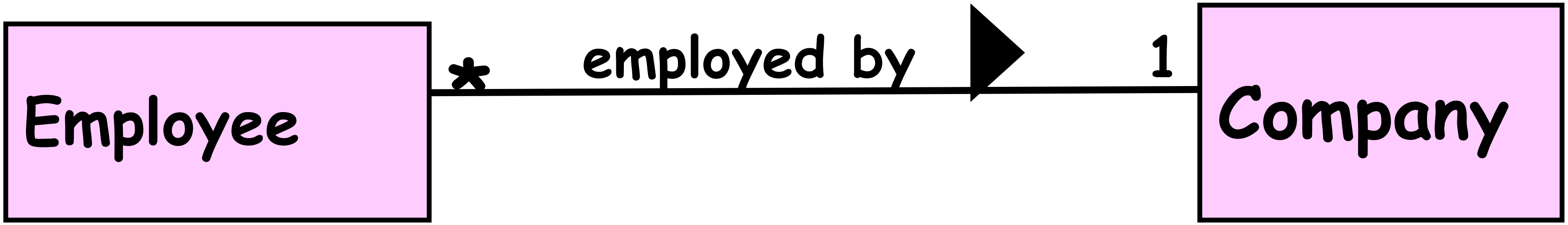
Code for Association Multiplicity

```
class Customer{
      private ArrayList <Account> accounts =
                   new ArrayList<Account>();
    public Customer() {
        Account defaultAccount = new Account();
          accounts.add(defaultAccount);
     }
}
```

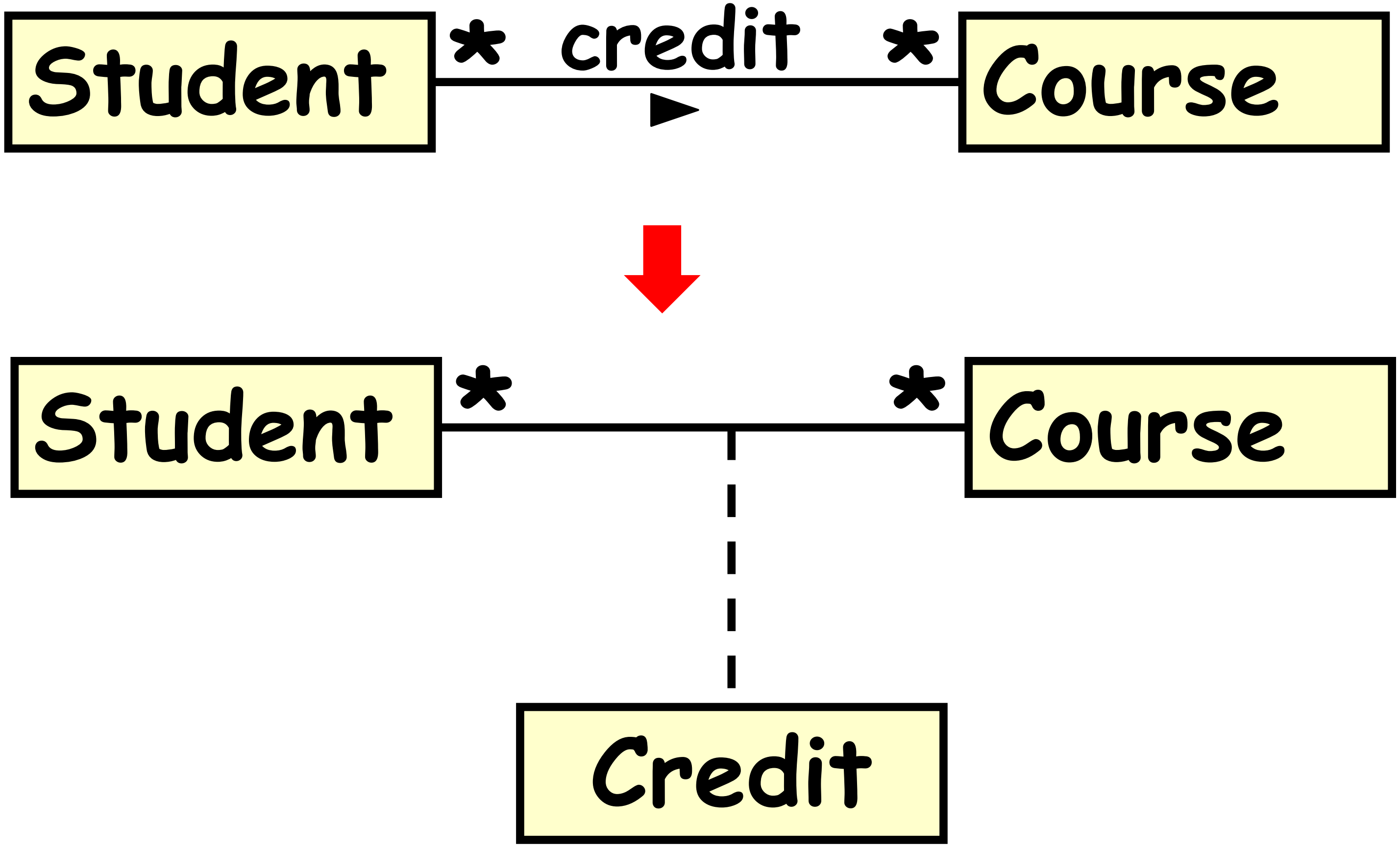


HW: Write Code for Example Association Relationships

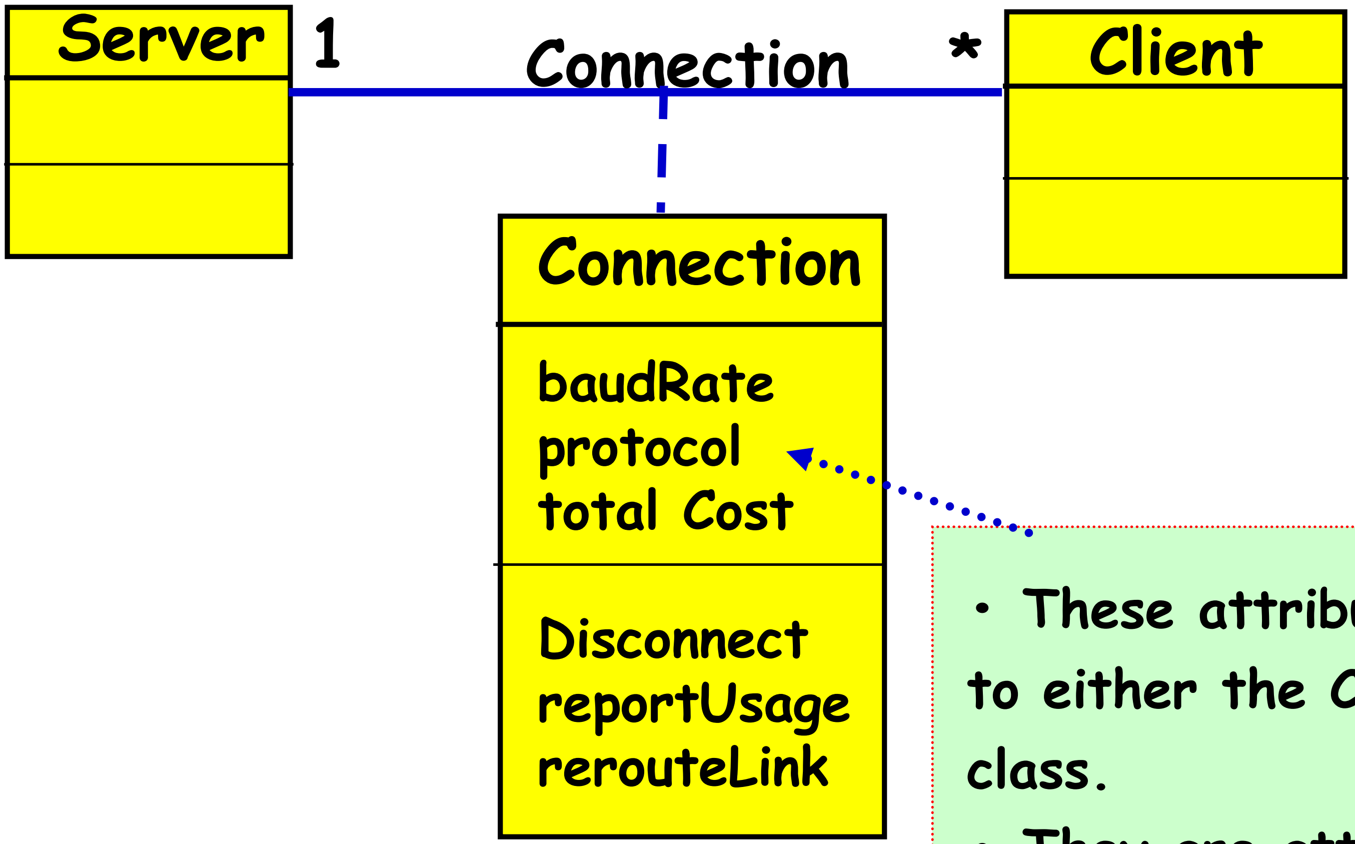




Association Class



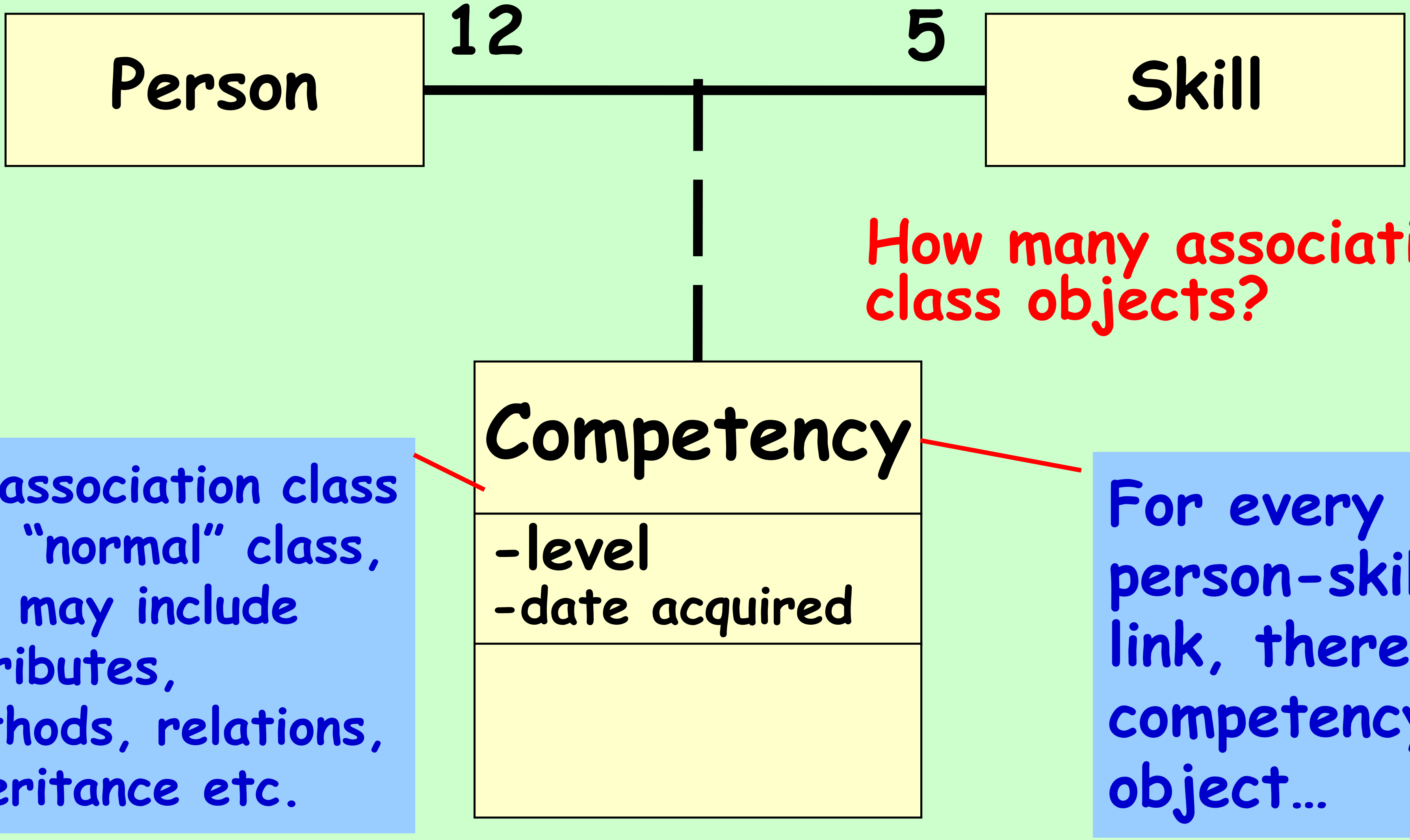
Association Class: Example 1



- These attributes don’t belong to either the Client or Server class.
- They are attributes of the connection itself.

- An association class can have methods as well as attributes.

Association Class: Example 2



Person

12

5

Skill

Competency

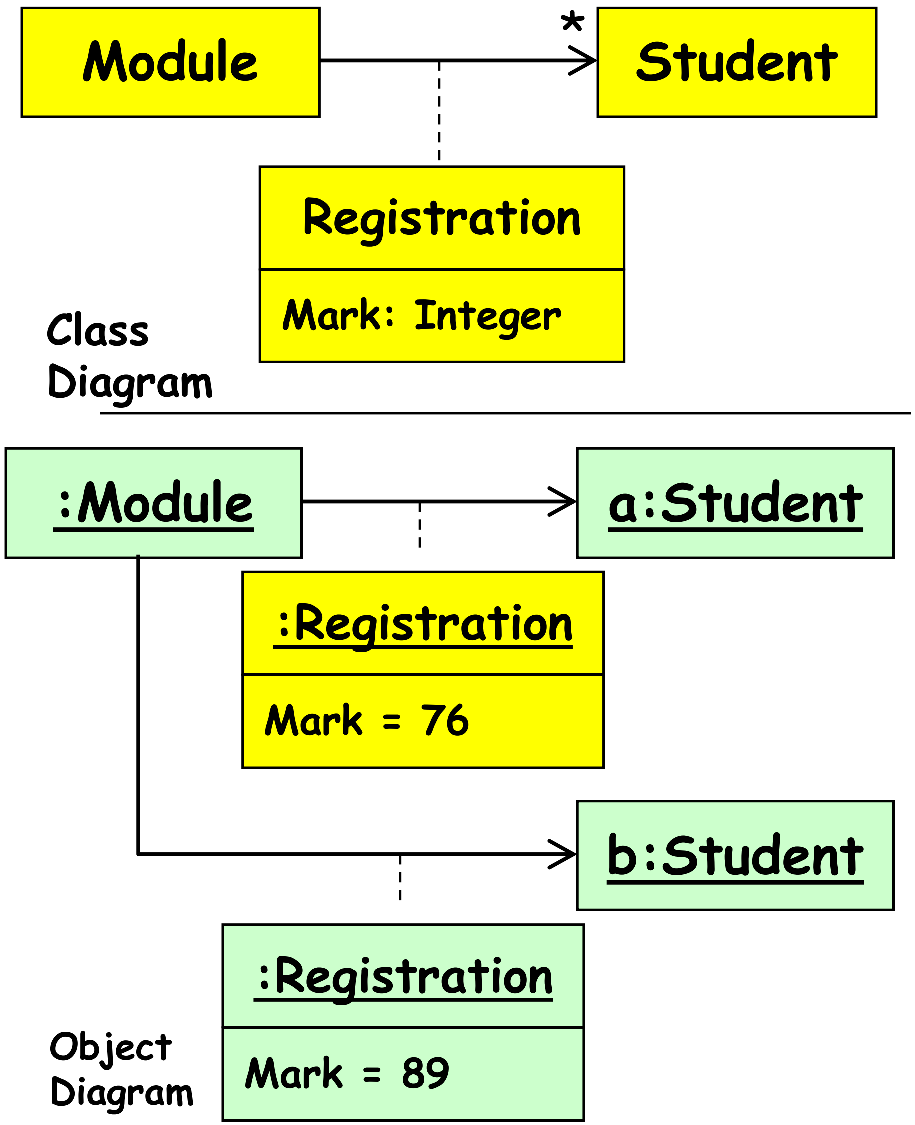
-level
-date acquired

How many association class objects?

An association class is a "normal" class, and may include attributes, methods, relations, inheritance etc.

For every person-skill link, there a competency object…

Implementing Association Class



Module

*

Student

Registration

Mark: Integer

Class Diagram

:Module

a:Student

:Registration

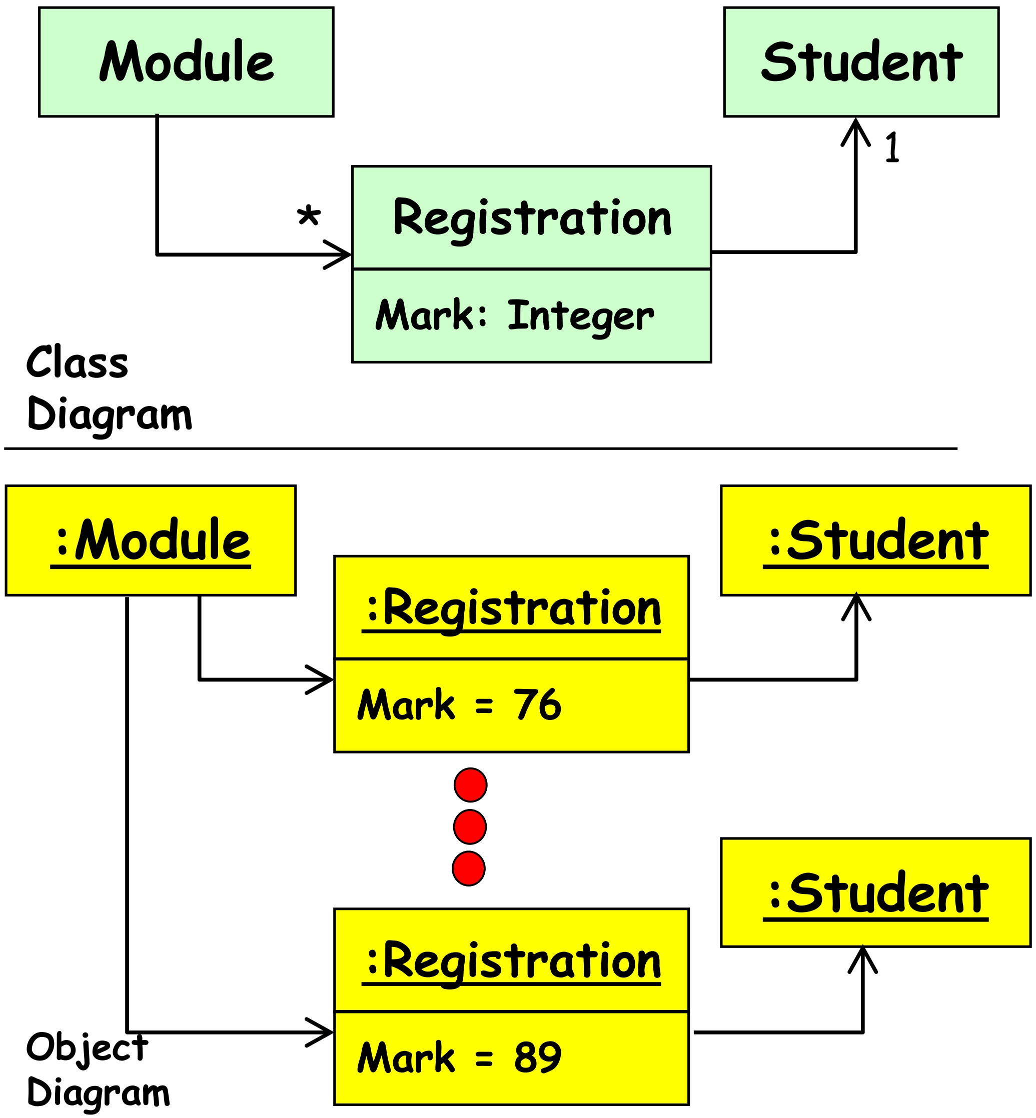
Mark = 76

b:Student

:Registration

Mark = 89

Object Diagram



Module

Student

1

*

Registration

Mark: Integer

Class Diagram

:Module

:Student

:Registration

Mark = 76

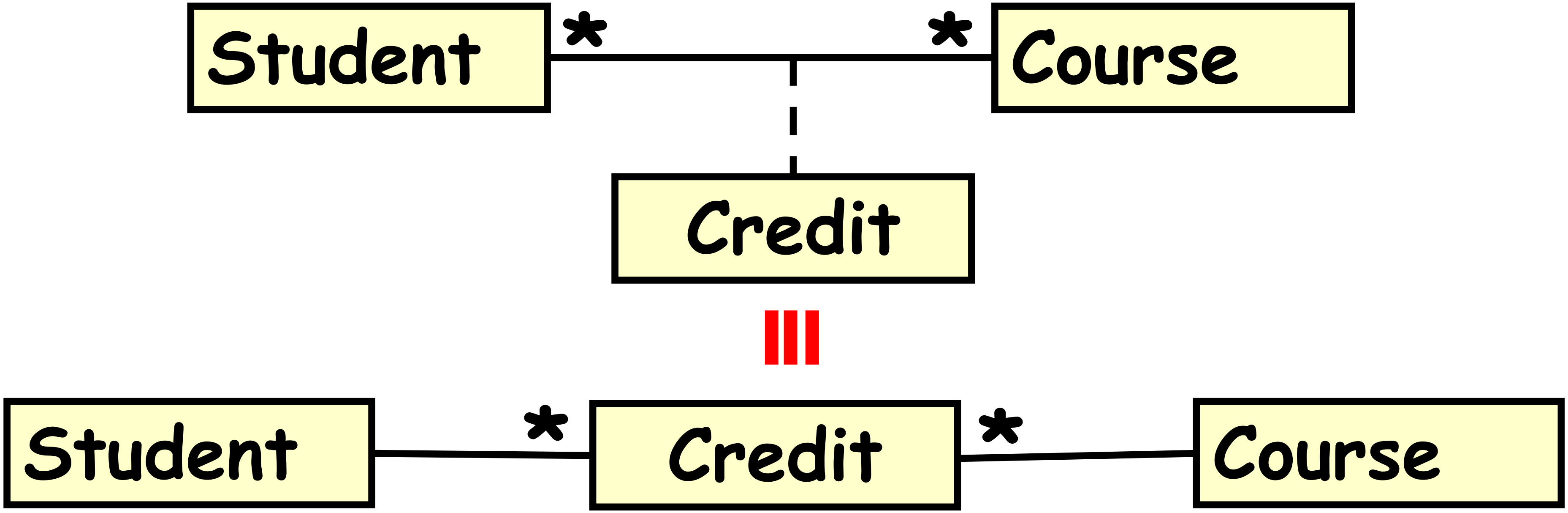
:Registration

Mark = 89

:Student

Object Diagram

Reification



- *Reification* means “Replacing an association class with a normal Class”.
- **Dictionary:** **Reification** is when you treat something abstract as a physical thing.

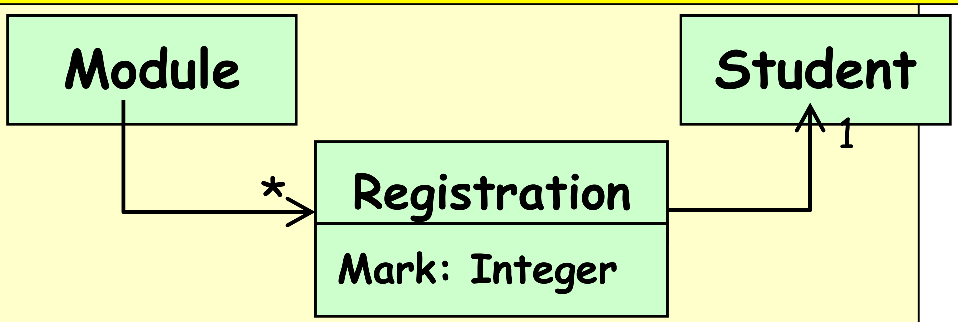
Association Class: Java Code

Pass the Student to Registration.

```
public class Module {
   private Vector <Registration> reg=new Vector<Registration>();
   public void enrol(Student st) {
      reg.addElement( new Registration(st) );
   }
   ...
}
```

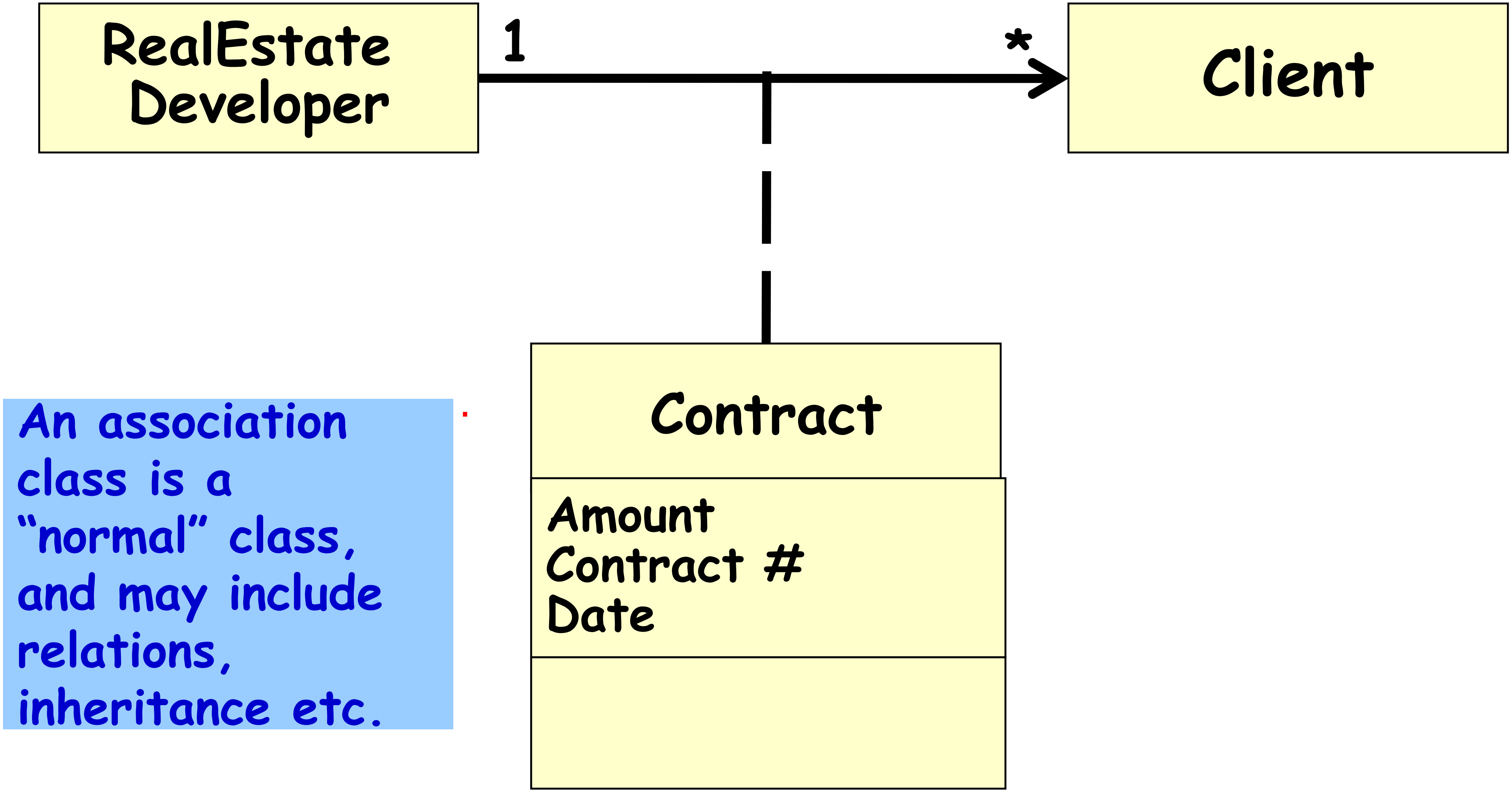
Maintain the link to Registration

```
class Registration {
   private Student student;
   private int mark;

   Registration(Student st) {
      student = st; mark = 0;
   }
   ...
}
```



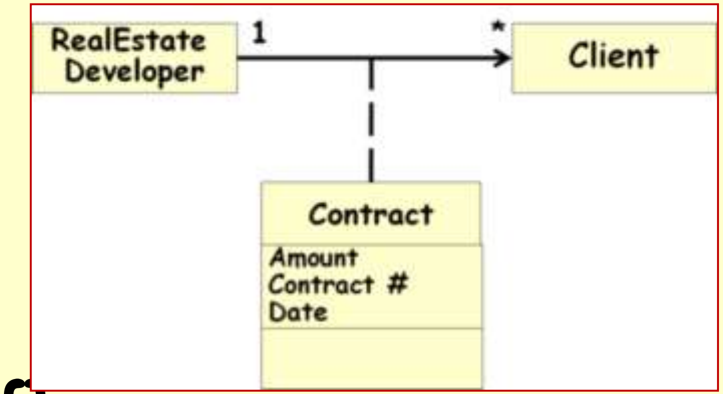
Keep track of the Student reference.

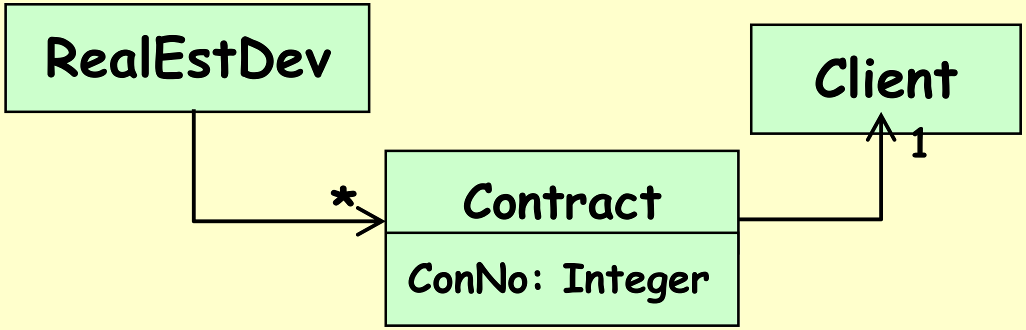
Association Class: Example 3



An association class is a "normal" class, and may include relations, inheritance etc.

```
public class RealEstateDeveloper{
  private Vector <Contract> contracts= new Vector
  <Contract>();
public void buy(Client c){contracts.add(new Contract(c))};
}
public class Client{
  private Address address;
  public Address getCurrentAddress(){}
}
public class Contract{
  private Client client;
  private int contractNo;
  public Contract(Client c){ client=c;}
}
```

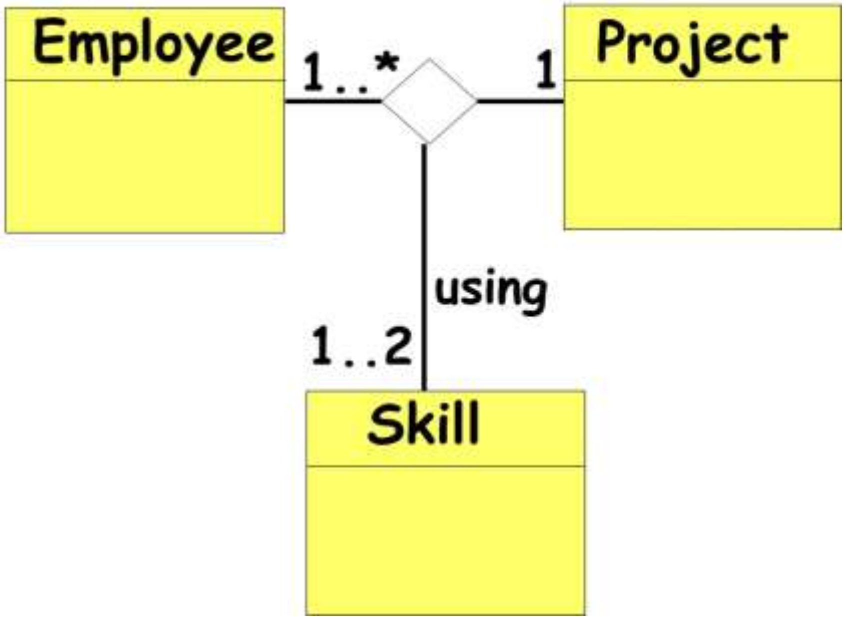




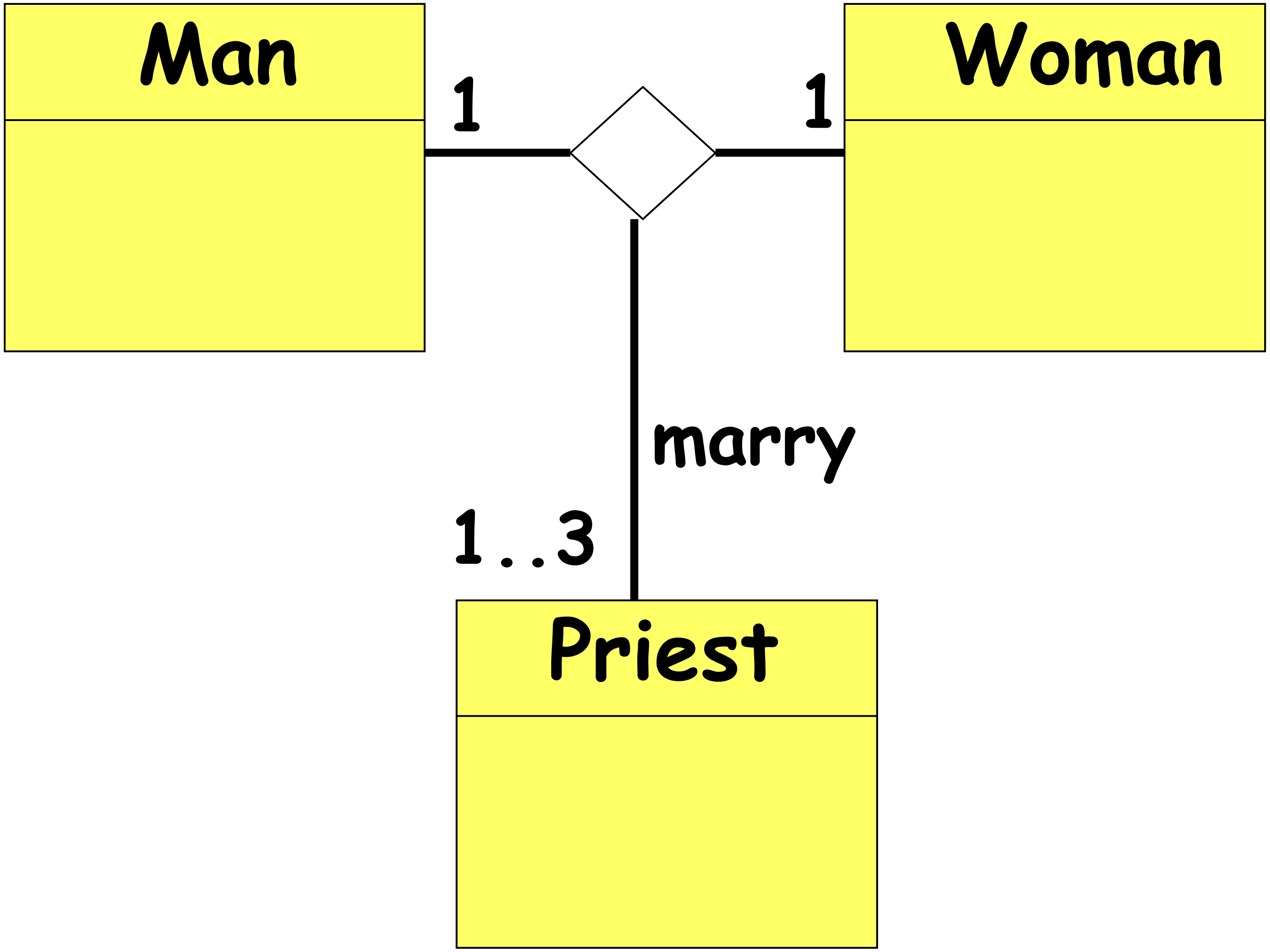
Ternary Association

- Some times three (or more) classes may be associated.

- An object of an association class:
  - Stores the details for the two associated classes.



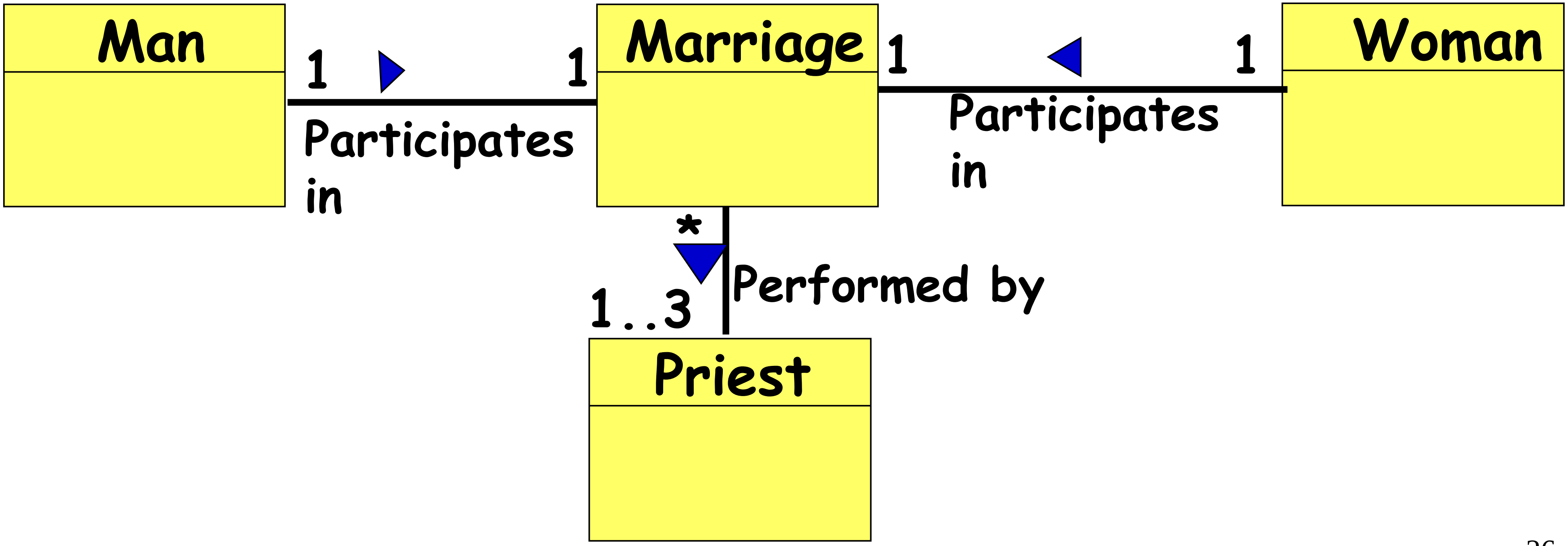
Ternary Association

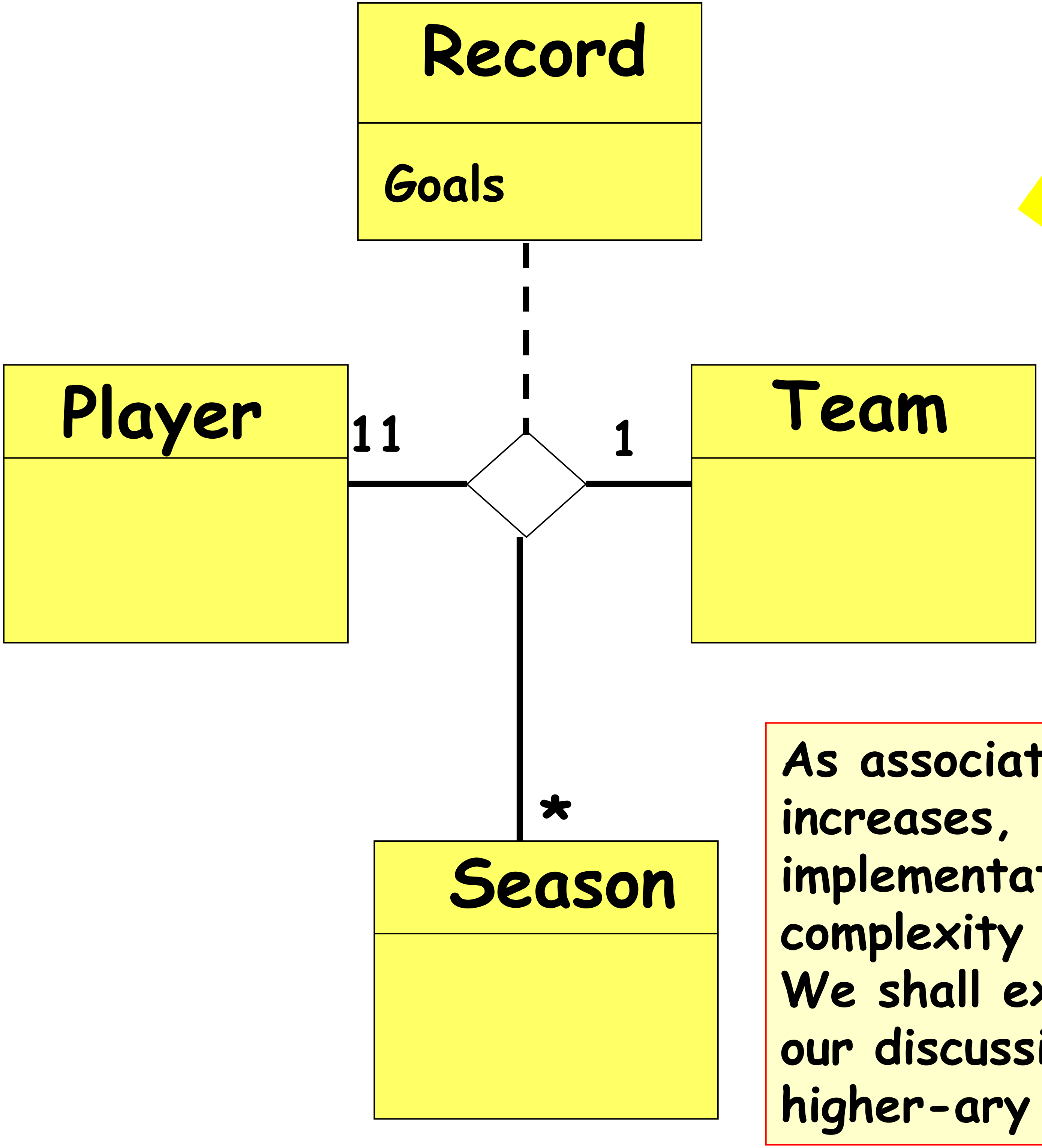


and we can add more classes to the diamond…

Implementation of Ternary Association

- There are several ways in which ternary association can be implemented.
  - One is to decompose it into a set of binary associations.

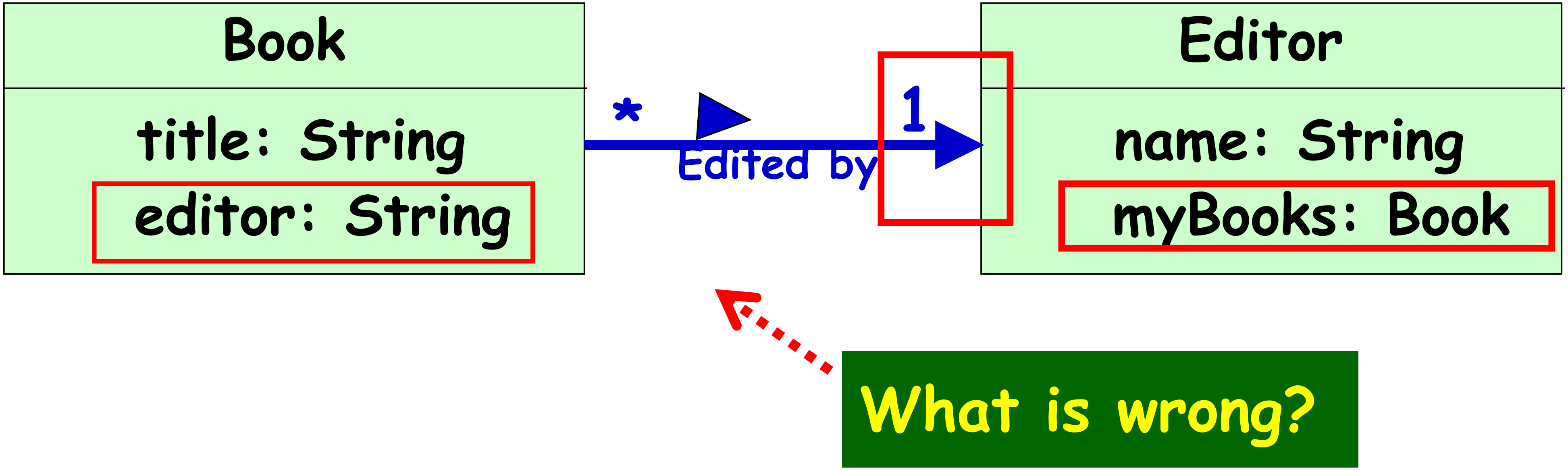




Info

As association arity increases, implementation complexity increases… We shall exclude from our discussions 3ary and higher-ary associations…

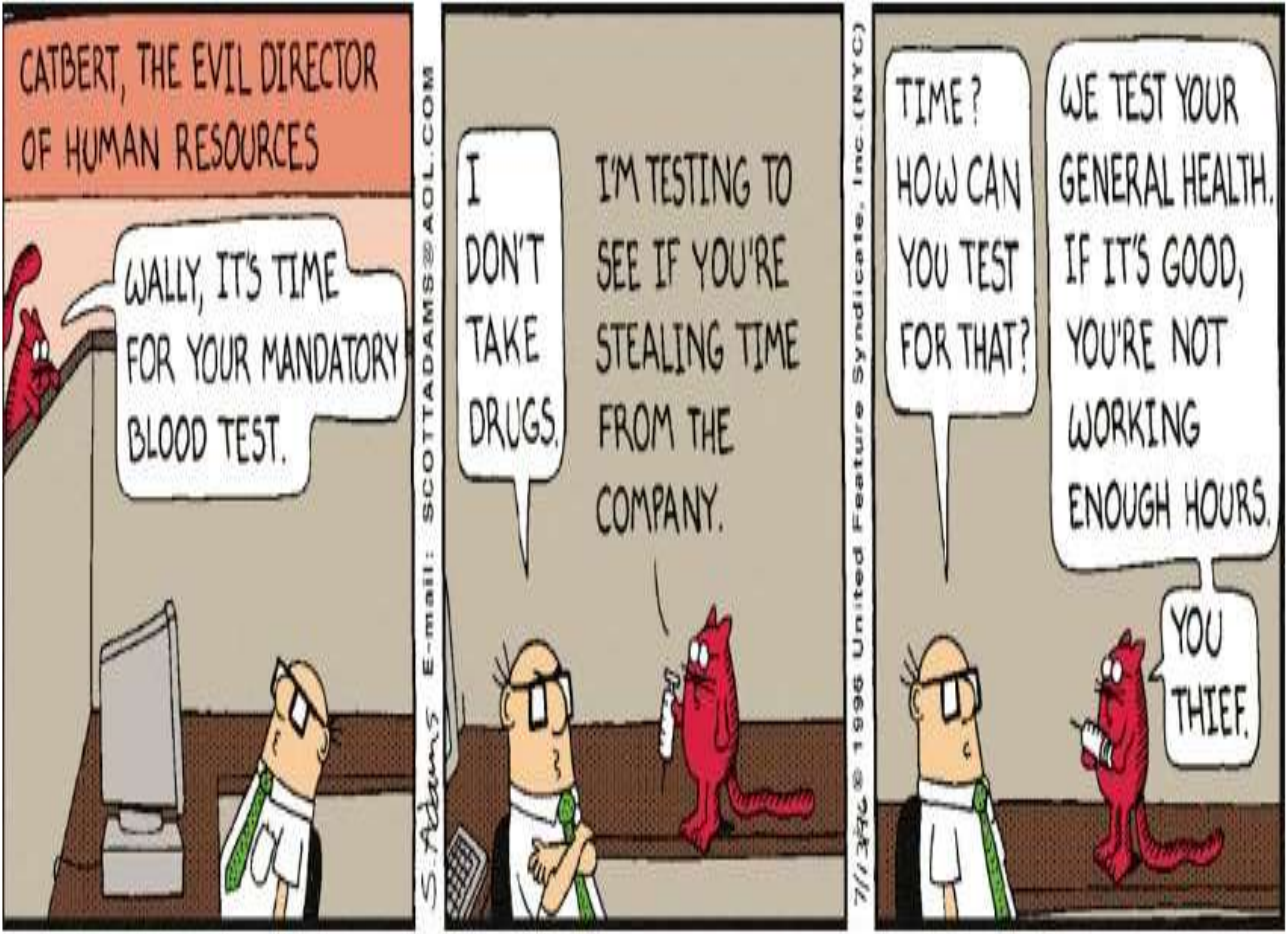
Association Quiz



What is wrong?

1. **Association denoted by symbol not attributes.**
   - Implementation (pointers, arrays, vectors, ids etc) is left to the detailed design phase.

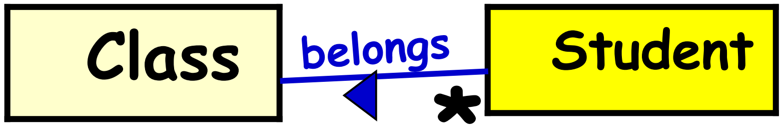
2. **Wrong arrow type**

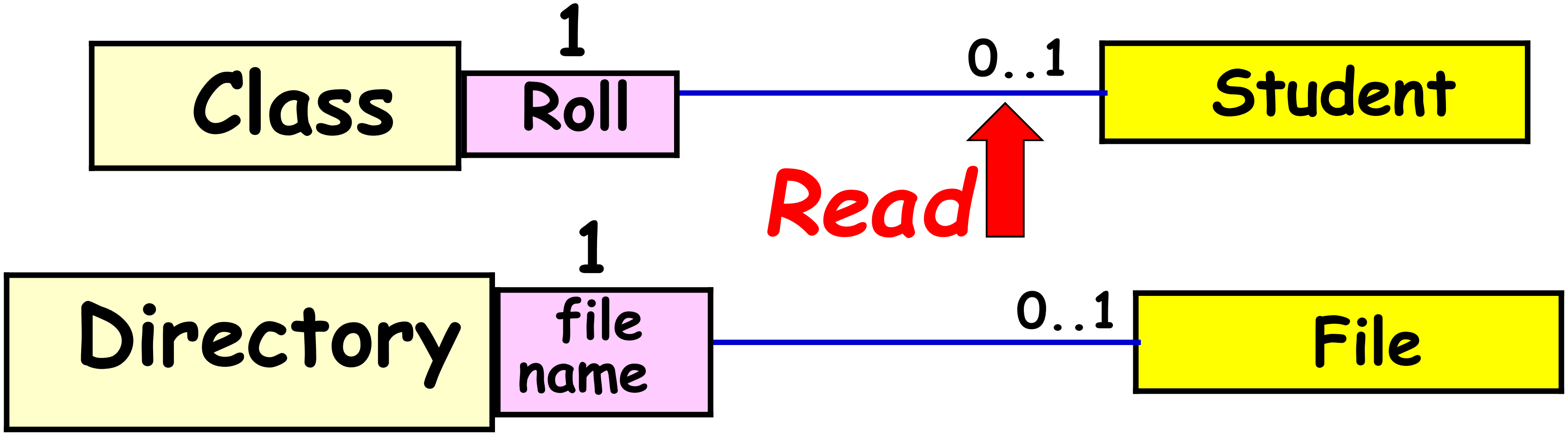

CATBERT, THE EVIL DIRECTOR OF HUMAN RESOURCES
WALLY, IT'S TIME FOR YOUR MANDATORY BLOOD TEST.
SCOTTADAMS@AOL.COM E-mail: S. Adams
I DON'T TAKE DRUGS.
I'M TESTING TO SEE IF YOU'RE STEALING TIME FROM THE COMPANY.
7/13/96 © 1996 United Feature Syndicate, Inc.(NYC)
TIME? HOW CAN YOU TEST FOR THAT?
WE TEST YOUR GENERAL HEALTH. IF IT'S GOOD, YOU'RE NOT WORKING ENOUGH HOURS.
YOU THIEF.

Qualified Association

Qualified Association

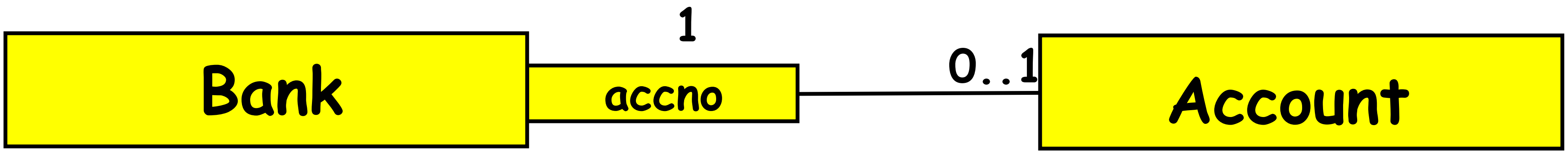
- **Allows us to express uniqueness…**
  - Implemented by hash tables, maps, dictionaries.





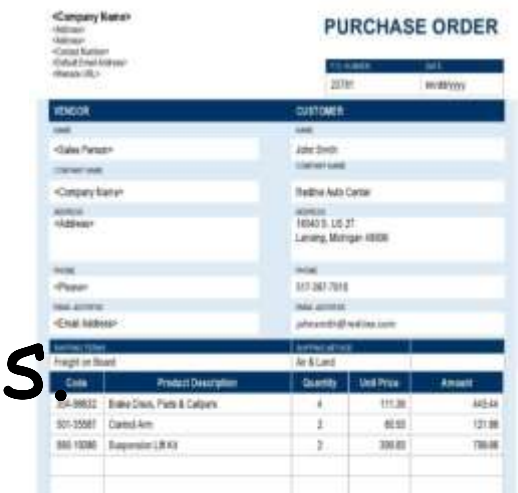
- How to read?
- **There exists upto one file for each instance of filename in the directory .**

Qualified Association



- Qualifier hints at setting up efficient access to linked objects:
  - For example, access accounts based only on the account number;
  - Implement to avoid a linear search through all accounts.

Setting up Qualified Association -- An Example



<Company Name>
<Address>
<Address>
<Contact Number>
<Default Email Address>
<Website URL>

PURCHASE ORDER

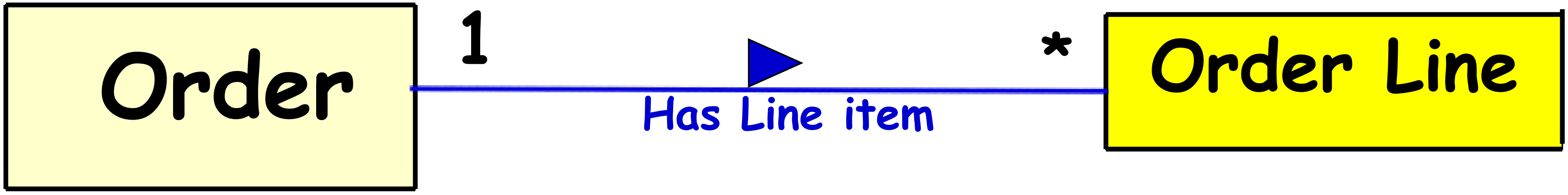
P.O. NUMBER	DATE
23781	mm/dd/yyyy

VENDOR	CUSTOMER
NAME	NAME
<Sales Person>	John Smith
COMPANY NAME	COMPANY NAME
<Company Name>	Redline Auto Center
ADDRESS	ADDRESS
<Address>	16040 S. US 27 Lansing, Michigan 48906
PHONE	PHONE
<Phone>	517-387-7818
EMAIL ADDRESS	EMAIL ADDRESS
<Email Address>	johnsmith@redline.com

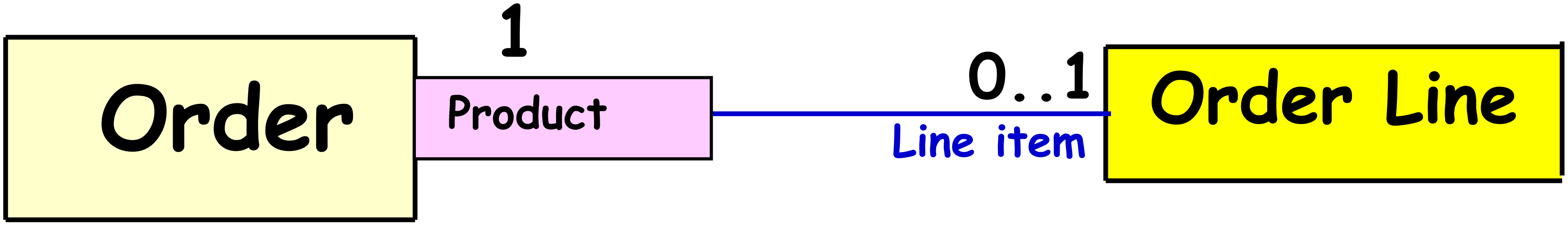
SHIPPING TERMS	SHIPPING METHOD
Freight on Board	Air & Land

Code	Product Description	Quantity	Unit Price	Amount
304-99832	Brake Discs, Pads & Calipers	4	111.36	445.44
301-35587	Control Arm	2	60.93	121.86
880-10086	Suspension Lift Kit	2	399.83	799.66

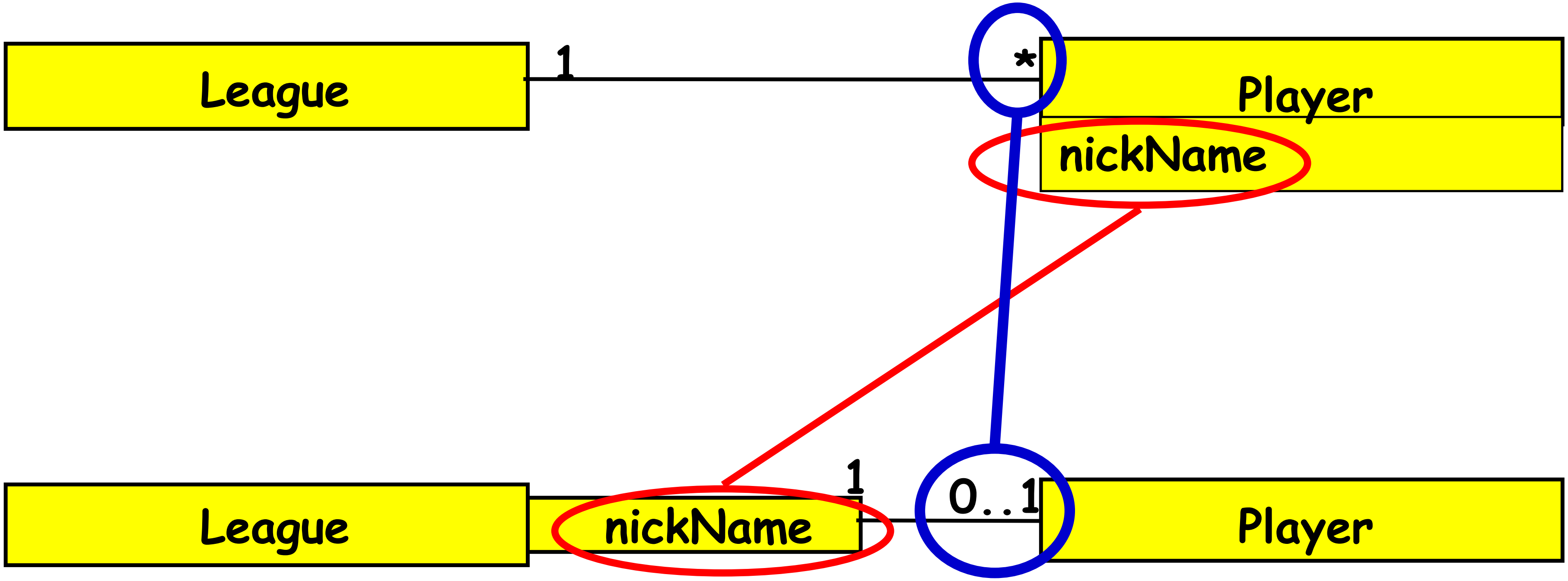
- An order has of many Order lines.



- How do you represent: There is at most one Order Line in the Order for each instance of Product.

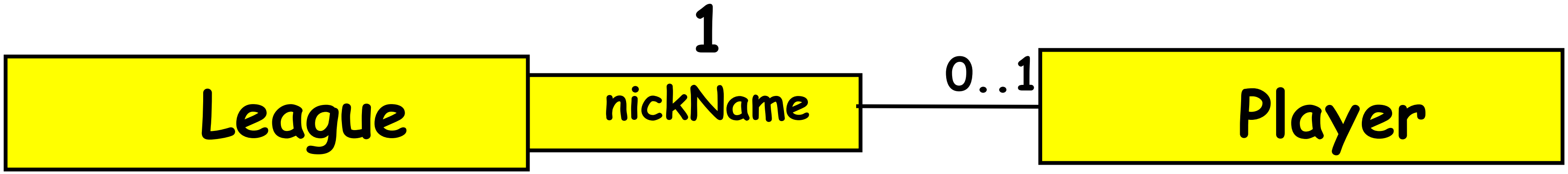


Qualified Association…



- The second conveys more information…
- Hints on implementation…
- Effectively reduces multiplicity…

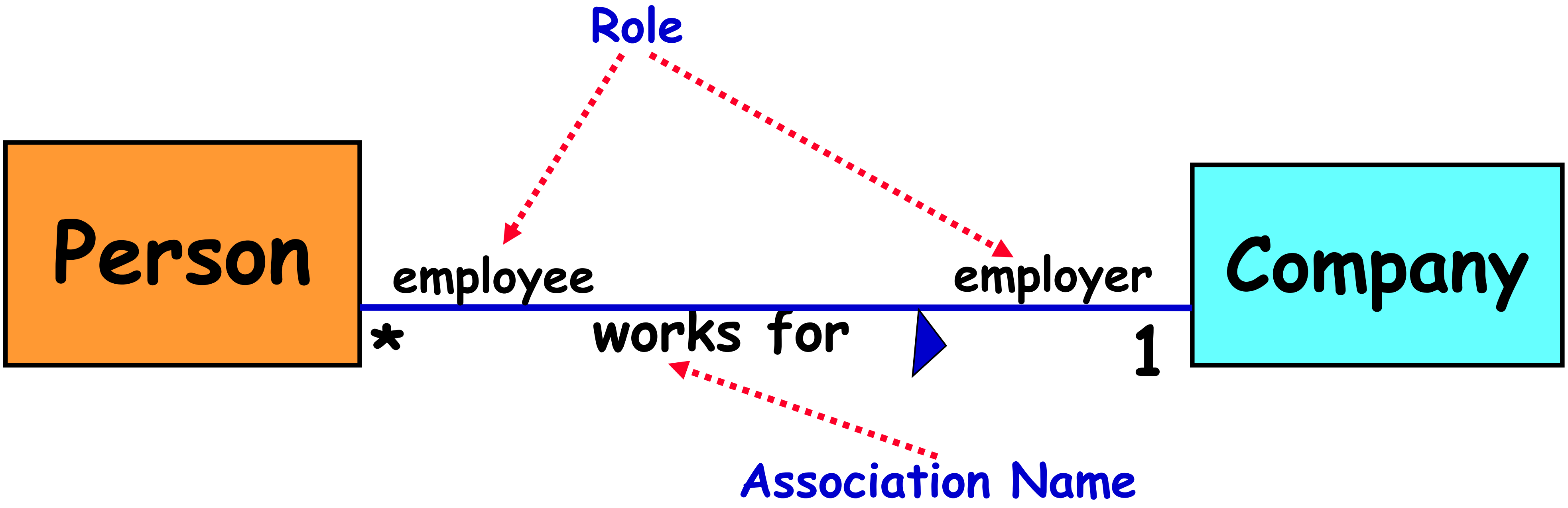
Qualified Association: Implementation



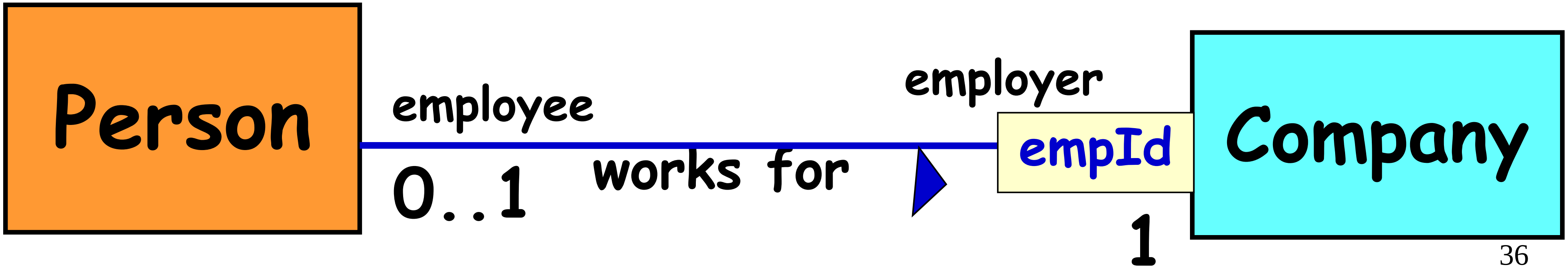
```
public class League {
  private Map players=new HashMap;

  public void addPlayer
   (String nickName, Player p) {
    if(!players.containsKey(nickName))
        players.put(nickName, p);
    }
}
```

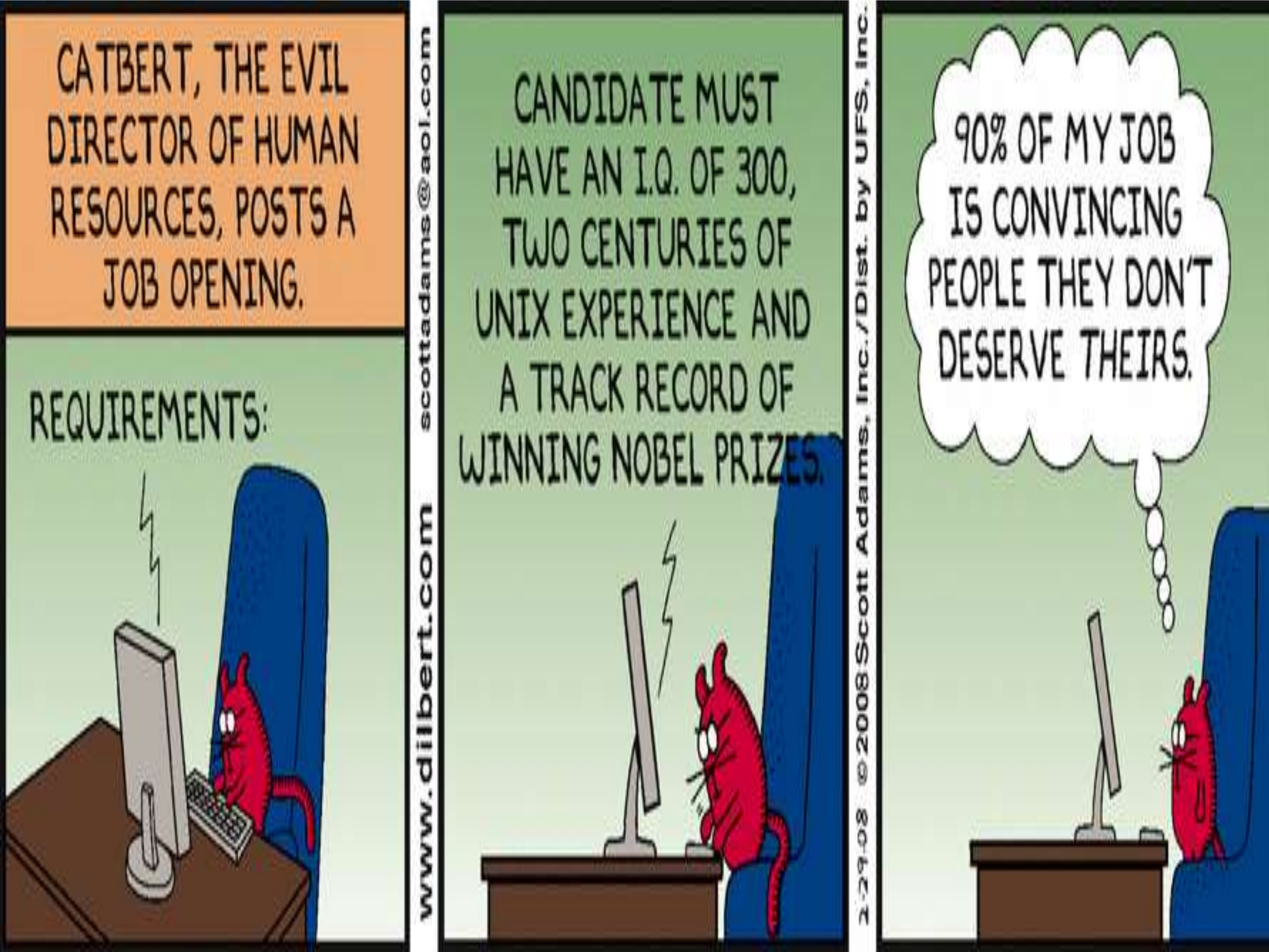
```
public class Player
  {
     private League
                  league;
  }
```

Converting to Qualified Association



Assume company assigns each employee a unique empId. Give qualified association representation…



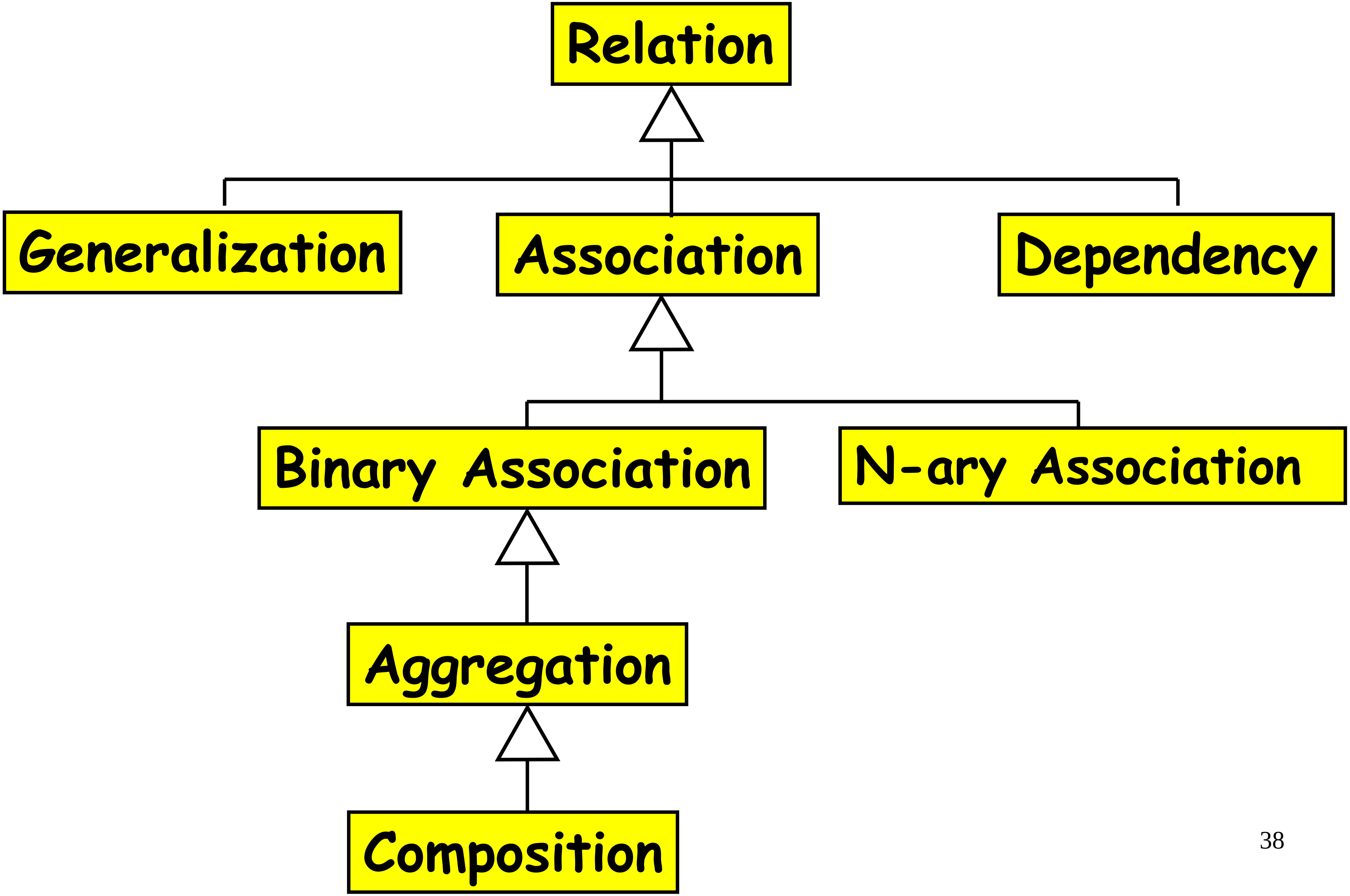

CATBERT, THE EVIL DIRECTOR OF HUMAN RESOURCES, POSTS A JOB OPENING.
REQUIREMENTS:
CANDIDATE MUST HAVE AN I.Q. OF 300, TWO CENTURIES OF UNIX EXPERIENCE AND A TRACK RECORD OF WINNING NOBEL PRIZES.
90% OF MY JOB IS CONVINCING PEOPLE THEY DON'T DESERVE THEIRS.

scottadams@aol.com

www.dilbert.com

1-29-08 ©2008 Scott Adams, Inc./Dist. by UFS, Inc.

Types of Class Relationships



Relation

Generalization

Association

Dependency

Binary Association

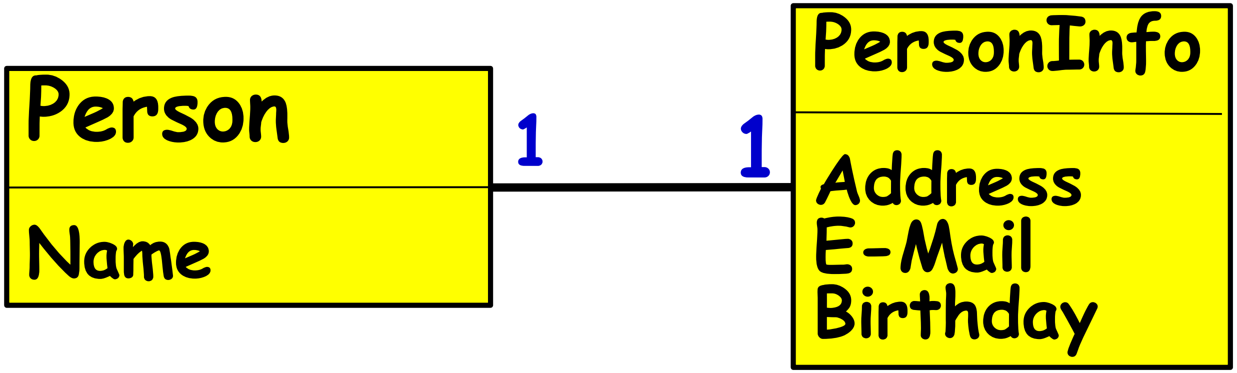
N-ary Association

Aggregation

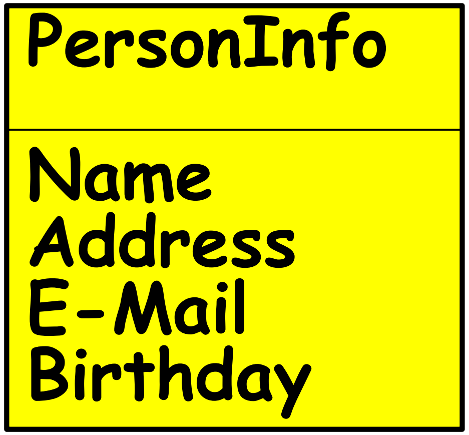
Composition

Overdoing Associations

- Avoid unnecessary Associations



Avoid This…



Do This

"Or" Association

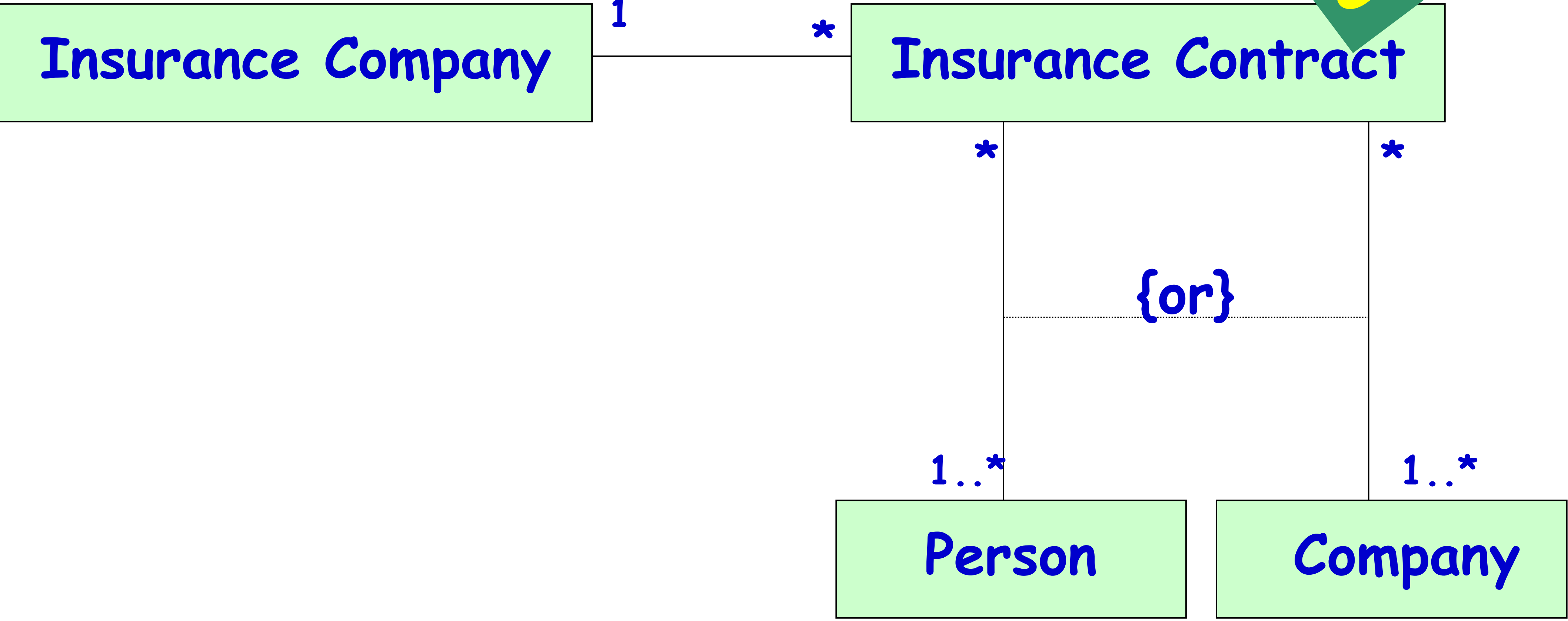
- **Used when all association combinations in a class model are not valid.**

Skip

- **Example:**
  - A person can have an insurance contract with an insurance company
  - Also a company can have an insurance contract with an insurance company
  - A person and a company CANNOT have the same insurance contract with an insurance company.

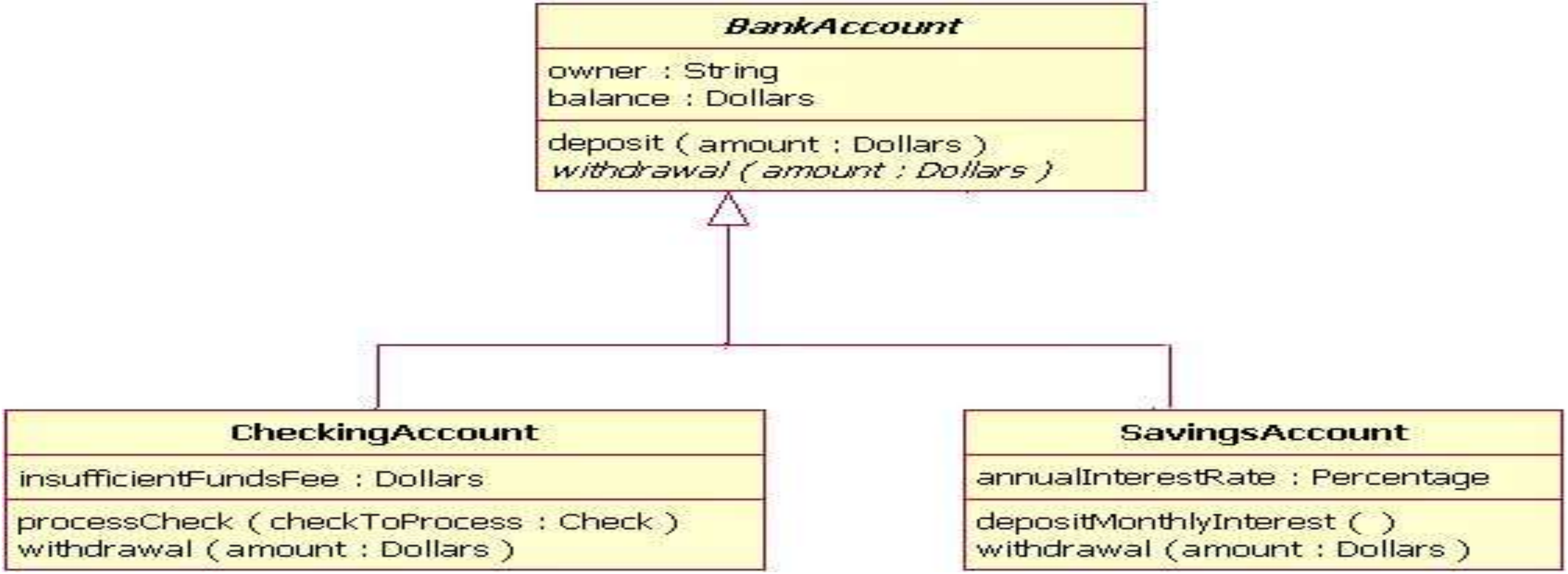
“Or” Association Example

Skip



Summary of Implementation of Association

- **1-to-1 association:**
  - Role names become attributes
- **1-to-many association:**
  - Translate into a Vector or ArrayList
- **Qualified association:**
  - Translate into a Map or Hash table



BankAccount

owner : String
balance : Dollars

deposit (amount : Dollars)
withdrawal (amount : Dollars)

CheckingAccount

insufficientFundsFee : Dollars

processCheck (checkToProcess : Check)
withdrawal (amount : Dollars)

SavingsAccount

annualInterestRate : Percentage

depositMonthlyInterest ( )
withdrawal (amount : Dollars)

Which sentences are true?

a) CheckingAccount implements BankAccount ✗

b) CheckingAccount and SavingAccount are BankAccount ✓

c) CheckingAccount and SavingAccount are associated ✗

d) BankAccount is associated to CheckingAccount ✗

e) SavingAccount can processCheck ✗

f) CheckingAccount has a balance ✓