Design Class Diagrams

PB007 Software Engineering I

Bruno Rossi

25. 11. 2015



Software Engineering I (PB007)

A **Class Diagram** gives a static view of the classes, their attributes, operations and relationships.

Analysis Class Diagram

- business model of the domain object types and relationships
- the effort is to maintain clarity and simplicity without clogging with implementation details.

Design Class Diagram

• the analysis model classes and the implementation details of the classes.



A **design class** provides a level of abstraction such that it can be easily implemented.

can come from:

- Business domain including details at the analysis level (decomposition into more classes, complement implementation details).
- domain technical classes classes required by the technology used (classes for working with GUI, DB, ...)

Implementation details include:

- Visibility and type attribute.
- Visibility, arguments, return type from methods.
- Methods added to the analysis operations, such as constructors (destructors), getter/setter methods, implementation methods.



Design Classes - Example

analysis	design	
BankAccount	BankAccount	
name number balance	-name : String -number : String -balance : double = 0	
deposit() withdraw() calculateInterest()	+BankAccount(name:String, number:String) +deposit(m:double) : void +withdraw(m:double) : boolean +calculateInterest() : double +getName() : String +setName(n:String) : void +getAddress() : String +setAddress (a:String) : void +getBalance() : double	



- More advanced association types for implementation details: aggregation or composition.
- Are generally defined with a name, navigability and multiplicity.
- Decomposition of bidirectional associations.
- Type of associations 1:1, 1:M, M:1.
- Decomposition of associations M:N.
- Decomposition of association classes.



Aggregation

Aggregation is a type whole-part relationship.

- The whole may or may not exist without its parts
- Parts can exist independently from the whole
- The whole is in a sense incomplete if some parts are missing.
- Part may theoretically be shared by several units.
- Aggregation is transitive and asymmetric (without cycles).





Composition

Composition is a stronger form of aggregation

- At a specific time parts can only belong to one group (they cannot stand alone).
- The whole is responsible for the creation and deletion of the parts.
- If the whole is deleted, it must either delete all its parts, or shift responsibility for them to another object.
- The composition is asymmetric and transitive (without cycles).





Revision of 1:1 associations

Analysis:



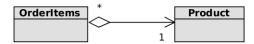




Revision of M:1 associations

Analysis:



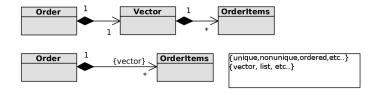




Revision of 1:M associations

Analysis:



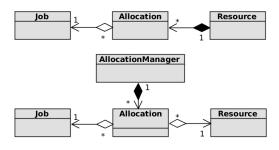




Decomposition of M:N associations

Analysis:

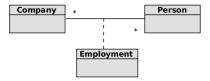






Decomposition of association classes

Analysis:



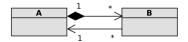




Decomposition of bi-directional associations

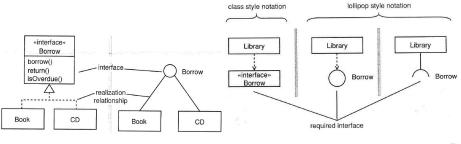
Analysis:







Interfaces are special classes that define a set of public services, attributes and relationships, but do not implement them. They are used to define the contract that classes provide.





Tasks

- Extend the analysis model into the design model by using class diagrams.
- Specify visibility and type of all attributes.
- Add methods that originated from the decomposition of analysis operations, implementation and support methods (constructors, getter / setter methods, ...), determine their visibility, arguments and return types.
- Please specify further the analysis associations (with naming, multiplicity, navigability, **aggregation / composition**, decomposition of association classes and M: N associations)
- Fill relations of dependencies among classes.
- If necessary, add other implementation classes or interfaces
- Upload the **PDF report** into folder (Week 09). **Deadline:** 30.11.15 23:59 (Groups 2,3)



Customization of PDF Reports

tions		Details	
Generate table of contents	?	Children	References References
	C C C C C C C C C C C C C C C C C C C	C Diagram-based Members ERD Column Details Properties Project management properties Relationships	Soft days and soft of the sof
7 Generate reference (file/URL) link Generate model elements/dlagrams link Skip heading for empty model element section Convert multiline model heading to single line Show multiline model having Treat HTML content as HTML source Suppress details if duplicated Table cell keep together with page		Quality information Anti-alkasing Graphics Font Font: Unspecified	₽ Text
Irrap : Word wrap hape type style : Icon TF content appearance : Preserve formatting	2		