

# Interaction diagrams

PB007 Software engineering I

Stanislav Chren

15. 11. 2021



# Interaction diagrams

**Interaction diagrams** model collaboration between classes/objects/actors during the realization of use cases (or their parts).

**1. Sequence diagram** captures the communication between classes/objects with an emphasis on the temporal ordering of passed messages.

**2. Communication diagram** captures the communication with an emphasis on relationships between classes/objects.

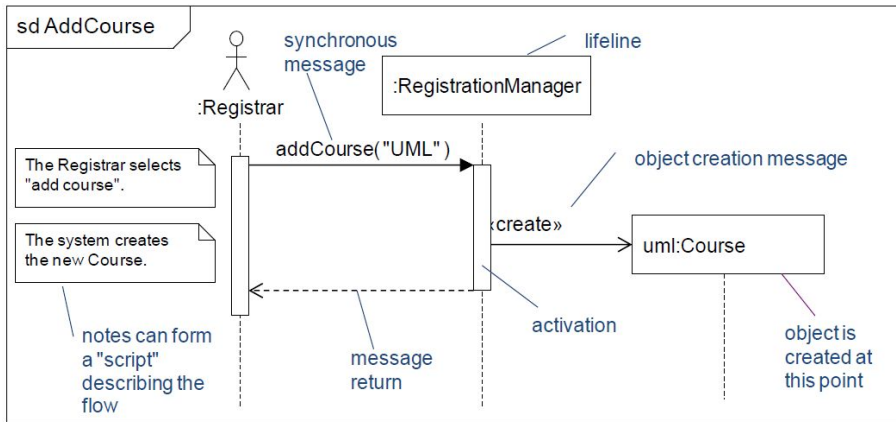


**Sequence diagram** models the interaction as a temporal sequence of messages between objects/classes/actors.

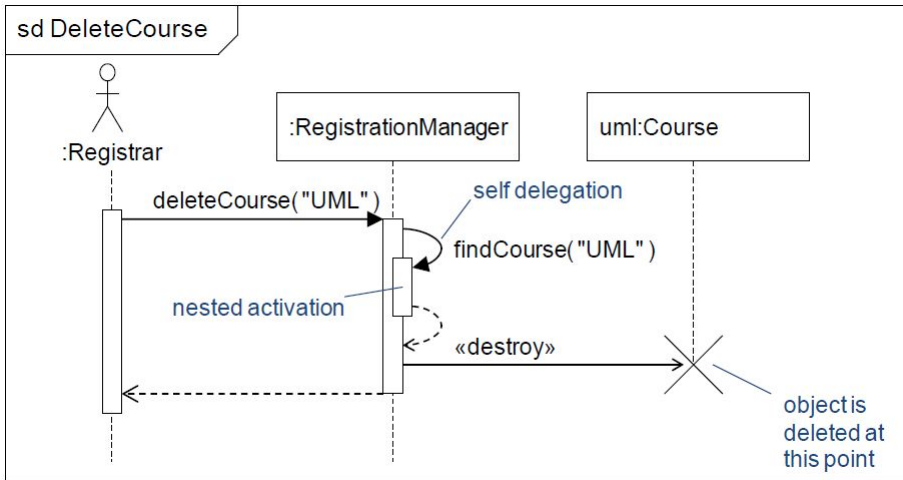
## Basic elements:

- Actors, classes, objects
- Lifelines
- Activations (focus of control)
- Messages
- Fragments

# Sequence diagram - Example



# Sequence diagram - Example II



# Sequence diagram - Combined Fragments

**Combined fragments** divide the sequence diagram into multiple areas with different behaviour

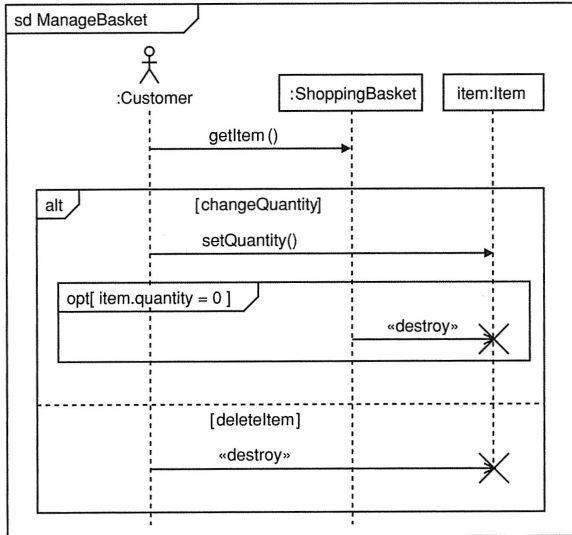
Each combined fragment is marked with *operator*, one or more *operands*, and *condition(s)*.

The most important operators are:

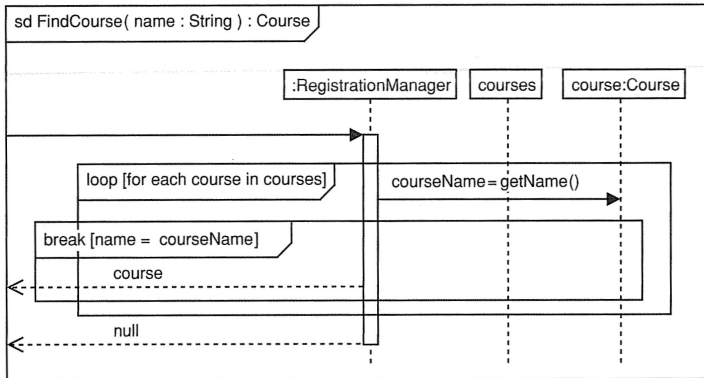
- **opt (option)** - has one operand, which is executed only if the specified condition is satisfied.
- **alt (alternatives)** - has multiple operands, only the one with the satisfied condition is executed.
- **loop** - repeated execution of the operand.
- **break** - operand is executed if its condition is satisfied and it terminates the execution cycle.



# Sequence diagram - Combined Fragments II



# Sequence diagram - Combined Fragments III





# Communication diagram

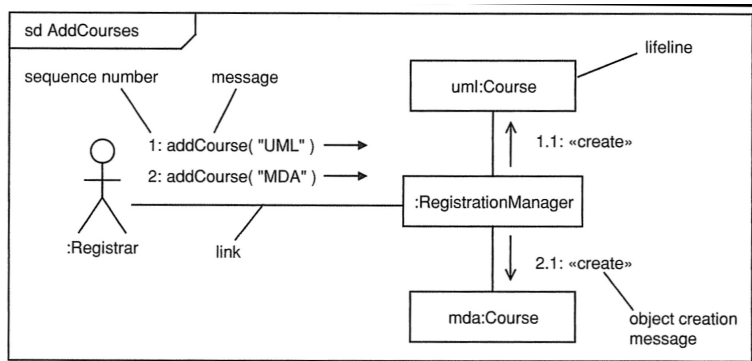
**Communication diagram** captures the interaction as connection between objects/classes/actors and the mutual communication occurring over these connections.

## The basic elements:

- Actors, classes, objects
- Connections (links)
- Messages



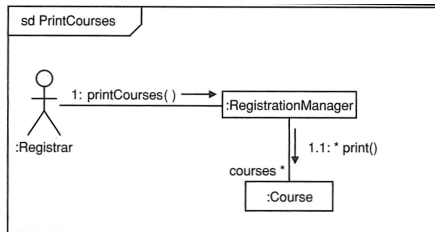
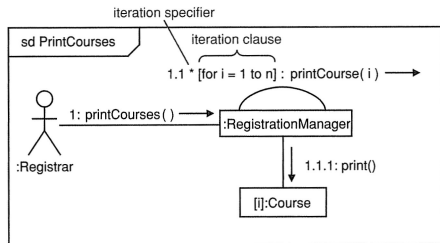
# Communication diagram - Example



# Communication diagram - Iterations

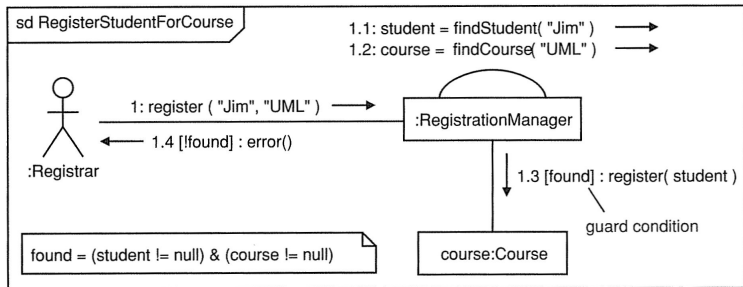
**Iterations** are expressed with *iteration expression*.

Syntax:  $* [loop\ min, max\ [condition]]$



# Communication diagram - Branching

**Branching** is modelled by adding a guard condition to the message. The message is sent only if the guard condition is satisfied.



# Tasks

- Open the use case and design class models and for each use case, think about the interaction between the objects that are participating in the use case (ideally, in terms of a pseudocode).
- Select the 5 most interesting (sufficiently complex) use cases.
- Two of them (the simpler ones) should be modelled by communication diagrams
- The remaining 3 should be modelled by sequence diagrams
- Update the design class diagram if necessary.
- **BONUS:** Complete the sample exam assignment for sequence diagram.
- Submit the **pdf report** to the homework vault (**Seminar 10**).  
**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**

Content Options Page Setup Cover Page Header/Footer Document Info Watermark

**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

Reset Reset to Default Set as Default Generate Cancel Apply Help