Bonus UMLDemo

Martin Macák

6. 12. 2024

FI MUNI, Brno

Introduction

1. Organizational info
2. Test section
   - Tips and tricks
   - Illustrative test
   - Q&A
3. Modeling section
   - Tips and tricks
   - Possible diagrams
   - Q&A
4. Bonus: where to next?

Organizational info

- 90 minutes – testing and modeling section (each worth 35 points)
- To pass, you only need to score enough points to have at least 50 points with the questionnaires
- Don't forget to bring ISIC, pen, and water

Test section

- 7 ABCD questions
- 2 types of questions:
  - **exact fit** – if you have no idea, you can guess
  - **sum of options** – mark only what you are 90% sure of
- go through them ASAP, so you have enough time for the diagram (at least an hour)

Modeling section

- task: Create a diagram
- Sequence, Design Class, State, Activity, Use Case diagram
- 2 problems:
  - Come up with a diagram
  - **fit it onto the paper**
- request a scratch paper or use the other side of the paper for a draft of the diagram
- then redraw the draft into a cleaner diagram on the official page
- printed letters are more readable than handwritten ones
- for longer strings, you can use a note and *, or a substitution

Modeling section – Demo

- Sequence diagram
  - get, set, create, replace, delete, conditions, loops
  - who is calling whose method?
- Design Class diagram
  - entities, enums, helper and manager classes, associations, encapsulation
  - what data do entities carry?
  - how is this data inserted into the entity?
  - how is this data retrieved and updated?
  - what functionality do we expect from the manager classes?

Modeling section – Demo

- State diagram
  - states, transitions
  - can an object be in two states at once?
  - order of event type *suitability* in transitions:
    1. call event
    2. time event
    3. change event
    4. signal receive event
- Activity diagram
  - actions, conditions, parallelism, events
  - what steps are necessary to implement the given use case?
- Use case diagram
  - actors, use cases
  - does it make sense to represent generalization, include, or extend?

Where to next?

1. **PA017** Software Engineering II
2. **PA116** Domain Understanding and Modeling
3. **PA103** Object-oriented Methods for Design of Information Systems
4. **PV167** Project in Object-oriented Design of Information Systems
5. **PV260** Software Quality
6. **PV178** -> **PV179** System Development in C#/.NET
7. **PB162** -> **PV168** -> **PA165** Enterprise Applications in Java

Conclusion

- Contact: **macak@mail.muni.cz** (.thason)