Analytical class diagram II

PB007 Software engineering I

Stanislav Chren

18. 10. 2021

Relationships between classes

The basic relationships include:

- Generalization
- Association
- Dependency

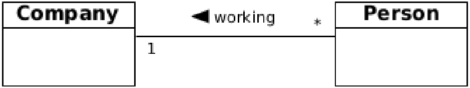
Association

Asociation is a relationship between classes.

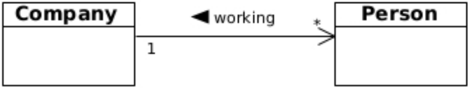
- represents a long-time relationship (i.e. there is a reference between instances of the given classes)
- the reference is often expressed by the presence of the attribute having the type of the other class
- relationship 1:1 corresponds to a simple attribute reference
- relationship 1:N corresponds to an array or collection
- the direction of the association (navigability) indicates the class that will contain the attribute.

Association - Navigability

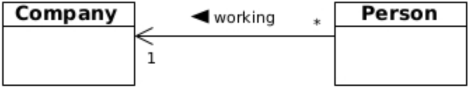
Both Company and Employee have references to each other



The Company has a reference (collection attribute) to its Employees. The Employee does not have reference to the Company



The Company does not have reference to its Employees. The Employee has an attributed ot the type Company as reference.

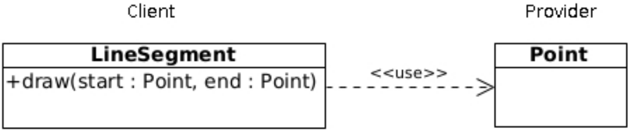


Dependency

Dependency is a relationship between two classes(client and provider), where change in provider may force change in the client. In other words, the client somehow depends on the provider.

The semantics of this dependency can be specified by different stereotypes.

The most common stereotype is ≪`use`≫. It indicates that some of the client operations use the provider object as input argument or as return value, or it is used in its implementation and at the same time, there is no attribute/reference to that class.



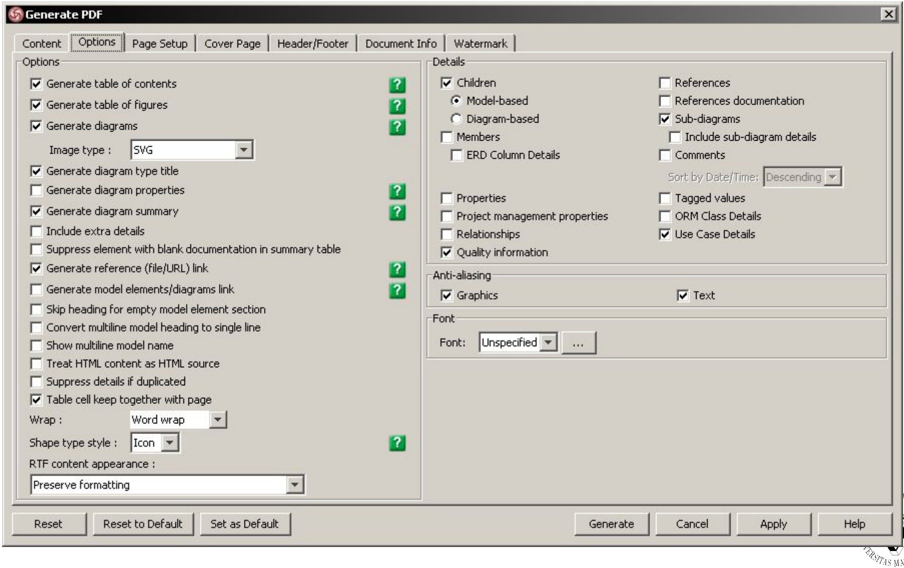
Tasks

- Finalize the analytical class diagram
  - The diagram should contain the **analytical classes, attributes, operations** and the different relationships (**inheritance, association, dependencies**)
  - The associations should specify **name, multiplicity** and **navigability**
  - Add manager classes which manipulate with the base analytical classes (maintaining the list of objects, lifecycle management, ...). Move the appropriate responsibilities.
- Update the use case diagram. For each analytical class, there should be a use case that manipulates with its instances.
- Think about the interaction between different class instances. Based on your class diagram, can you represent all use cases?
- Submit **pdf report** to homework vault (**Seminar 06**).
  Deadline:
  - Saturday (Groups 03, 04)
  - Monday (Group 11)
  - Tuesday (Groups 06, 07)
  - Wednesday ([06:00 AM] Groups 08, 09)

Rules for report submission

1. Submit the PDF report, not the VP source file and not an exported image.
2. PDF report must be created using the procedure shown on the seminars including the report settings.
3. The name of the PDF report file should be *lastname1-lastname2-lastname3* of the team members.
4. PDF report must contain all diagrams modelled until now.
5. PDF report must be uploaded to the homework vault by the specified deadline.
6. PDF report must be uploaded to the correct homework vault. The name of the homework vault is always specified on the slides.
7. Each team uploads only a single PDF report for the whole team.
8. Submitted diagrams must be clear and readable.
9. Submitted diagrams should not contain serious mistakes. At least, they should not contain mistakes mentioned in the *Catalogue of common mistakes*.

VP report settings



Generate PDF

Options

- ☑ Generate table of contents
- ☑ Generate table of figures
- ☑ Generate diagrams
- Image type : SVG
- ☑ Generate diagram type title
- □ Generate diagram properties
- ☑ Generate diagram summary
- □ Include extra details
- □ Suppress element with blank documentation in summary table
- ☑ Generate reference (file/URL) link
- □ Generate model elements/diagrams link
- □ Skip heading for empty model element section
- □ Convert multiline model heading to single line
- □ Show multiline model name
- □ Treat HTML content as HTML source
- □ Suppress details if duplicated
- ☑ Table cell keep together with page
- Wrap : Word wrap
- Shape type style : Icon
- RTF content appearance : Preserve formatting

Details

- ☑ Children
  - ● Model-based
  - ○ Diagram-based
- □ Members
  - □ ERD Column Details
- □ Properties
- □ Project management properties
- □ Relationships
- ☑ Quality information
- □ References
- □ References documentation
- ☑ Sub-diagrams
  - □ Include sub-diagram details
- □ Comments
  - Sort by Date/Time: Descending
- □ Tagged values
- □ ORM Class Details
- ☑ Use Case Details

Anti-aliasing

- ☑ Graphics
- ☑ Text

Font

Font: Unspecified