

CSE70: Lecture 6

- Today
 - Second release deadline now!
 - Did you submit your feedback?
- Final Checklist
 - Tests and Coverage
 - All the tasks of the labs
 - ...
- For Today:
 - UML and Design (from the reader)

Design

- Conception of a scheme that represent and specifies a software product
- Design involves
 - Requirements
 - Limitations
 - Tradeoffs
 - Creativity
 - Heuristics

Unified Modeling Language

- Graphical notations
 - To *model* software...
 - and aspects of software
- Why a model and why graphical?
 - Visualization
 - Specification
 - Construction
 - Documentation
- What makes a good model?

Views & Diagrams

- Use case
 - Design
 - Process
 - Implementation
 - Deployment
- Use case
 - Class
 - Object
 - Interaction
 - Sequence
 - Activity
 - Deployment
 - ...

Object Oriented Design

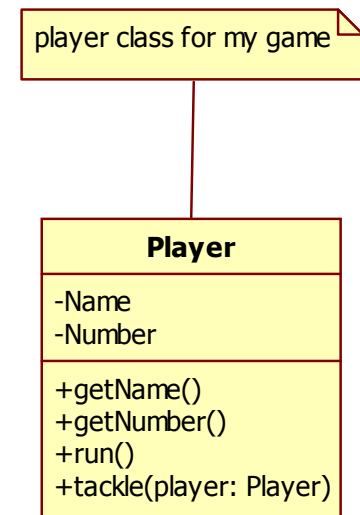
- Class
 - *Abstraction* of an object's characteristics
 - With data
 - And actions
- Attribute
- Method
- Object

A player has a name,
and a number. A player
can run.



Class

- In a UML a class has properties
 - Attributes
 - Operations (methods)
 - With visibility, name, type, ...
- Defining class responsibilities
 - Identifying *what it knows*
 - Identifying *what it does*
 - And *what is shared*
- Abstract Data Type (ADT)

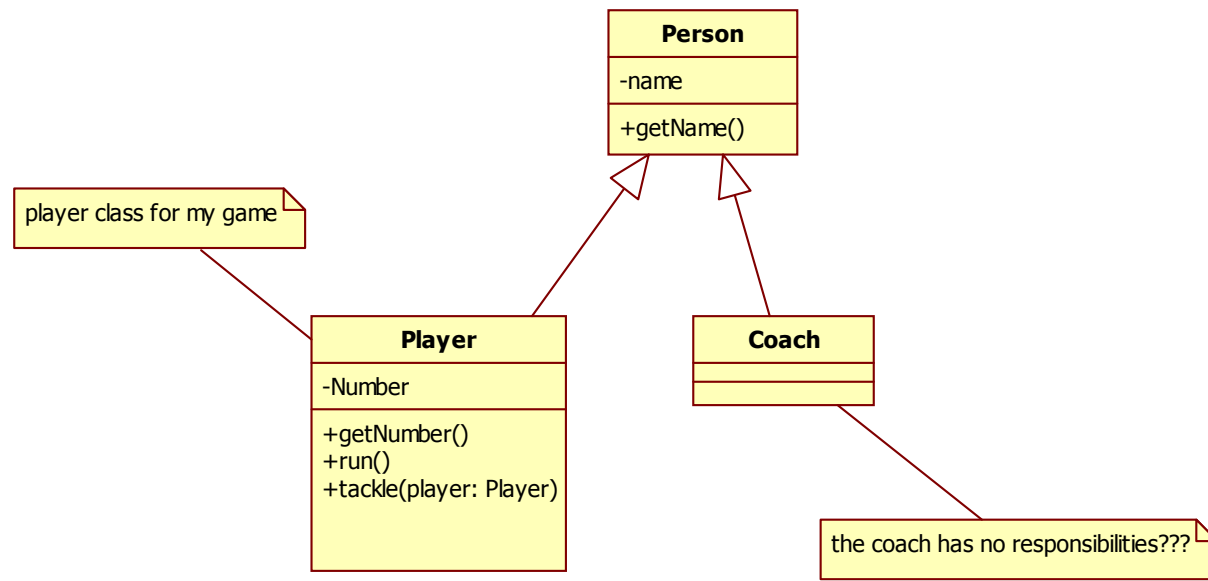


Design Issues

- A system definition
 - Collection of classes (e.g. *class diagram*)
- Abstraction
- Encapsulation
 - *visibility*
- Information hiding
 - *And visibility*
- Dealing with change

Inheritance

- Share common *properties*
 - Promote reuse and avoid duplication
- Define an *is-a* relationship



Inheritance

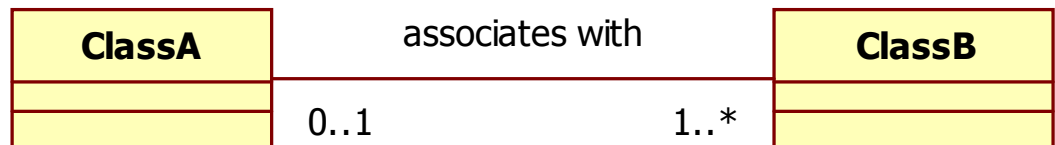
- When to use inheritance?
 - Look for similarities
 - Look for existing classes
 - Follow the *sentence rule*
 - Avoid implementation inheritance
 - Inherit everything
- Single vs Multiple inheritance
- Abstract vs Concrete classes

Association

- Interactions between classes
- Associations can have

- *label*

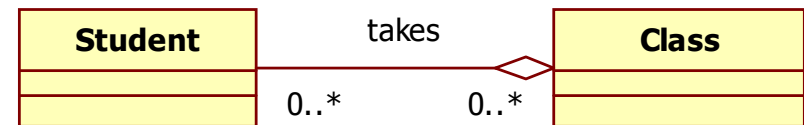
- *multiplicity*



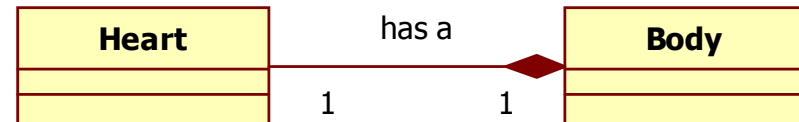
- Associations can be
 - *recursive*
 - *bidirectional*

Aggregation & Composition

- Aggregation
 - *is part of* relationship



- Composition
 - *is part of* relationship
 - No sharing
 - Same life span



Summary

- UML
 - Views and diagrams
- Basic OO Design
 - Class
 - Abstraction
 - Encapsulation
 - Information hiding
 - Inheritance
 - Association(s)

Let's look at the project