

# Introducing to Computing & Security -- INFO 101

Exercises on basic UML:  
Structures

Rui Abreu

 @rmaranhao

# Summary

- This is a set of exercises on writing and understanding the UML notations
- Some exercises have more than one correct answer
- Most of them are taken from the net

# Diagrams covered

- Class diagrams
- Object diagrams
- Domain diagrams
- Context diagrams
- Feature diagrams

Think about it!



# Reading exercises

- Read the UML class diagram
- Pay attention to the multiplicity
- Try to understand the relationship and describe it in natural language

# Veterinary System

- Try to read & understand this UML diagram



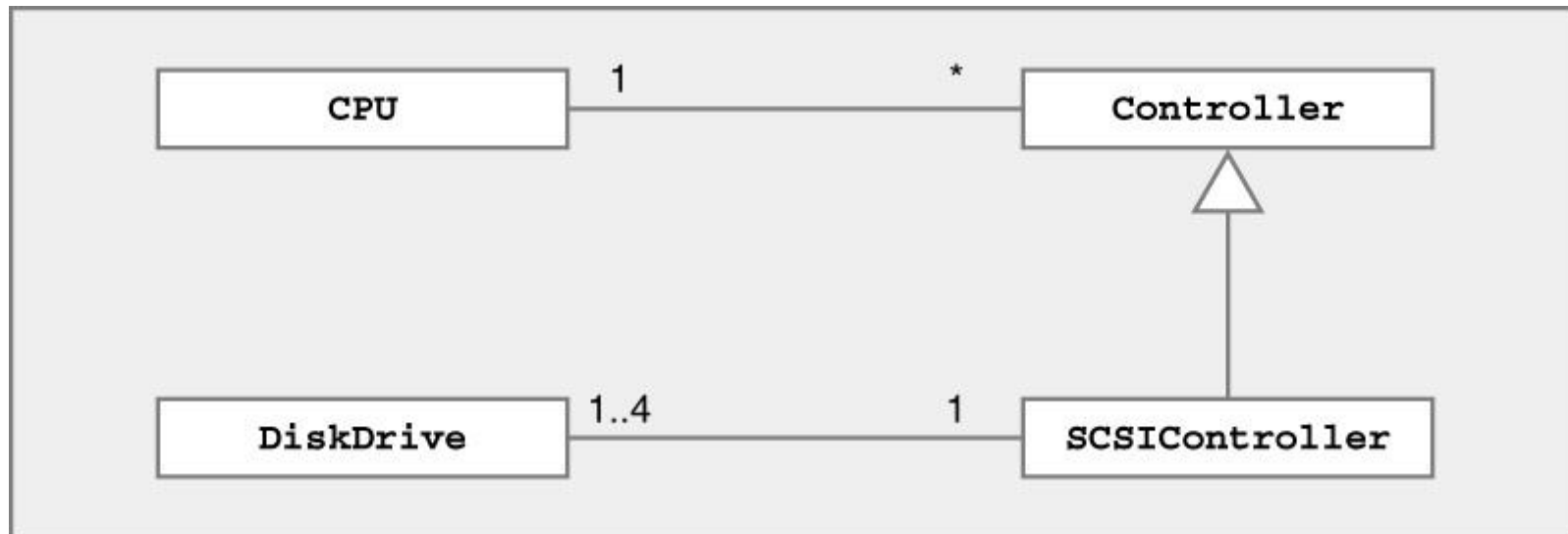
# Veterinary System

- Read and understand this UML diagram



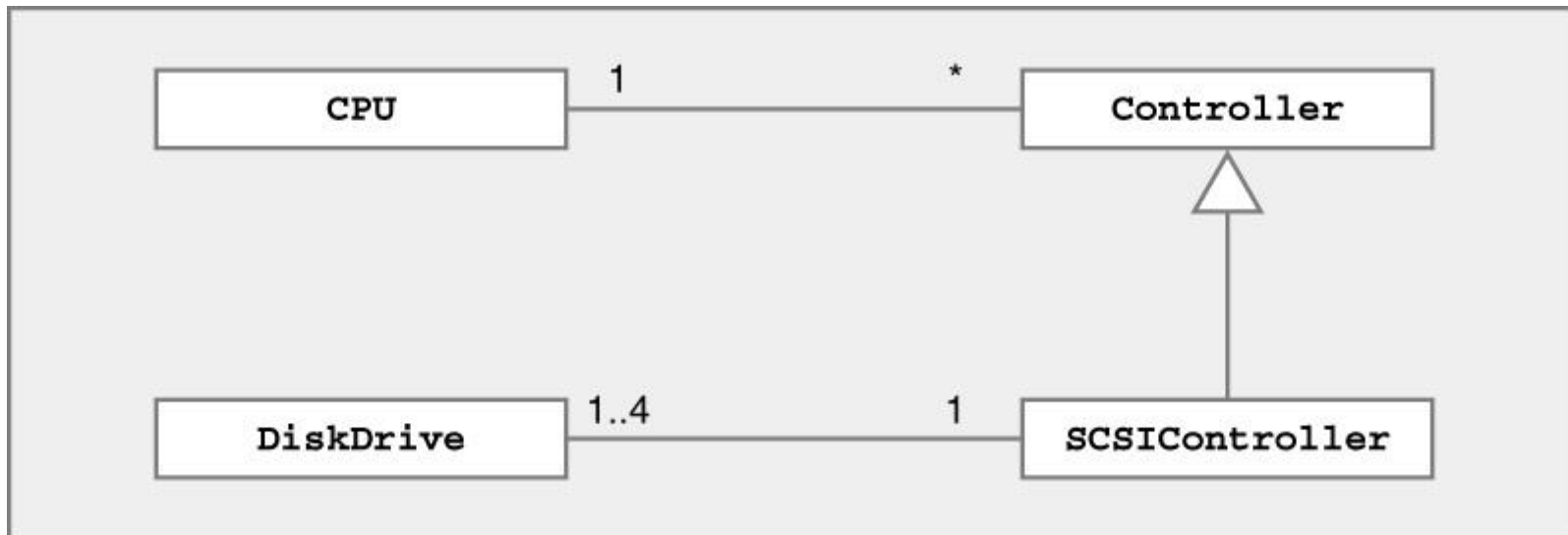
- 1 or more Pets associated with 1 PetOwner
- Each pet has exactly one PetOwner

# Computer System



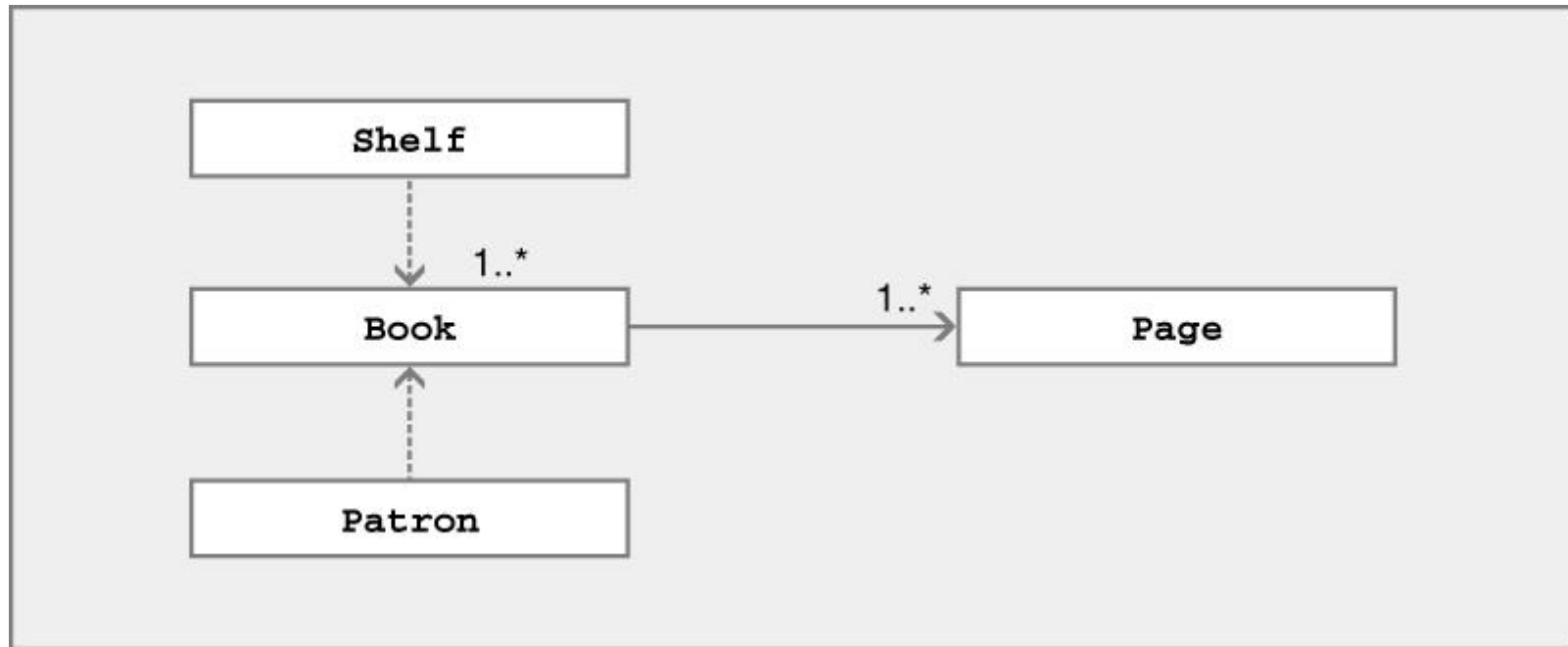


# Computer System

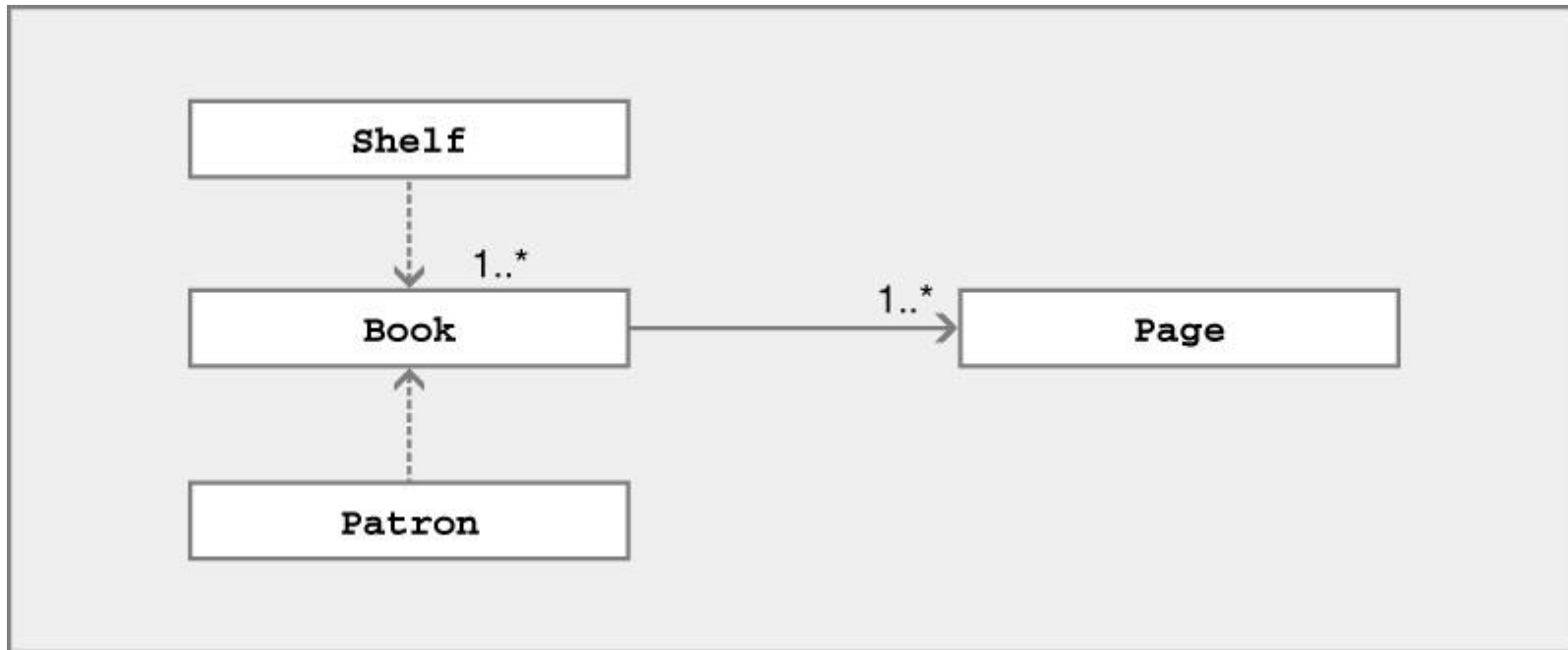


- 1 CPU associated with 0 or more Controllers
- 1-4 DiskDrives associated with 1 SCSIController
- SCSIController is a (specialized) Controller

# Library System

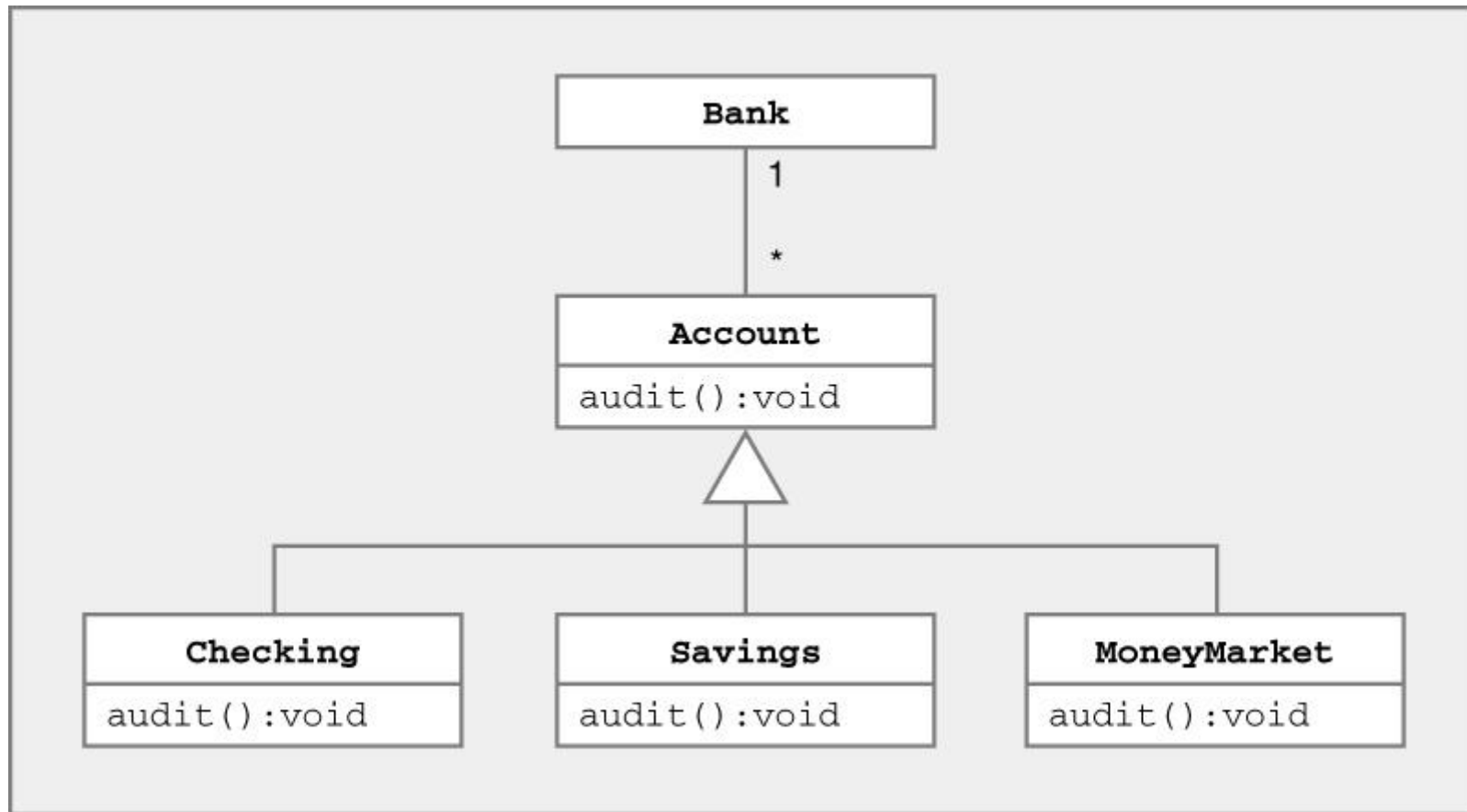


# Library System

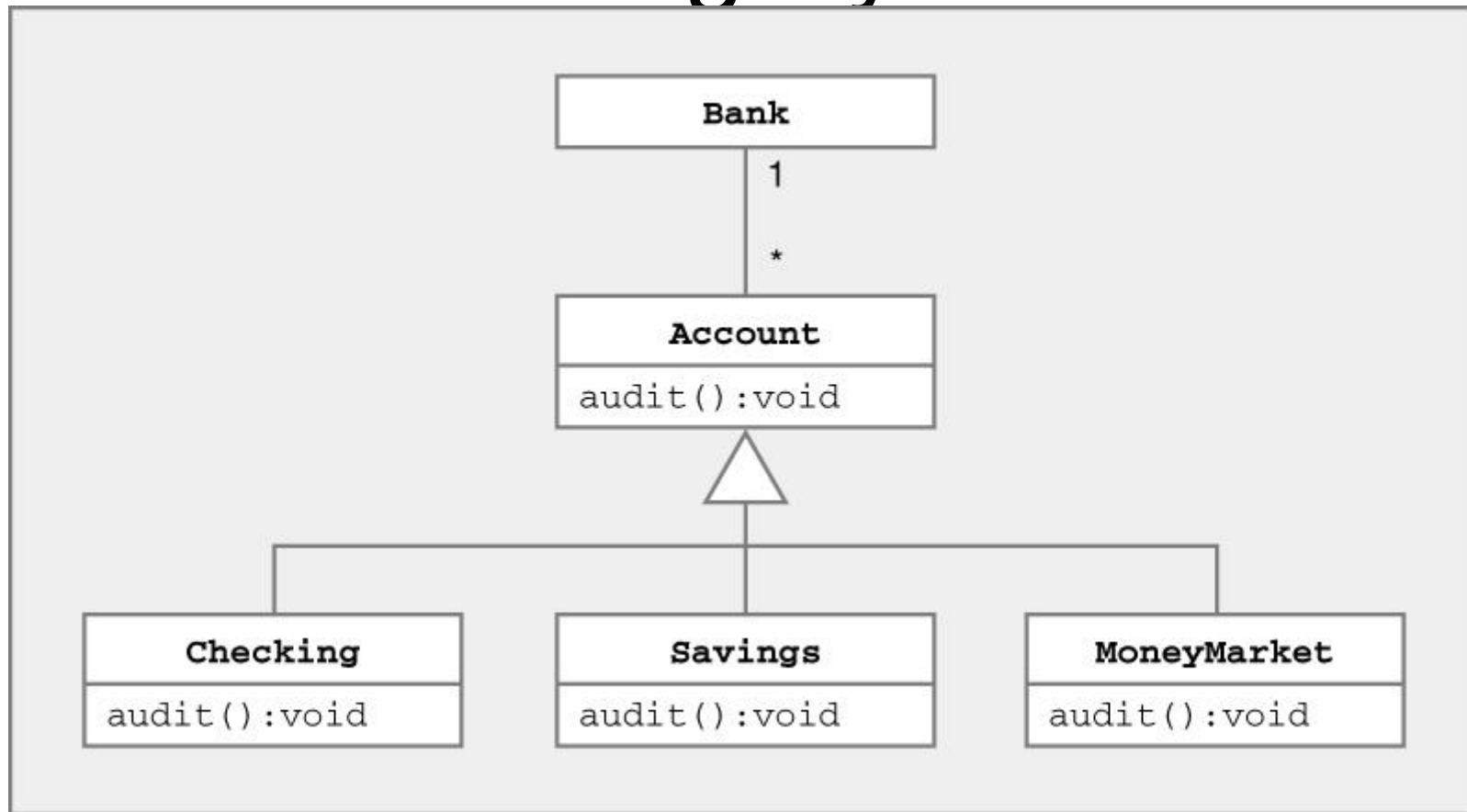


- 1 or more Book associated with 1 or more Pages
- Patron & Shelf use (depend on) Books

# Banking System

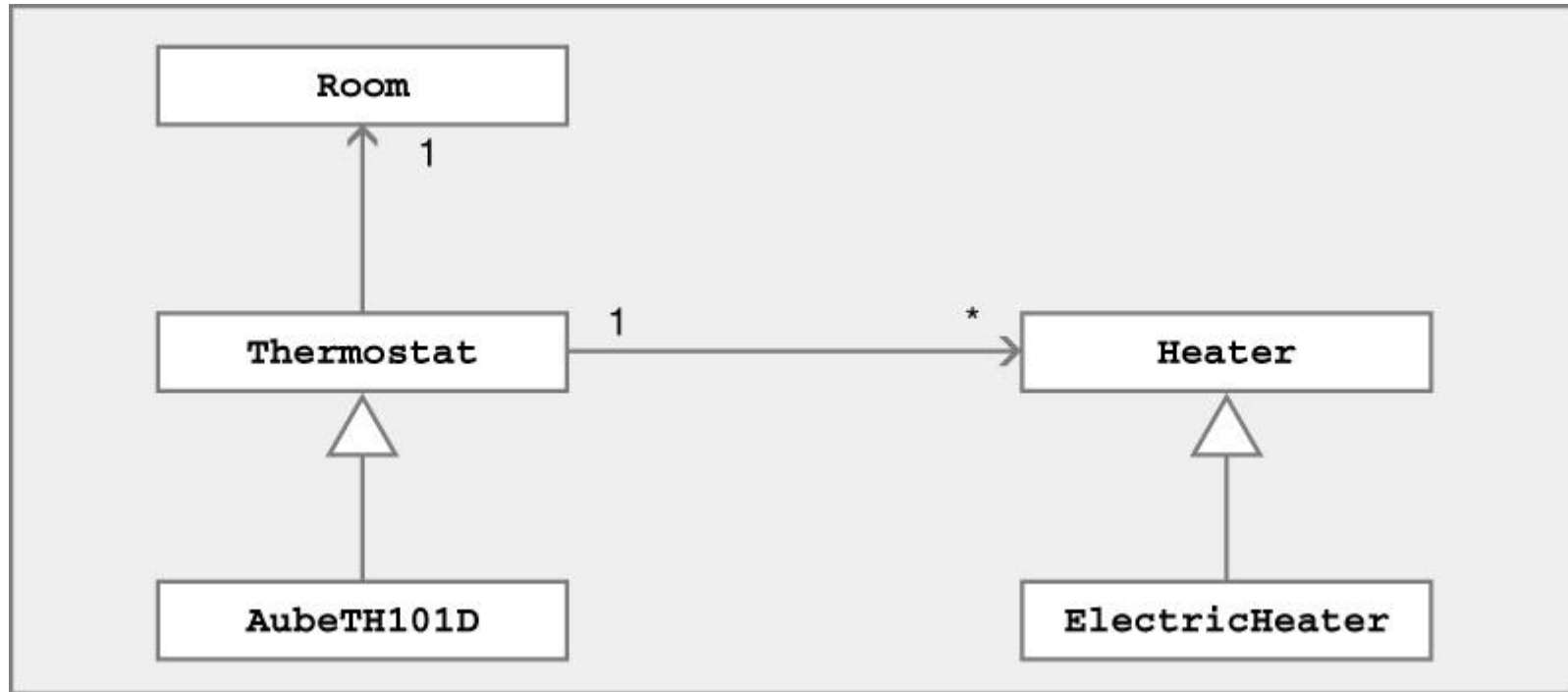


# Banking System

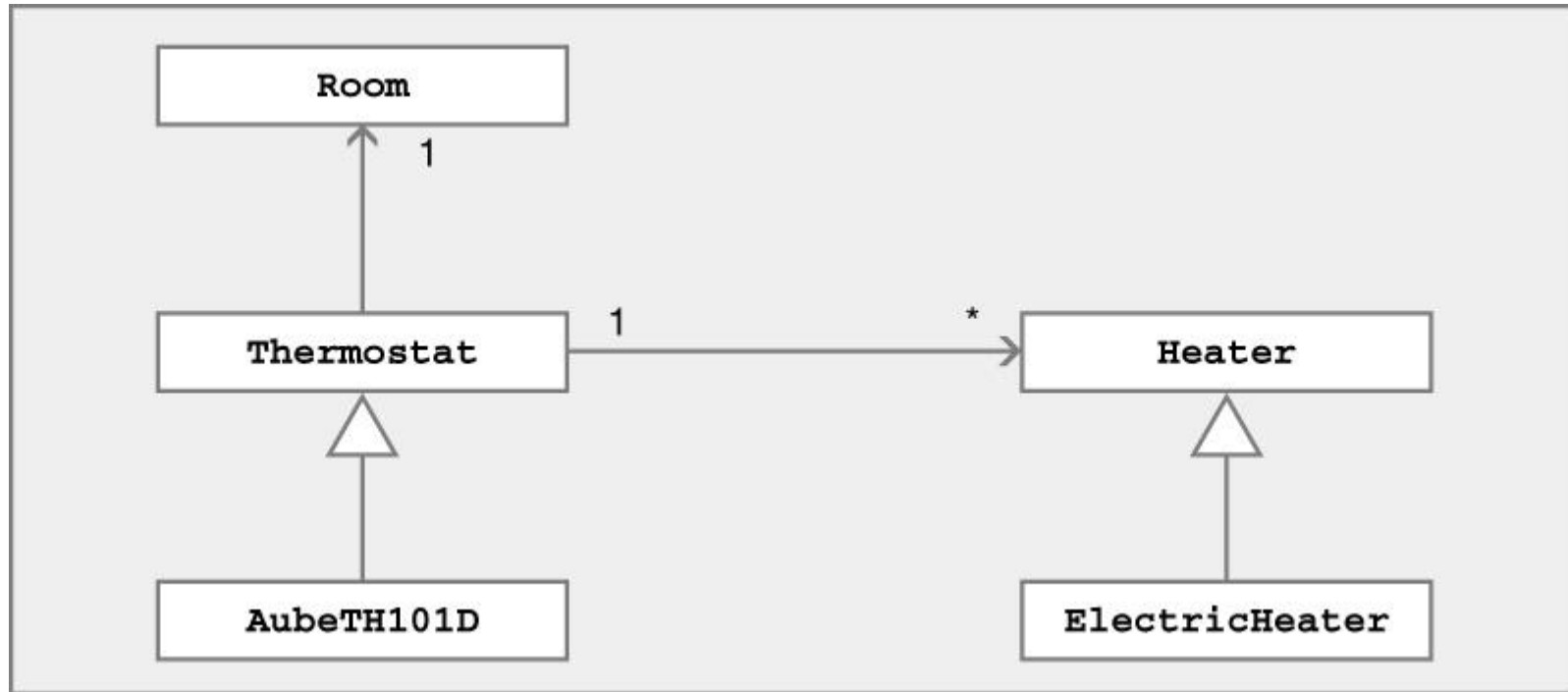


- 1 Bank associated with 0 or more Accounts
- Each Account is associated with exactly one bank
- Checking, Savings, and MoneyMarket are Accounts

# Home Heating System

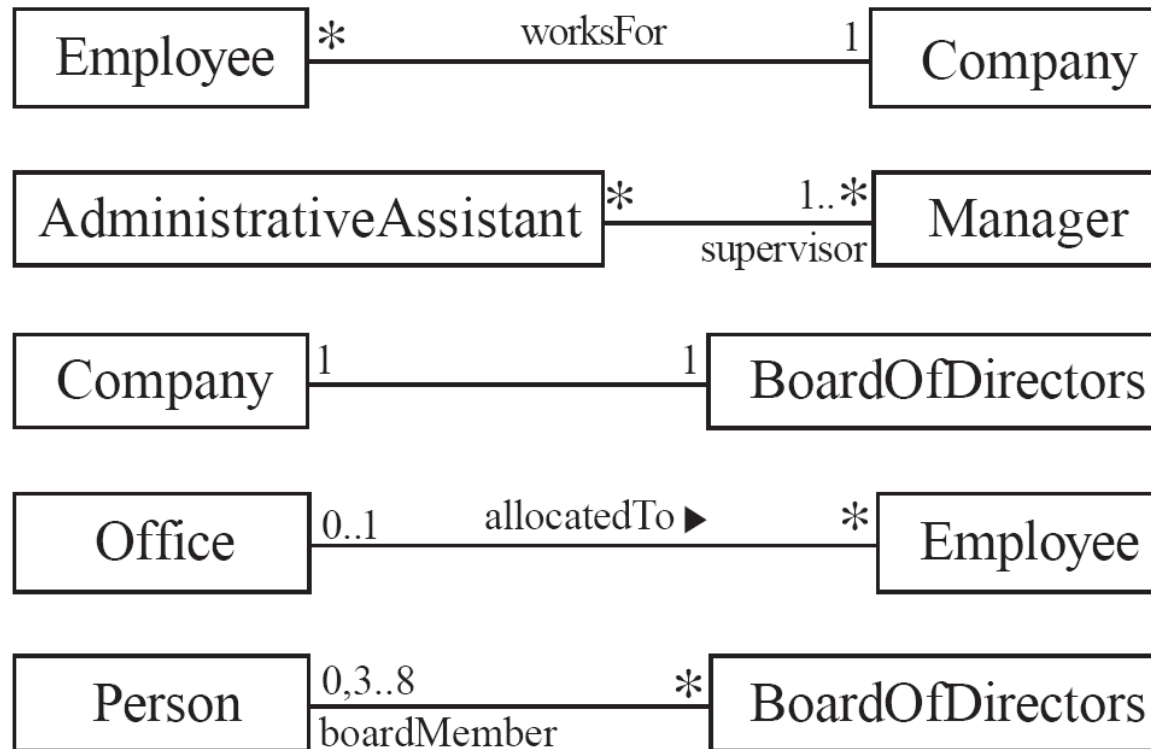


# Home Heating System



- Room has 1 Thermostat
- Each Thermostat is associated with 0 or more Heaters
- A Heater has exactly one Thermostat
- ElectricHeater is a specialized Heater
- AubeTH101D is a specialized Thermostat

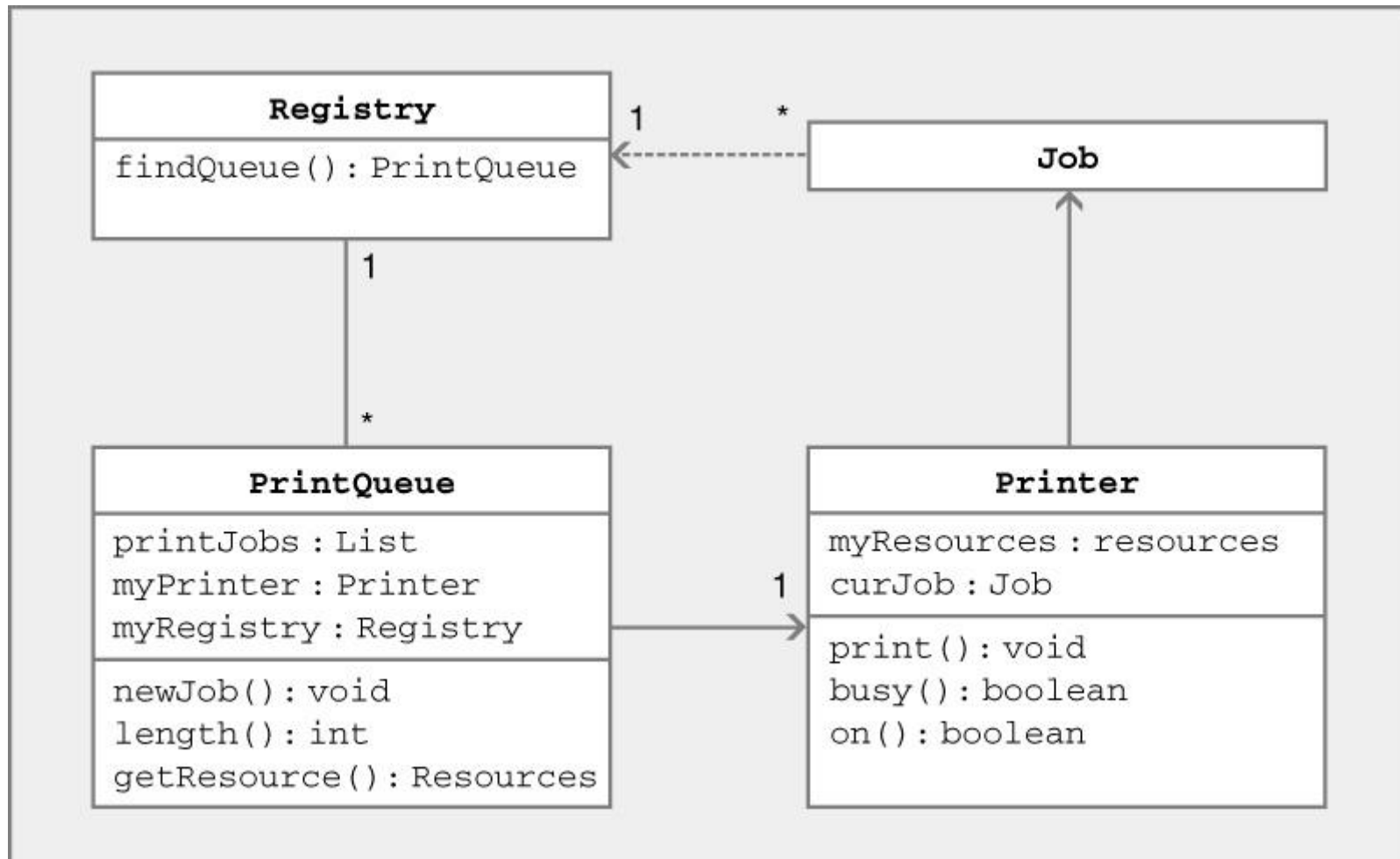
# Company



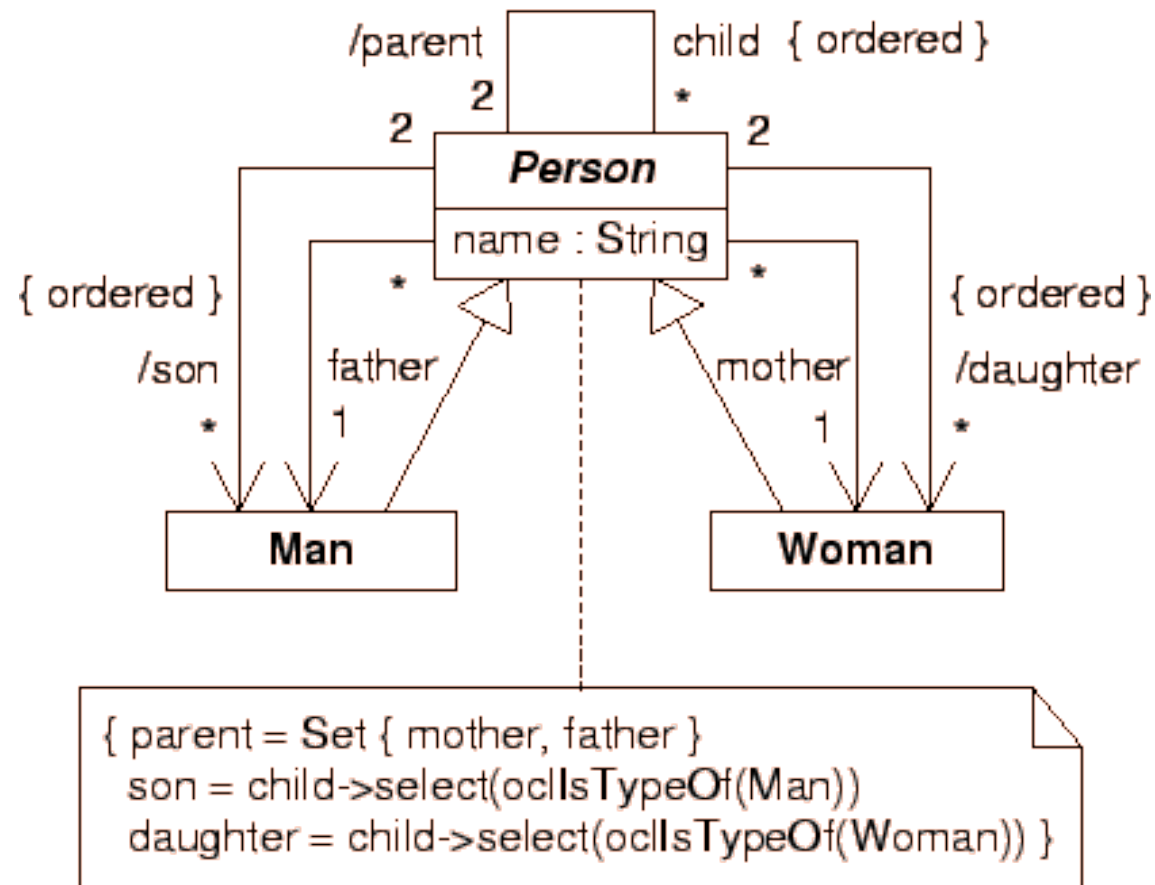
- Each employee works for one company (which can have 0 employees)
- Each AdministrativeAssistant has one or more supervisors (who can have 0 or more employees)
- Each Company has exactly one BoardOfDirectors (and viceversa)
- Each Office is allocated to zero or more Employees (an Employee can have no office or at most one)
- A Person is boardMember of 0 or more BoardOfDirectors (each BoardOfDirectors has from 3 to 8 Persons)



# Printing System



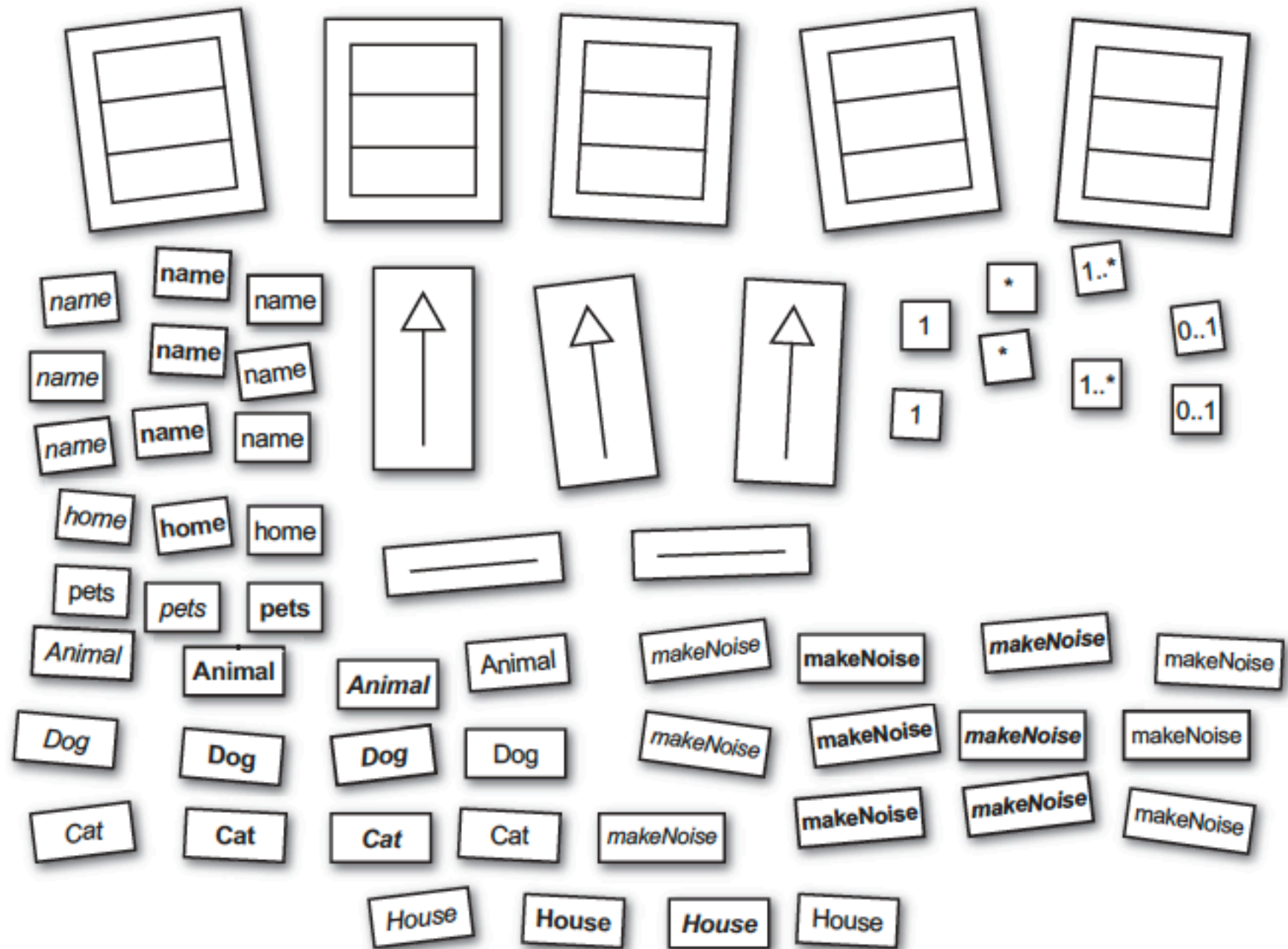
# Family



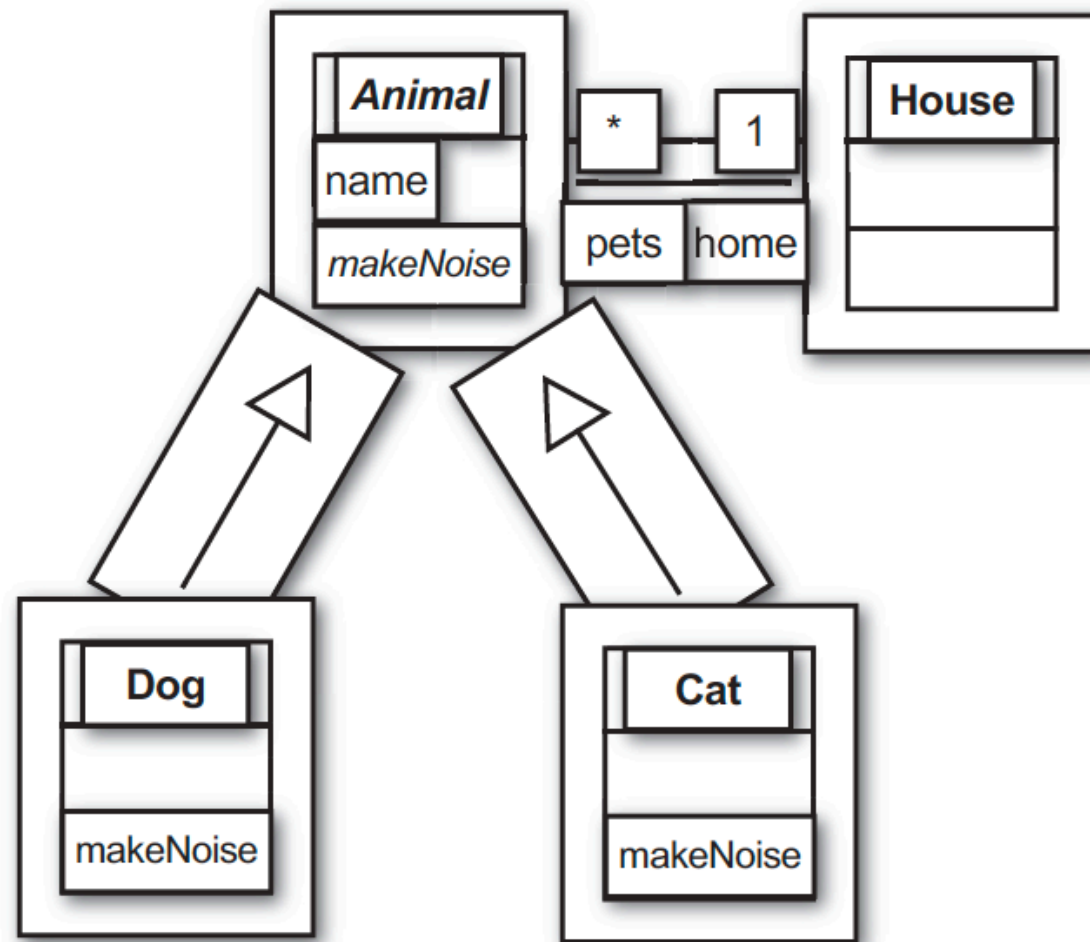
A woman can have as sons two twins?

# On class diagrams

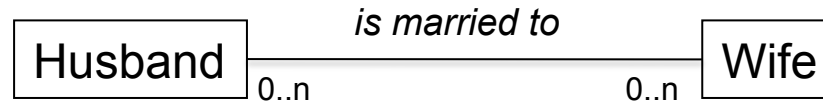
- A house may have any number of pets living in it
- The two possible types of pets that can live in a house are dogs and cats
- Each dog or cat has a name
- An animal's house is its one and only home
- You can tell an animal to make noise and it will do its thing



# Solution



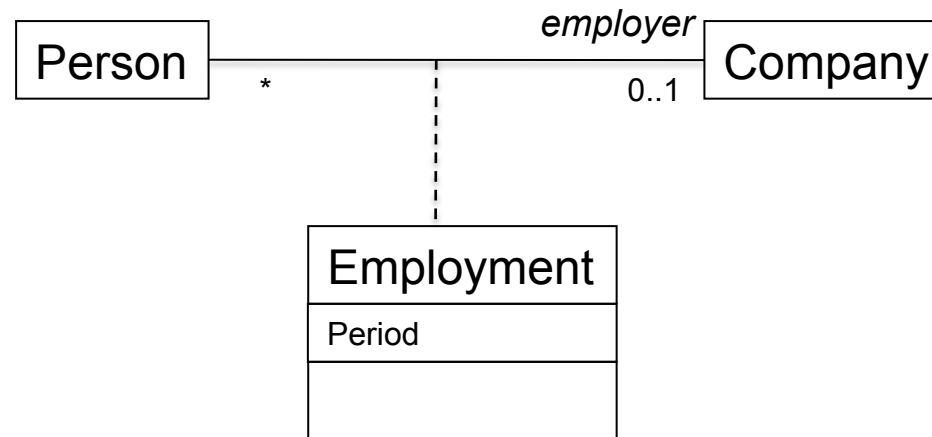
# On class diagrams



**Given that having multiple spouses at the same time is prohibited, but it is allowed to have many spouses over time ...**

- a) The diagram represents the concepts to be modeled and no changes are required
- b) Use the <<history>> stereotype to show the history over time and snapshot in time constraint in the model.
- c) Promote the “is married to” association to an association class called Marriage and break up the Many-to-Many association between Husband and Wife
- d) Create a link attribute called “wedding date” and attach it to the association, rather than to either Class in the Association
- e) Create a link attribute called “wedding date” and attach it to either husband or wife class

# On class diagrams



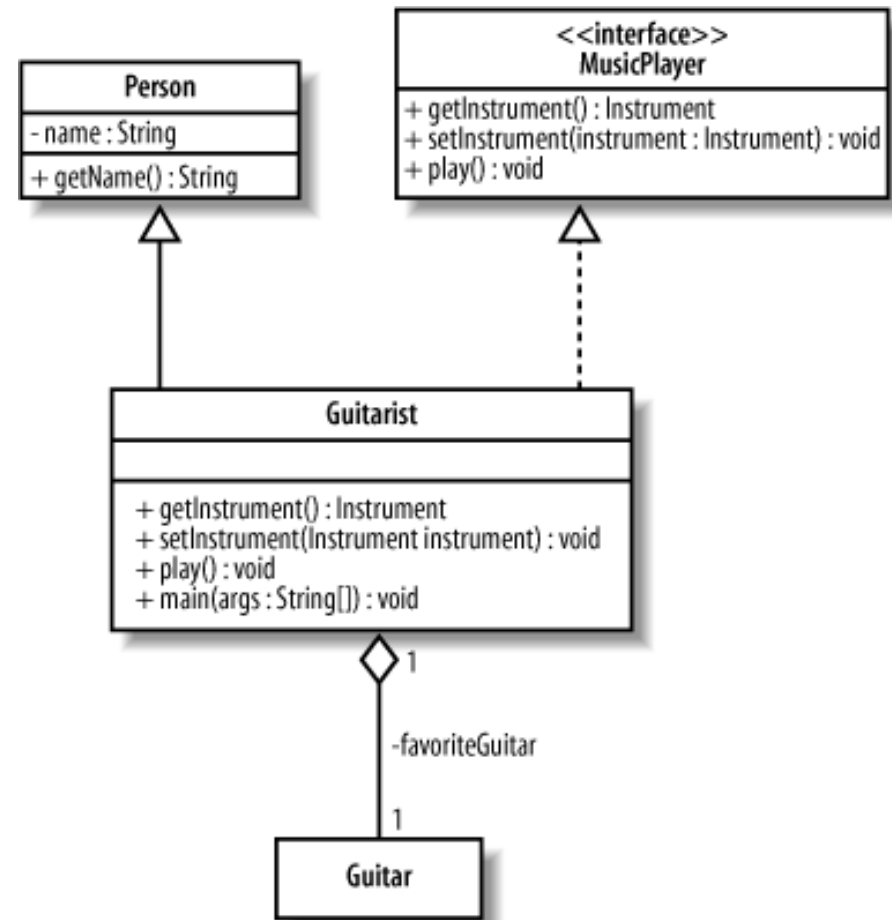
Which sentences are coherent with this model?

- a) A company may employ 0 or 1 person
- b) A person may work for a single company
- c) A person has one employment
- d) A company has one employer that is a person
- e) A company may have zero employers

# On class diagrams

This diagram says that objects:

- a) Persons have a name
- b) Guitarists have a name
- c) Guitars have a name
- d) MusicPlayers have a name

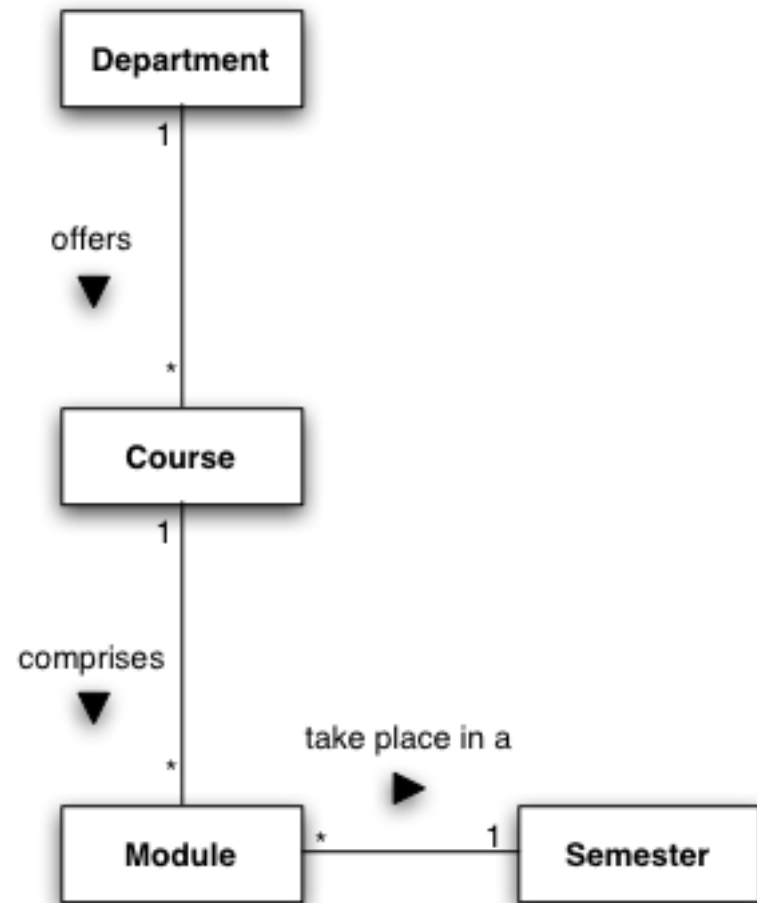




# On class diagrams

This diagram describes

- a) a sequence
- b) a domain
- c) a system
- d) a use case
- e) a temporal behavior



# On class diagrams

Which of the following aspects is not part of a design class diagram?

- a)Attributes with their types
- b)Operations with arguments and results
- c)Composition relationship
- d)Events and actions
- e)Visibility information
- f)Navigation information

# On class diagrams

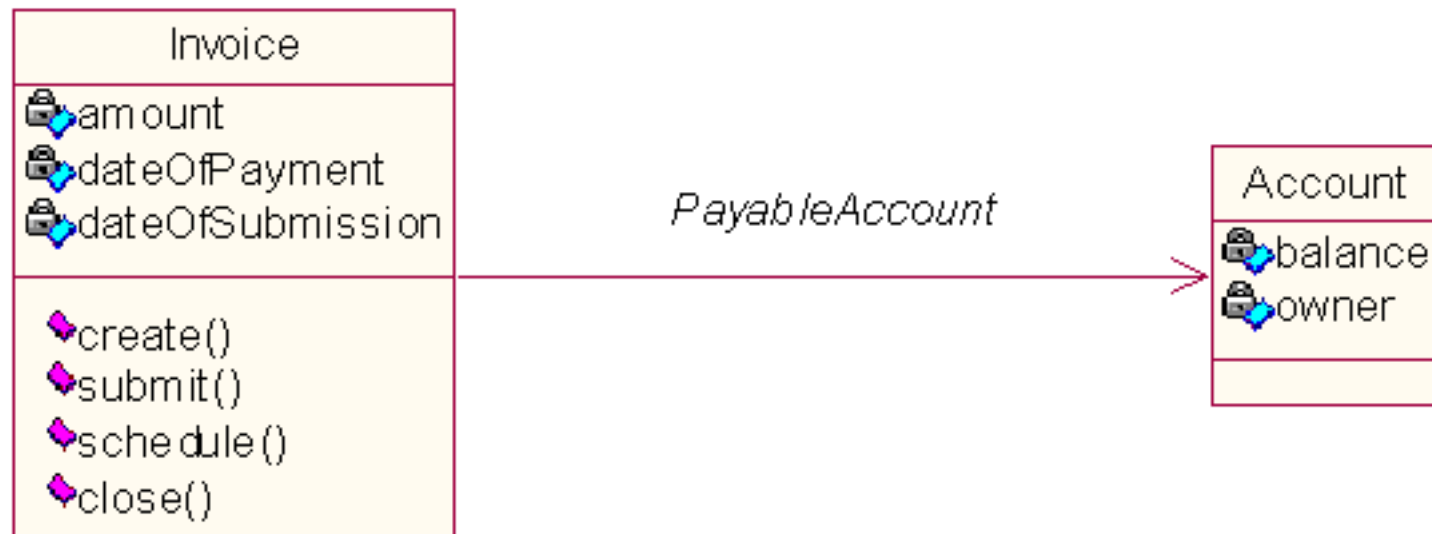
Classify the following into *generalization* (G), *association* (A), *aggregation* (AG), or *composition* (C):

- a) A country has a capital city
- b) A dining philosopher uses a fork
- c) A file is an ordinary file or a directory file
- d) Files contain records
- e) A class can have several attributes
- f) A relation can be association or generalization
- g) A polygon is composed of an ordered set of points
- h) A person uses a computer language on a project

# On class diagrams

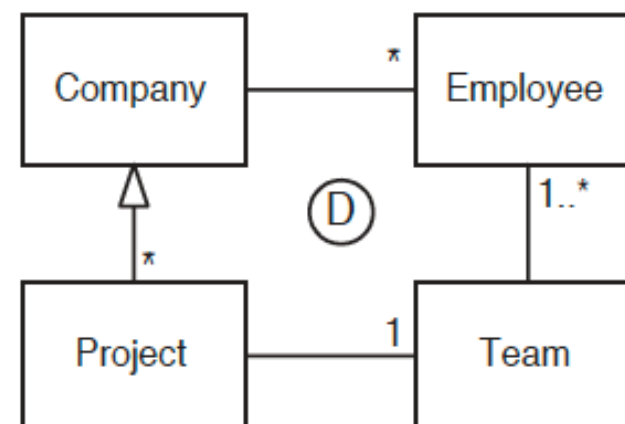
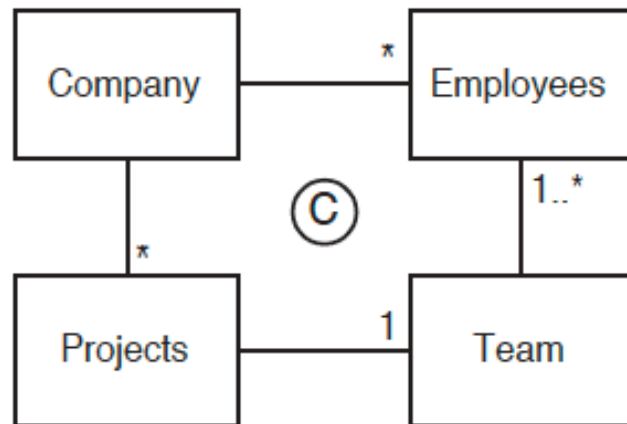
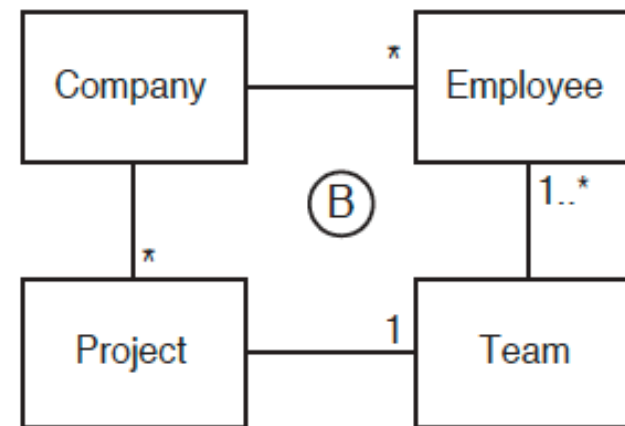
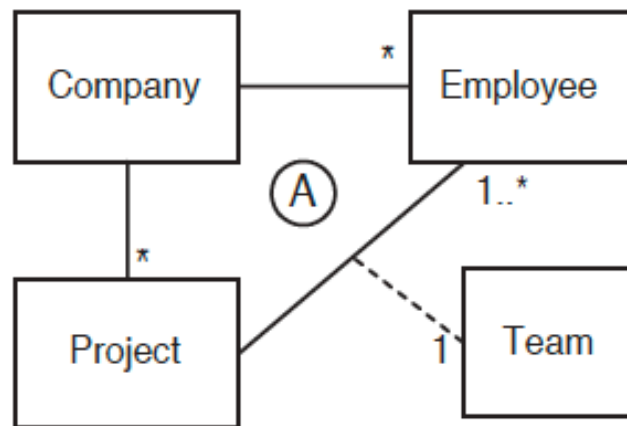
The arrow between the two classes indicates:

- a) Inheritance
- b) Association
- c) Dependency
- d) Sending a message



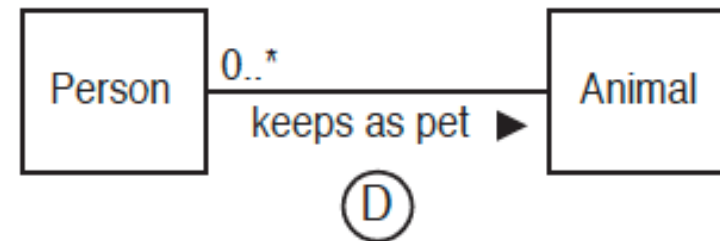
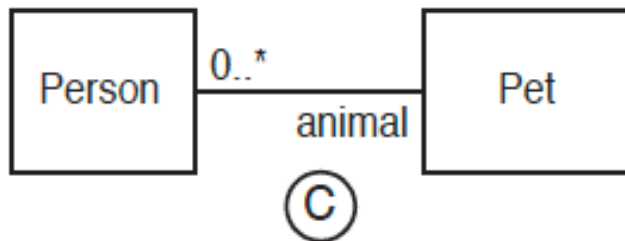
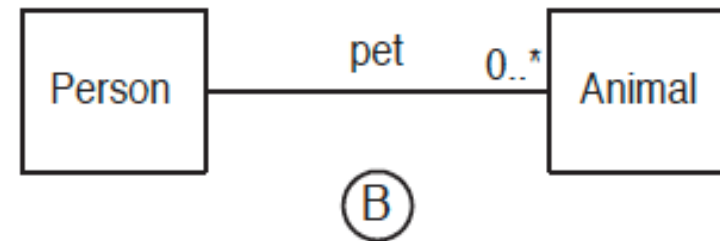
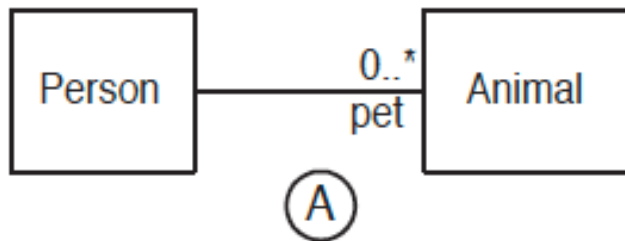
# On class diagrams

Consider the following situation:  
a company realizes projects; each project is executed by a team of employees.  
Which would be a suitable conceptual UML diagram?



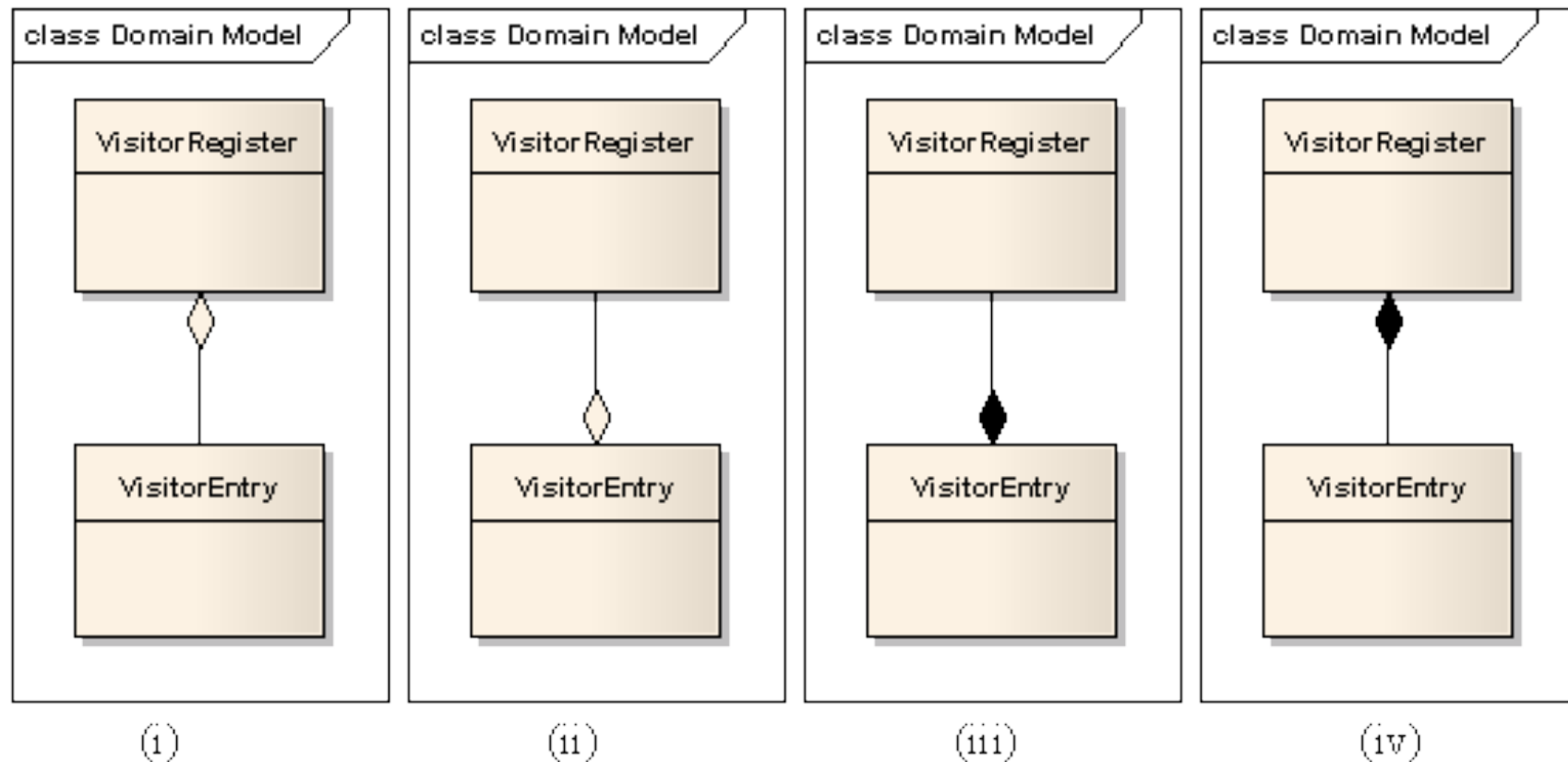
# On class diagrams

How do you express that some persons keep animals as pets?



# On class diagrams

A visitor register is made of visitor entries



# On class diagrams

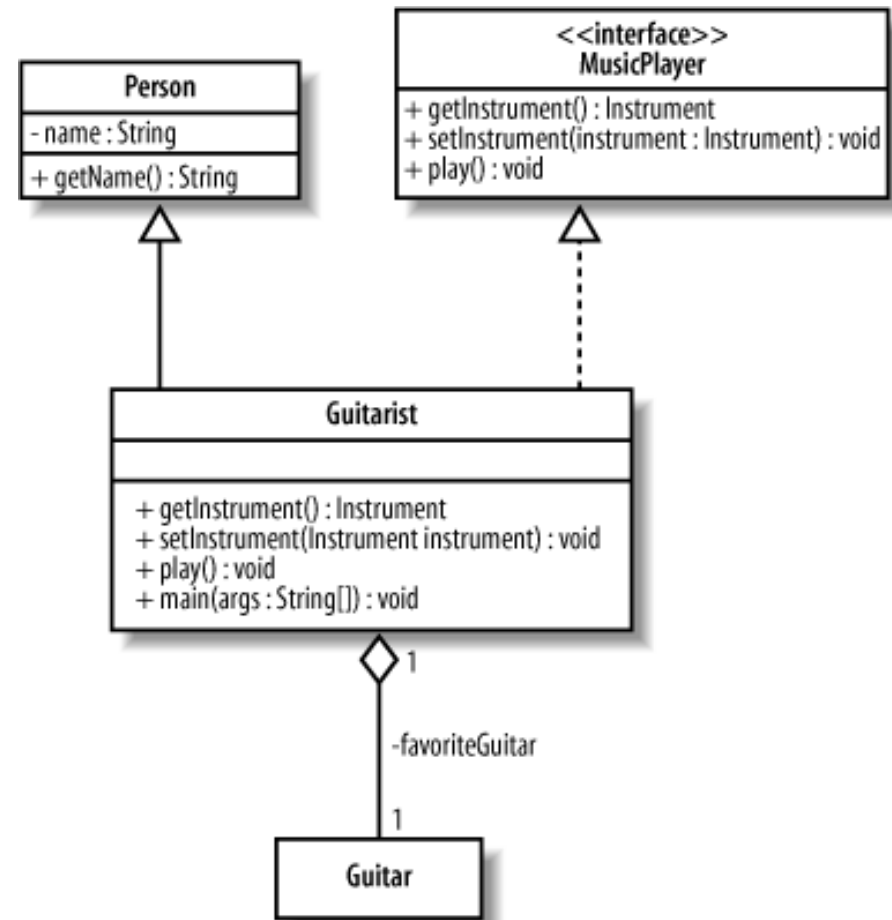
This diagram says that objects having a “main” method are

a)Persons

b)Guitarists

c)Guitars

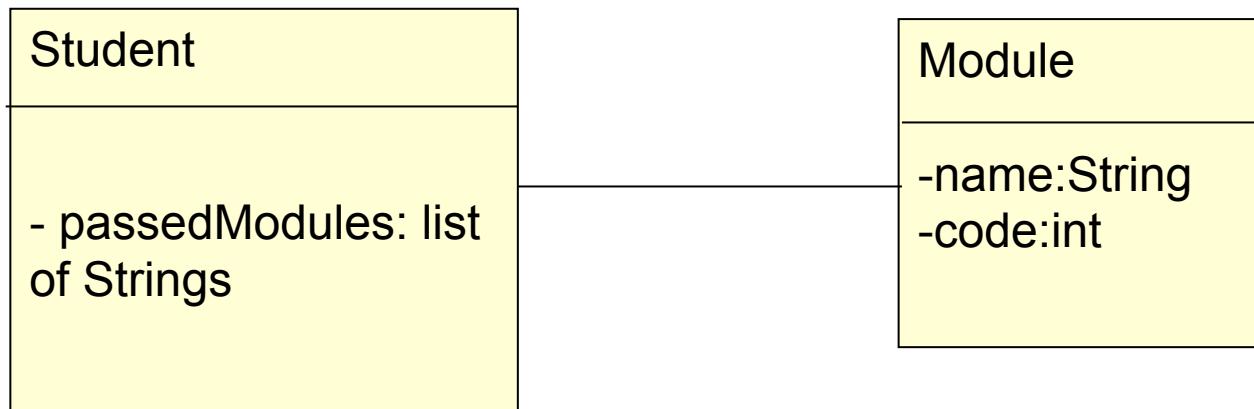
d)MusicPlayers, when implemented by Guitarists





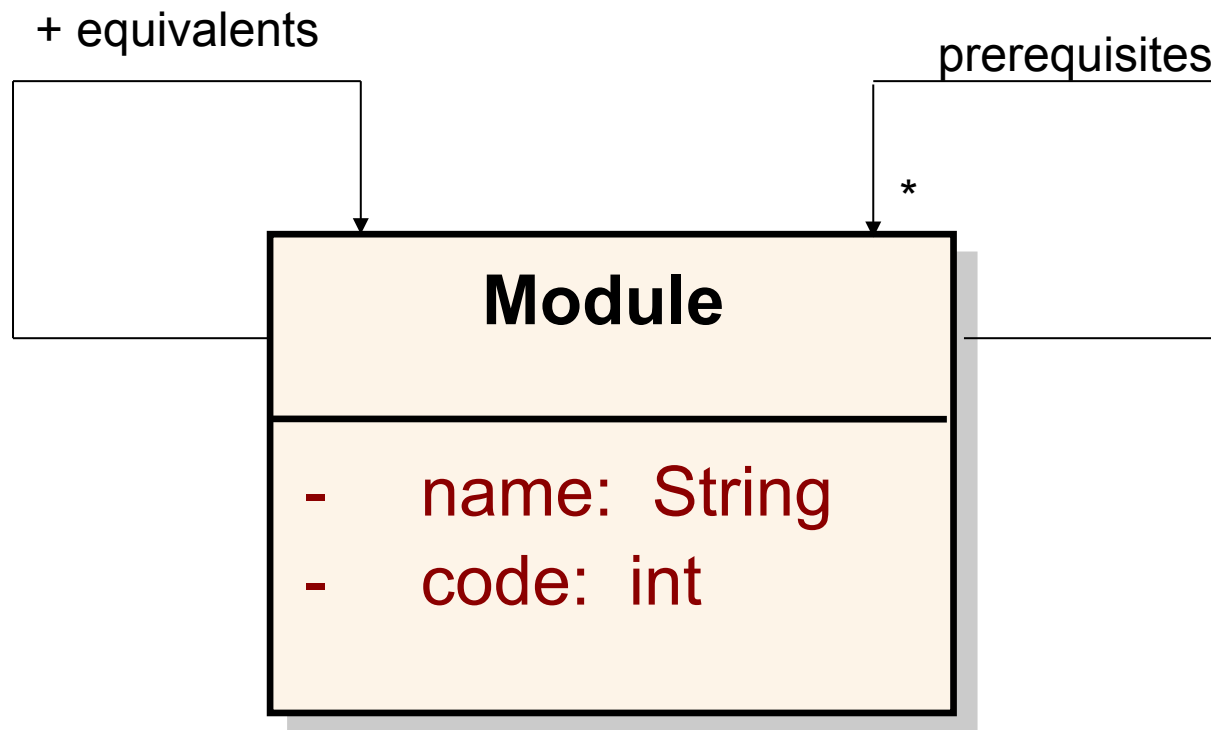
# On class diagrams:

- Right or wrong?



# On class diagrams

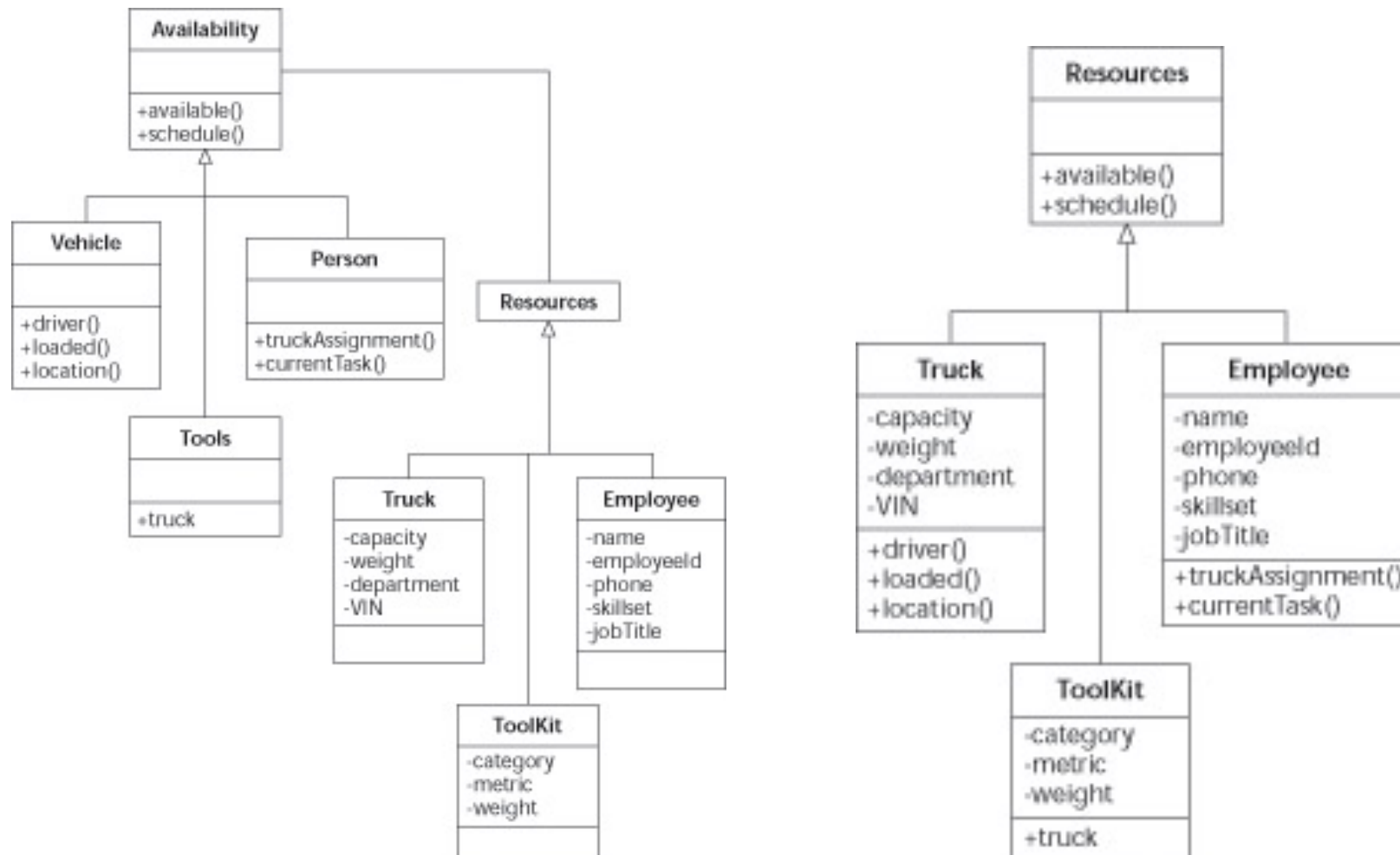
- Right or wrong?



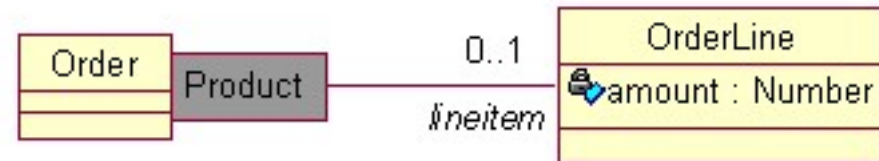
# On class diagrams

Examine these two diagrams; true or false?

- a) The right diagram has more information
- b) The left diagram has more information
- c) They are equivalent, but the right one is simpler
- d) They are equivalent, but the left one is simpler



# On class diagrams



This association could be translated in an interface like:

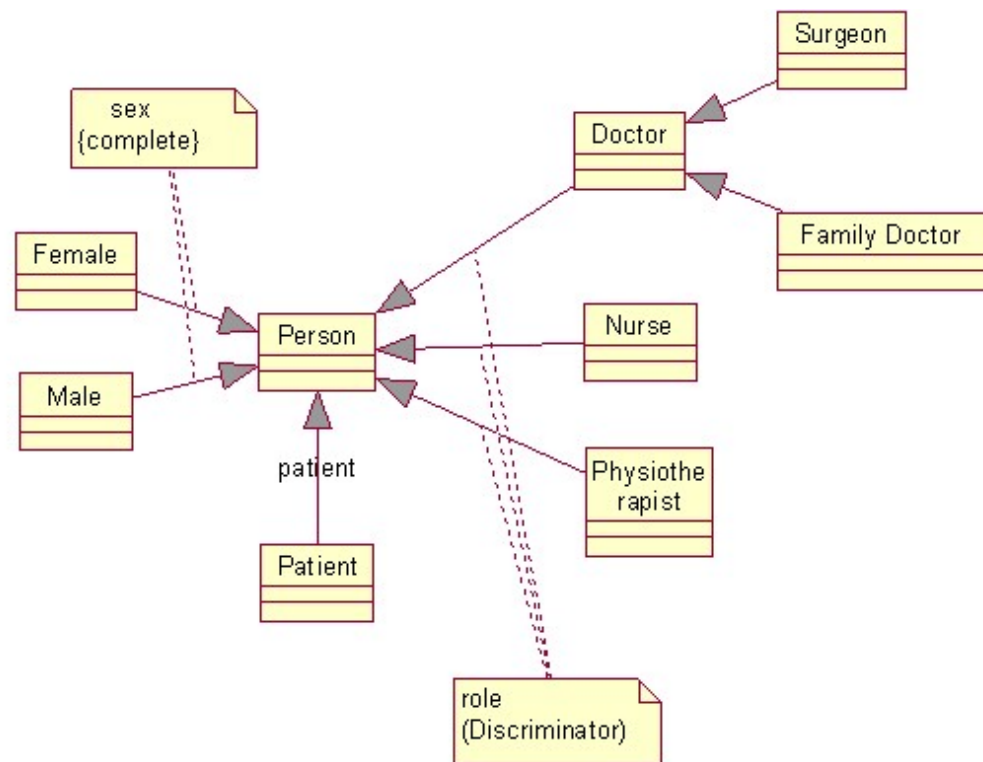
- a) `class Order { public OrderLine getLineItems() ;  
public void addLineItem(Number amount);`
- b) `class Order { public OrderLine getLineItems() ;  
public void addLineItem(OrderLine lineItem);`
- c) `class Order { public OrderLine  
getLineItems(Product aProduct) ; public void  
addLineItem(Number amount, Product aProduct);`
- d) `class Order { public Object getLineItems() ;  
public void addLineItem(OrderLine lineItem);`



# On class diagrams

Which of the following are **illegal** combinations of subtypes in the diagram?

- a) Female, Patient, Nurse
- b) Male, Physiotherapist
- c) Female, Patient
- d) Female, Doctor, Surgeon
- e) Patient, Doctor
- f) Male, Doctor, Nurse



# On class diagrams

A benefit of using polymorphism is a reduction of:

a) methods in the associated classes

b) subclasses needed to accomplish the same functionality

c) case statements and conditionals

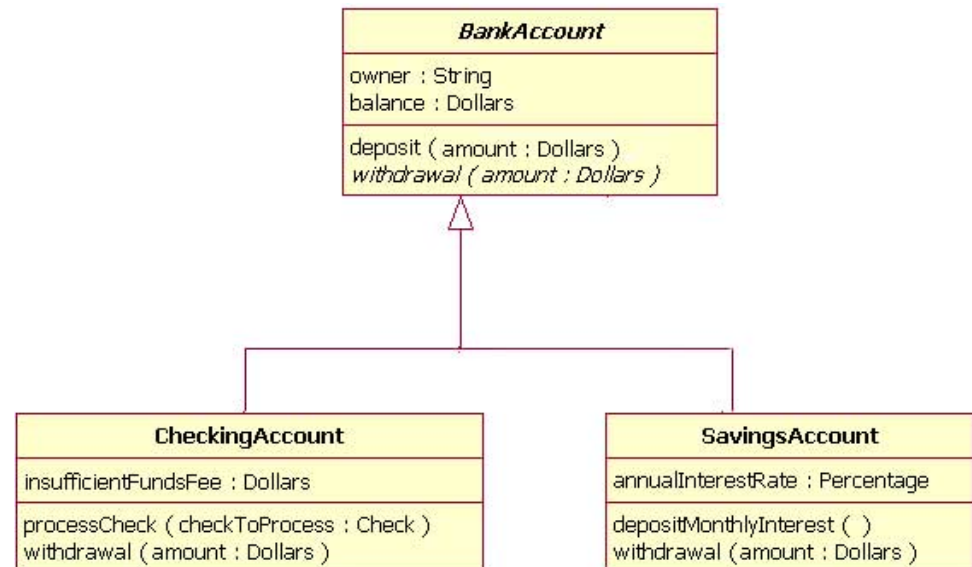
d) coupling between classes in the system



# On class diagrams

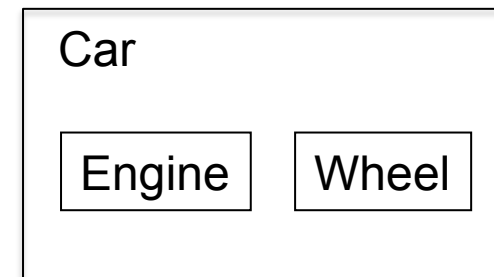
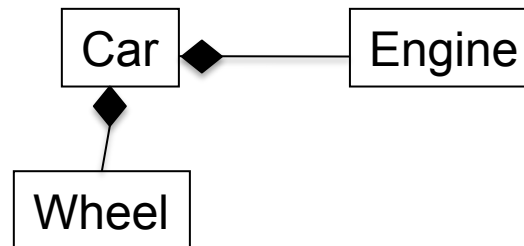
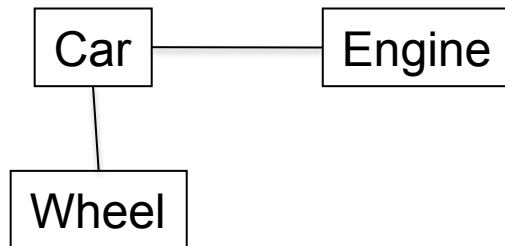
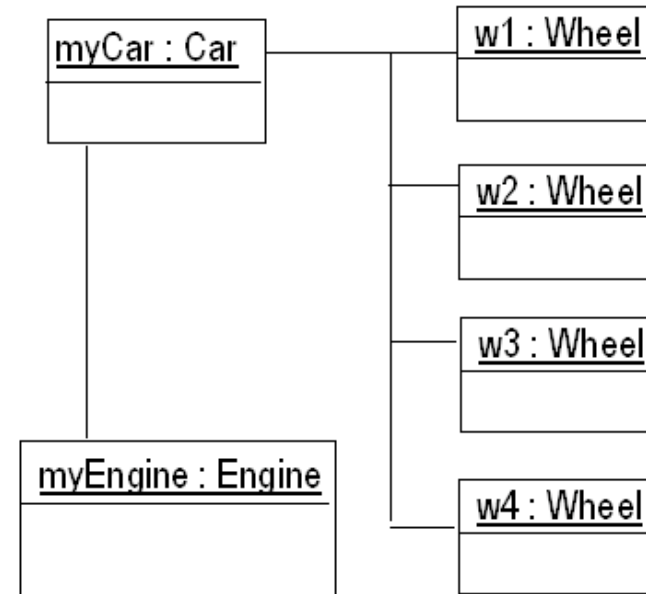
Which sentences are true?

- 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



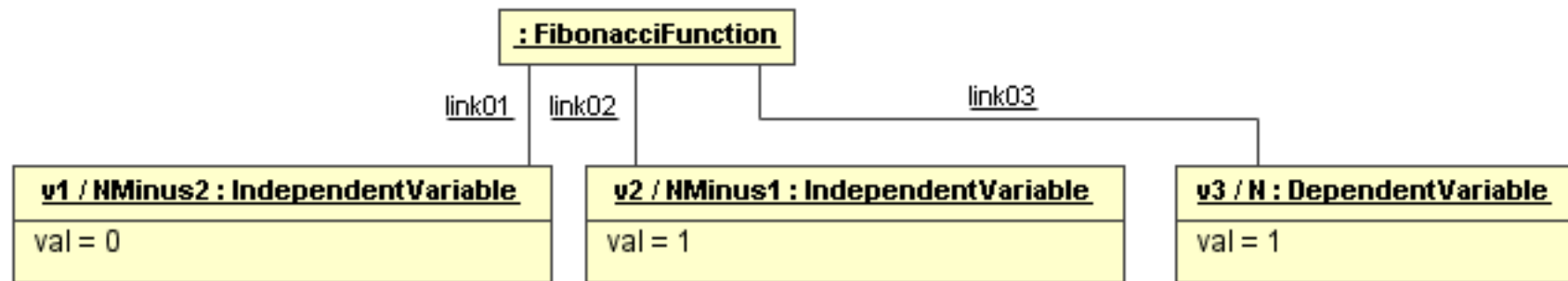
# On object diagrams

- This object diagram is instance of which class diagram(s)?





# On object diagrams



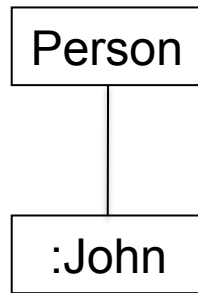
This object diagram

- a) Includes an anonymous object
- b) Includes a class
- c) Includes four instances
- d) Includes three instances

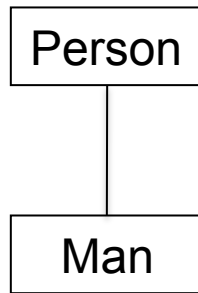


# On object diagrams

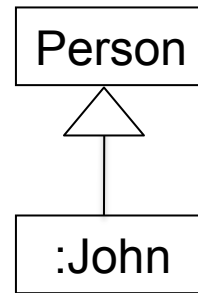
Which diagrams are correct?



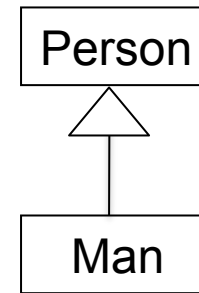
a)



b)

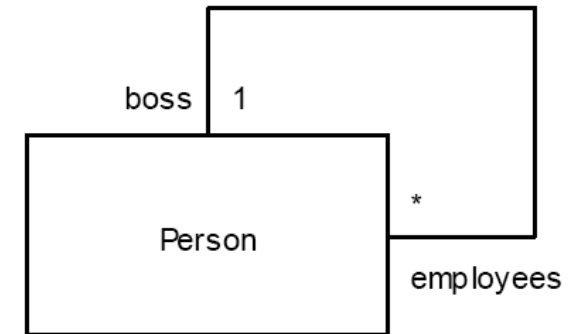


c)

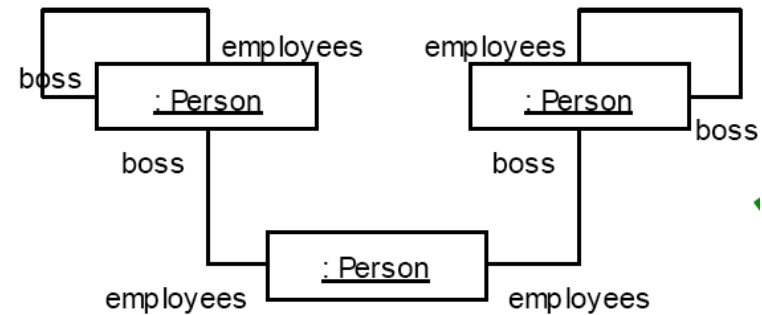
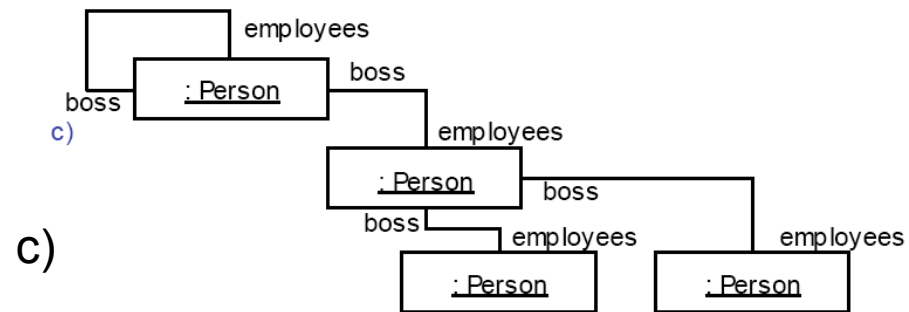
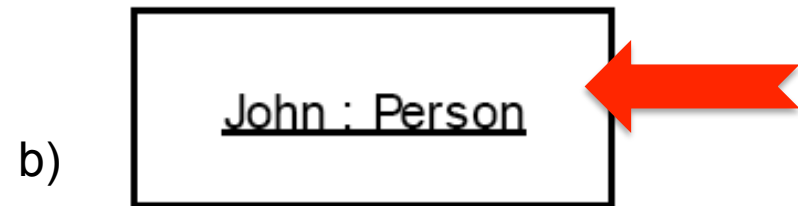
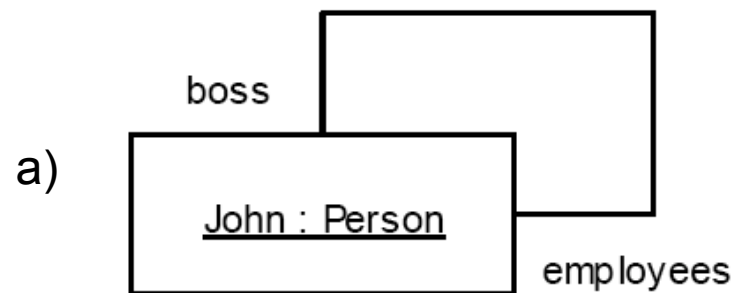


d)

# On objects diagrams

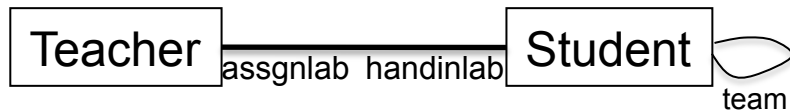
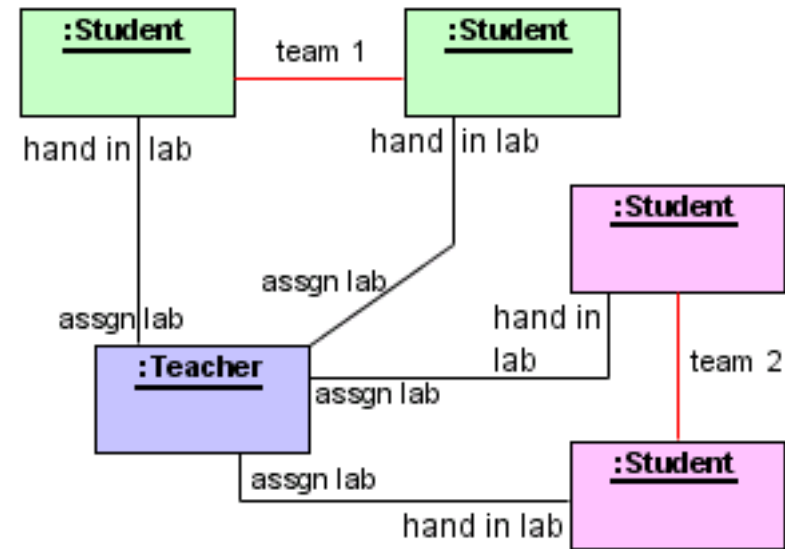


Which one is a non valid instance of this model?

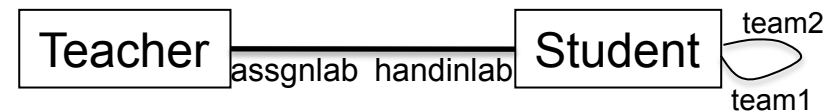


# On object diagrams

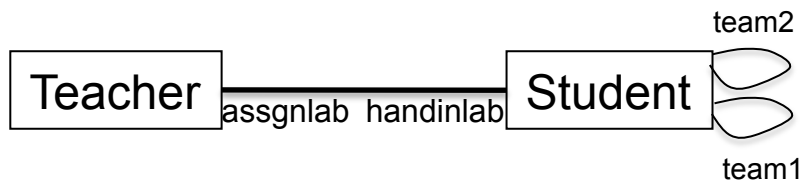
- Which is the class diagram which best corresponds to this object diagram?



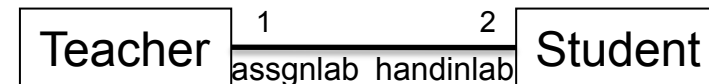
a)



c)



b)



d)

# Do-it-yourself exercises

**Use case diagram** describe how to play chess via an interface able to connect either different engines or a chess server via internet

**Class diagram** describe the pieces, chessboard and the game tree

**Object diagram** describe by object snapshots a chess position during a game

**Activity diagram** describe a game workflow including two players playing via a (telnet) chess server

**Statechart** describe a game workflow from the viewpoint of the chessboard

**Sequence diagram** describe a multiagent system evaluating a position

**Communication diagram** describe a multiagent system evaluating a position

# References

[www.pearsonvue.com/omg/](http://www.pearsonvue.com/omg/)

[www.objectsbydesign.com/projects/umltest/bparanj-answers-1.html](http://www.objectsbydesign.com/projects/umltest/bparanj-answers-1.html)

[dn.codegear.com/article/31863](http://dn.codegear.com/article/31863)

[parlezuml.com](http://parlezuml.com)

[www.proprofs.com/quiz-school/story.php?title=quiz-uml-20](http://www.proprofs.com/quiz-school/story.php?title=quiz-uml-20)

Think about it!

