Probe Klausur 2019

Einführung in die Softwaretechnik (IN0006) (Technische Universität München)

Your subm [redacted]

Export PDF

Exercises

Exercise 1

Bumpers Analysis Object Model [6p]

Bumpers Sprint 3

Bumpers is a small single-person game. The player has their own car and is able to steer it across the game board with their mouse. Several other cars are driving around randomly. The player has to destroy all other cars in order to win the game. When the player's car is destroyed by one of the bot cars, the player loses.

Sprint 03 includes the following backlog items:

- Item 7: The game supports different car types.
  - Hint: Fast and slow cars are already supported, come up with at least one more car type. Remember to add new pictures too!
- Item 12: Collisions follow the “right before left” rule, and thus right-most cars win the collisions.
  - Hint: The loser car is crunched and stops driving
  - The player gets notified when he looses or wins the game
- Item 15: Implement a new type of collision as subclass from the class “Collision”, be creative!
  - If you need inspiration, here are a couple of examples:
    - The winner has to hit the car twice
    - The winner is the car from the top
    - Handle cars like a wave

Your Job

Similar to the lecture, update the analysis object model (UML Class Diagram of L02 slide 87) of Bumpers with classes, attributes, and associations for the backlog items of Sprint 3. Also think about if there is common functionality among classes that could be extracted to make the model better readable. * Hint: Review abstraction, encapsulation, and inheritance from lecture 01.

[redacted]

[redacted]

Exercise 2

Inheritance vs. Delegation [4p]

Describe the differences, advantages, and disadvantages between inheritance and delegation in your own words. Choose your own example for each concept and explain why you would choose this concept.

Inheritance means that a class implements methods and attributes from parent class and may be specifies some methods of parent class.
Delegation means that a class calls a method from other class.
Delegation's advantages is delegation is flexible, we can change any object at runtime as long as they have the same type. Delegation 's disadvantage is it's not efficient. it has to invoke many objects.
example: Chatclient using delegation. Each layer calls a function from other class(chatclient calls the under Layer). With delegation we can easy sending request from chatclient to under layer without knowing details about these classes.Otherwise we can easily check if the request is legit
Inheritance's advantage is intuitive to use.it's easy to understand and implement. Many program language supports inheritance.Although (disadvantage) it makes some methods or attribute of parent class exposed. when we change something in parentclass we have to change this class too.
example: Bumper 3 choose fastCar inherits from Car. Because fastCar has all character of Car so by using it we dont have to reimplement many methods and attribute

Exercise 3

Subsystem decomposition of a Q&A system [4p]

Model the subsystem decomposition using a UML component diagram for the following problem statement of the Q&A system:

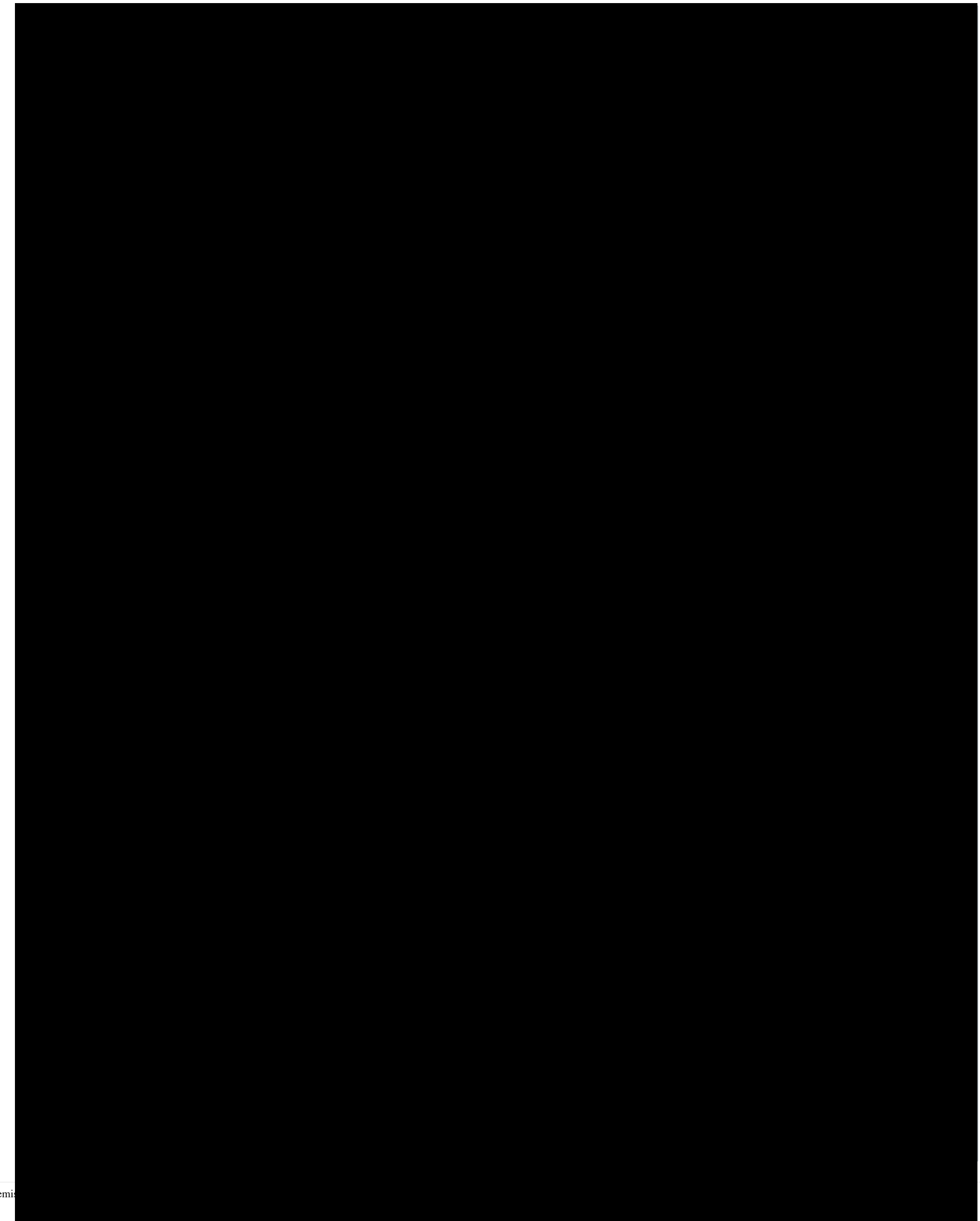
- **The Q&A system stores student questions and theirs answers.**
- **The Q&A system stores users with their access rights (student, tutor, instructor).**
- **Authenticated students can browse and view questions in a graphical user interface (GUI). They can ask new questions, however, they cannot answer questions.**

[redacted]

- Authenticated tutors and instructors view questions on a separate moderator GUI that can filter all unanswered questions posted by students. Tutors and instructors can answer questions. Only instructors can mark an answer as accepted if they determine it as correct. Students cannot access the moderator GUI.

Hints:

- Use meaningful names for components (e.g. Server is `not` a meaningful name).
- The subsystem decomposition should not include hardware nodes and should be independent of the used technologies.
- Apply functional decomposition to identify the most important functionalities of the system and map them to UML components (grouped in a logical way).
  - **Optional challenge:** Create a UML use case diagram (e.g. on **Apollon**) and present this in your tutor group.
- Try to address good system design principles (low coupling, high cohesion).
- Make sure to include the following components:
  - Authentication
  - Moderator GUI
  - Question GUI
  - Question Answer Storage
  - User Repository
- Model the connectors between components using `interfaces` when components provide or require a specific `service` (i.e. subsystem interface). If this is not applicable, you may instead use `dependency` to highlight that one component depends on another one.



Exercise 4

Bridge Pattern [14p]

Repository Link: [redacted]
Last Commit Hash: No commit was made

Exercise 5

Mock Object Pattern 2 [14p]

Repository Link: https://ge35wir@bitbucket.ase.in.tum.de/scm/EIST20TOETESTINGMOCK2/eist20toetestingmock2-ge35wir.git
Last Commit Hash: No commit was made

Exercise 6

Quiz 2 [12p]

1) Techniques, methodologies and tools

Which mappings are correct?

Please choose all correct answer options

Score: 1

- [ ] Bubble sort algorithm --> tool
- [ ] Functional decomposition --> tool
- [x] Object oriented analysis and design --> methodology

2) Requirements elicitation vs analysis

What are differences between requirements elicitation and analysis?

Please choose all correct answer options

Score: 1

- [ ] Analysis defines the system in terms of the user, wheras requirements elicitation defines the system in terms of the developer
- [x] Use cases are an output of requirements elicitation, whereas the dynamic model is an output of analysis
- [ ] Requirements elicitation creates the analysis model, whereas analysis creates the requirements specification
- [ ] Requirements engineering only includes requirements elicitation and not analysis

3) Client server

What are **limitations** of the client server architectural style?

Please choose all correct answer options

Score: 1

- [ ] The client cannot call a service on the server
- [x] The client does not offer an interface to the server
- [ ] The server cannot call a service on the client
- [ ] The server does not offer an interface to the client

4) In object oriented testing...

Please choose all correct answer options

Score: 1

- [x] ... doubles replace the SUT's collaborators during testing
- [ ] ... the SUT initializes mock objects during testing
- [x] ... the test model can contain dummy objects
- [ ] ... SUT stands for system unit test

5) Decomposition

Which of the following statements about decomposition are correct?

Please choose all correct answer options

Score: 1

- [x] When developers only apply functional decomposition in large and complex software projects, it leads to maintainability problems
- [x] In object oriented decomposition, the system is decomposed into classes/objects
- [x] Object oriented decomposition means that functions are decomposed into smaller functions
- [x] Decomposing the system into functions is called functional decomposition

6) Integration Testing

Which of the following answers are correct?

Please choose all correct answer options

Score: 1

- [ ] Horizontal integration testing strategies always test all layers of the system
- [ ] Top down integration testing can result in a lot of doubles if the lowest level of

- [x] the system contains many methods
- [ ] Bottom up integration testing is useful for testing real time systems
- [x] Vertical integration testing allows to always have an executable version of the system

7) Architecture

Which statements describe the relationship between architectural styles and software architectures?

Please choose all correct answer options

Score: 1

- [ ] An architectural style is an instance of a software architecture
- [ ] A software architecture is an instance of an architectural style
- [ ] A subsystem decomposition is a pattern for an architectural style
- [ ] Both terms describe the same

8) From analysis to system design

How do the requirements analysis artifacts **mainly** influence system design?

Please choose all correct answer options

Score: 1

- [ ] The dynamic model influences global resource handling
- [ ] Concurrency is influenced by the dynamic model
- [ ] Design goals are influenced by the dynamic model
- [ ] The functional model influences the boundary conditions

9) Coupling and cohesion

Which statements are correct?

Please choose all correct answer options

Score: 1

- [ ] Low coupling and high cohesion indicate a good system design
- [ ] High coupling and low cohesion indicate a good system design
- [ ] Cohesion measures the dependency among subsystems

Coupling measures the dependency among classes ☐

10) Unit Tests

Which of the following statements about unit tests are correct?

Please choose all correct answer options

Score: 1

- Unit tests are typically used in development environments ☐
- Unit tests can be executed with JUnit ☐
- An assertion verifies that a condition is met when the code is executed ☐
- Unit tests can test whether a method's observed output is the same as the expected output ☐

11) Design goal trade-offs

What are typical design goal trade-offs?

Please choose all correct answer options

Score: 1

- Rapid development vs. functionality ☐
- Usability vs. user-friendliness ☐
- Functionality vs. usability ☐
- Cost vs. robustness ☐

12) System model

What are the components of a system model?

Please choose all correct answer options

Score: 1

- Object model, task model, and dynamic model ☐
- Object model, functional model, and issue model ☐
- Object model, functional model, and dynamic model ☐

A Exercise 7

As-Is Scenario for Bumpers [4p]

Take another look at Bumper's problem statement:

Bumpers Problem Statement

Bumpers is a game where cars drive on a game board and can crash each other. In each collision, there is a winning car. The car that wins all collisions is the winner of the game. The player can start and stop the game. When the game is started, music is played. A car can be either fast or slow. There is one car controlled by the player. The player can steer the direction of the car with the mouse and change its speed. The game should be platform independent. It should visualize different parameters of the car, e.g. the speed, consumption and location of the car. When cars crash, there has to be a sound effect. The game should support different collisions. The determination of the collision winner should be changeable during gameplay.

Your Job

Create your own **formalized as-is scenario** for the Bumpers game. Your as-is scenario should include **at least four events**.

No submission