Robin Weiland

EIST Summary

01

L01E01 Quiz

1.) System model

> *What are the components of a system model?*

- [x] Object model, functional model, and dynamic model
- [ ] Object model, task model, and dynamic model
- [ ] Object model, functional model, and issue model

2.) EIST

> *What does EIST mean in this course?*

- [x] Einführung in die Softwaretechnik
- [ ] Enhanced Intel Speedstep Technology
- [ ] Earth Information Science and Technology
- [ ] European Information Society Technologies

3.) Software development

> *Which of the following is part of software development?*

- [x] Problem Solving
- [x] Dealing with complexity
- [x] Dealing with change
- [x] Writing Code

4.) Techniques, methodologies and tools

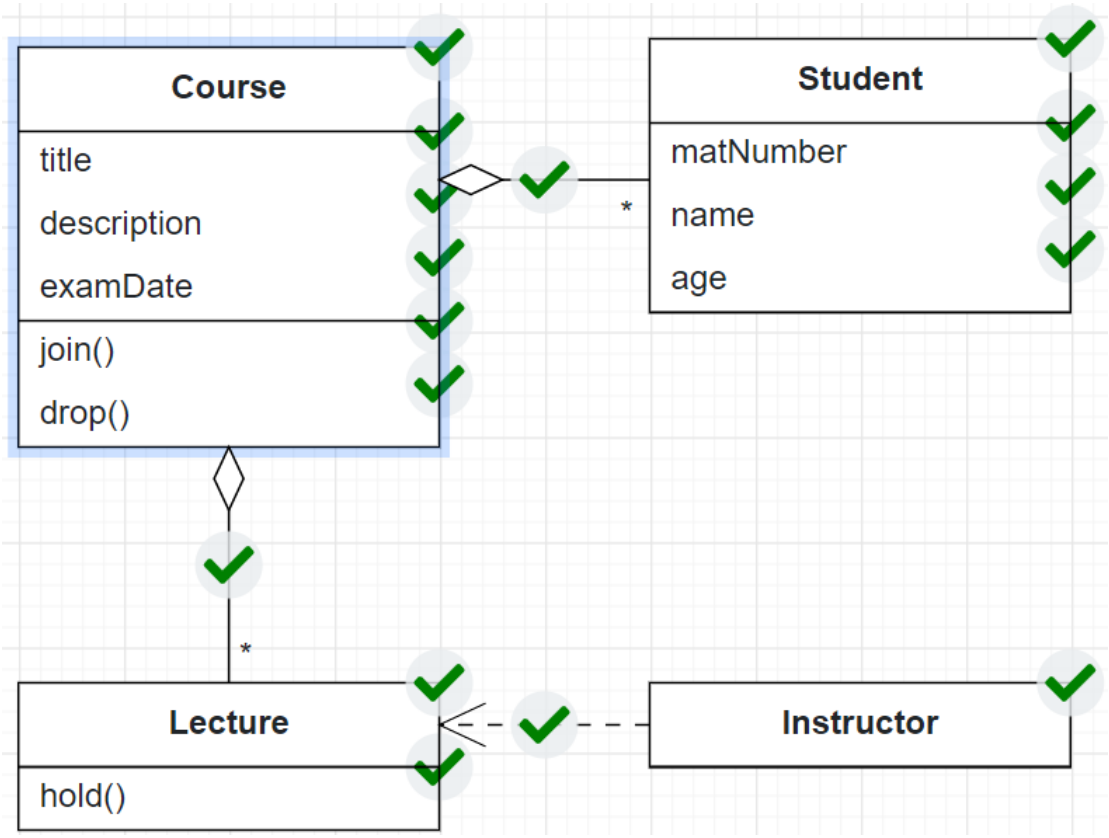
> *Which mappings are correct?*

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

L01E02 Modeling Tutorial

> *A university course has a title, a description and an exam date. Students have a matriculation number, a name and an age, they can join and drop a course. Instructors hold the lectures of a course.*

> ***Your task*** *Create an analysis object model (UML class diagram) to model this problem.*



02

L02E01 Quiz

1. Polymorphism

> *What is correct about* ***polymorphism****?*

- ◯ Java decides during compile time which method will be invoked
- ☑ Java decides during runtime which method will be invoked
- ◯ Two sub classes **cannot** implement the same method
- ◯ There is **no** difference between the compile time type and the runtime type in Java

2. Abstract classes

> *Which statements about abstract classes are correct?*

- ☑ Abstract classes cannot be instantiated
- ◯ Abstract classes are the same as interfaces
- ◯ Abstract classes can **not** implement methods
- ☑ Abstract classes can force subclasses to implement a specified method

G01E02

> *What is a* ***model*** *in the context of software engineering?*
> *Name one example of a model.*

A model is a abstracted representation of a much more complex system that might even no longer exist or isn't build yet.

In context of software development a model helps communicating ideas in a way that they are understood by the whole team or potentially even customers without requiring any extensive technical knowledge or skill.
A model gets created at the start of a project but might maintained and updated during the project while it serves as a reference.

In the municipal services center in my hometown Coburg they have a large table with a rough map of the town with small LEDs and markers on it that shows certain nodes and their status.
This model helps to get a clearer view over the complex system of electricity, water and gas distribution.

03

L03E01 Quiz

1. Use case diagram

   > *What are differences and similarities between **«includes»** and **«extends»**?*

   - ◯ Both model relationships between multiple use cases
   - ✔ Both model rarely used functionality
   - ◯ «extends» is used to model exceptions while «includes» allows to reuse functionality
   - ◯ There is **no** difference between the compile time type and the runtime type in Java

2. Software lifecycle

   > *What is correct about software lifecycles?*

   - ✔ Software lifecycles consist of development and management activities and their relationships
   - ◯ All software lifecycle models must include the activity testing
   - ◯ Defined process control regularly inspects and adapts
   - ✔ Empirical process control reacts to changes and tries to use the fast reaction as an advantage over the competition

L03E02 Quiz

1. Requirements elicitation vs analysis

   > *What are differences between requirements elicitation and analysis?*

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

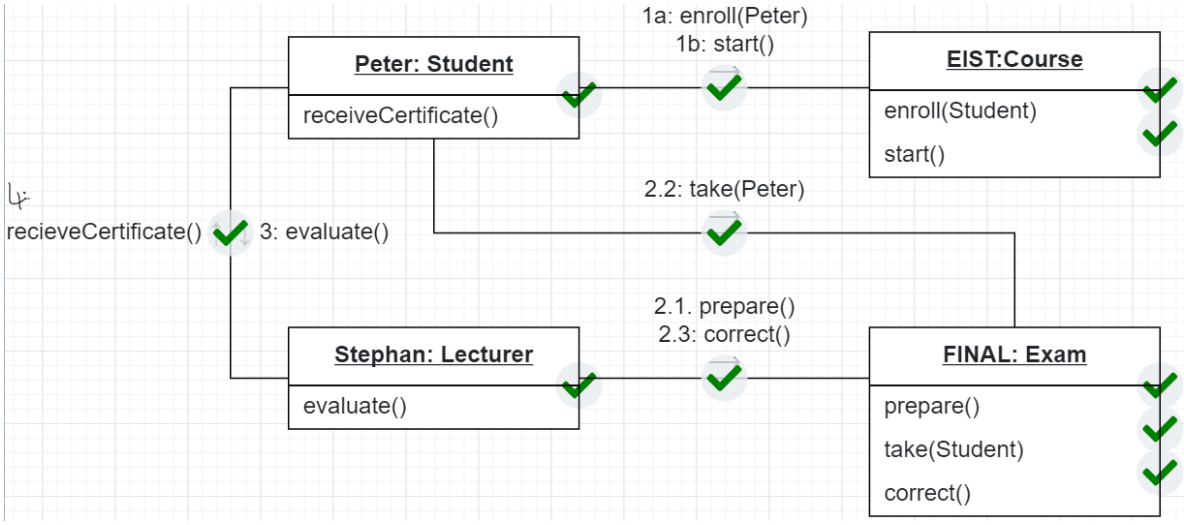
2. Pseudo requirements

> *Which of the following requirements of Bumpers are pseudo requirements?*

- ◯ The Car class must be extensible
- ◉ Bumpers must be developed in Java
- ◯ When pressing the start button, the game must start within 1s
- ◯ Users should be able to steer the car with the mouse

L03E03 Model a Communication Diagram

> *Model the dynamic behavior of the following scenario using a UML communication diagram.*
>
> 1. ***Name:*** *Pass EIST exam*
> 2. ***Participating actors:*** *Peter: Student, Stephan: Lecturer*
> 3. *Flow of events:*
>   1. *Peter enrolls in the EIST course and starts the course*
>   2. *Stephan prepares the final exam*
>   3. *Peter takes the final exam*
>   4. *Stephan corrects the final exam*
>   5. *If Peter has passed the exam, he receives a certificate*
>   6. *Peter evaluates Stephan at the end*



H01E01 Different Models in SE

> *What is the difference between a functional model, an object model, and a dynamic model?*

A model is an abstraction of a system that even might no longer exist or yet has to be built.

The **functional model** describes the functions of a system, primarily from an end-user interaction perspective. It helps to clearly determine what a user can and potentially cannot by declaring use-cases.
This model can be referenced to check the requirements with the customer, particularly those concerning users.

The **object model** defines the structure of a system and might outline potential subsystems, which can be tackled separately by the developers.
It can also clarify dependencies within the system, which are important for the finished product as well as during development when certain parts need to be finished before others. This is also quit useful for the management side of things since they need to plan around delays and bottlenecks during development.

The **dynamic model** illustrates the systems behavior regarding external events. It is highly dependent on both the functional and the object model since it describes the intrinsic connection between what happens outside and how the system reacts. While the object model provides a rather broad scaffolding for the implementation, the dynamic and to some extent the functional model give a clearer image on how some are specific aspects should work.

$\longrightarrow$ The **system model** is the congregation of the models above.

H01E02 Change in Software

> *Why does change happen in software development? Provide examples from the real world and discuss them in the 3 areas of requirements changes, technology changes, and organization change.*

Changes in software development depend on several factors:

```
A racing team might order a certain type of tire pressure sensor.
The requirements are already negotiated with the deliverer,
the problem is analyzed and an initial draft was developed.
Just as implementation started the racing association decides
the type of tire will no longer be allowed in races this season,
which overthrows almost all work done so far.
```

This example shows how adaptive software development can be an iterative process between a requirements/planning phase and actual implementation either caused by internal changes like unforeseen bugs and problems or external variations for example changes in rules or expectations.
Such challenges also require a good communication and mutual understanding on both sides, which shows that social skills are highly important for developers as well in changing development.

```
A game console manufacturer is planning to release a new
generation of consoles, which differs hugely from the previous
system, even in a way that it is not backwards compatible
(e .g. ps3(power pc with CELL-processor) --> ps4(x86-64)).
Although probably notified about such changes prior to any
public announcement, a game developer has to port their
highly anticipated game to this new console generation as well,
since their contract with the manufacturer who hopes to
increase sales with the game as an incentive for the new console.
```

In the situation above the management of said studio, would have to split and reorganize the team into core game, old and new generation.
From a technical perspective large parts of the underlaying engine or any other related software system.
Developers need to be able to learn how to use new systems, software or tools quite quickly to be able to tackle challenges of that nature.

```
Amy Henning, creative director of the early entries of
Naughty Dog' s Uncharted series, proposes a
movie-production-like approach for the games industry, where
developers, rather than working for a studio, meet in project
where everyone is a freelancer so to speak and publishers
can choose every role in the team individually instead of
contracting a whole studio.
```

Such a drastic organizational change would involve transitions on one hand at the management level, concerning organization of the composition of these teams as described above, as well as on an individual level for every "participant".
For instance, in a more or less consistent group within a game studio (to stick with the example above) all members know each other, at least to an extend whether they an estimate the broad skills and workflow of their peers, which is of great importance when working in teams together. Working in such an temporary team, developers need to adapt to the fact they don't know each other so well.

04

L04E01 Quiz

1.) UML

> *Given the following UML class diagram, which statements are correct?*

deemed invalid

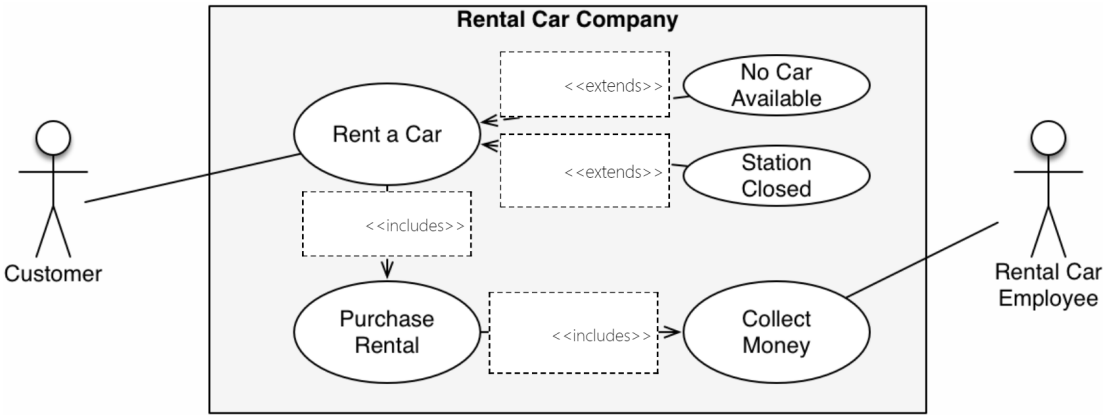
2.) Actor, Class & Object

> *Which statements are correct?*

- ◯ Actors are always the users of the system [deemed invalid]
- ◉ An object is a specific instance of a class
- ◯ Every class is an object
- ◯ Every object is a class

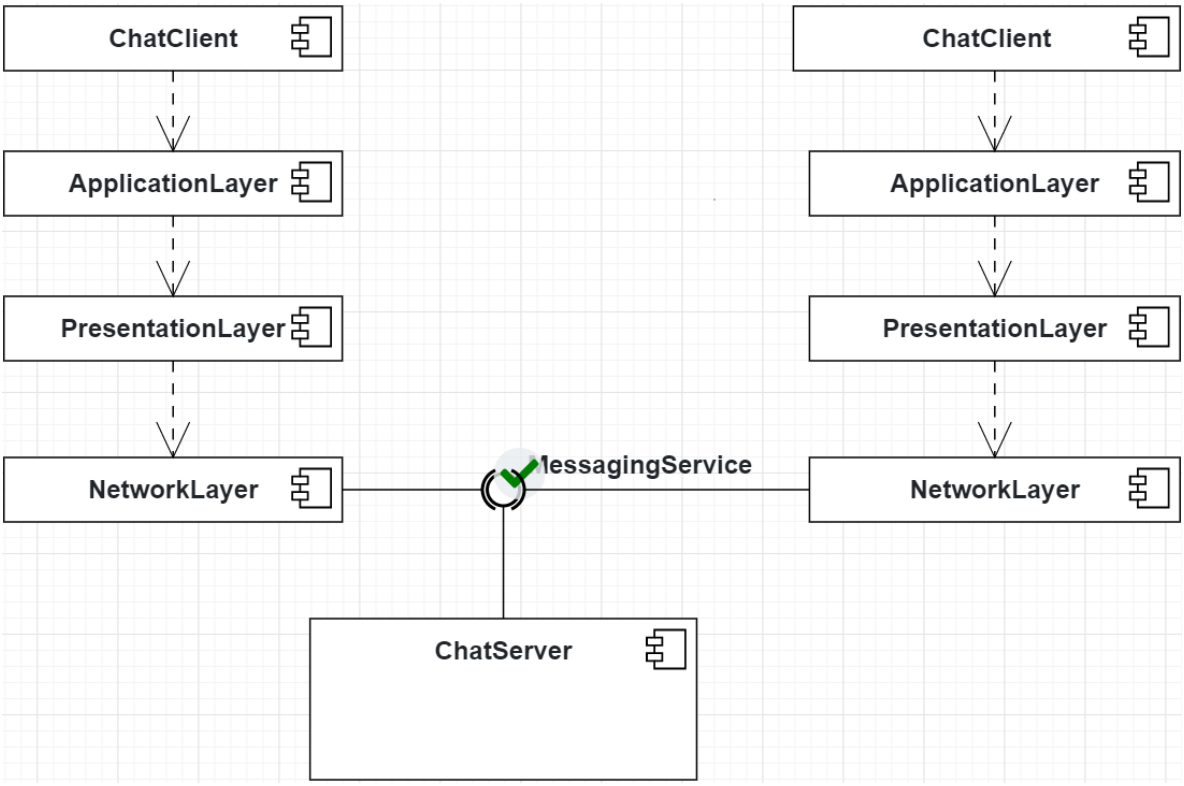
3.) Use Case Diagrams

> *Drag and drop the correct elements into the UML use case diagram.*



L04E03 Model a chat system

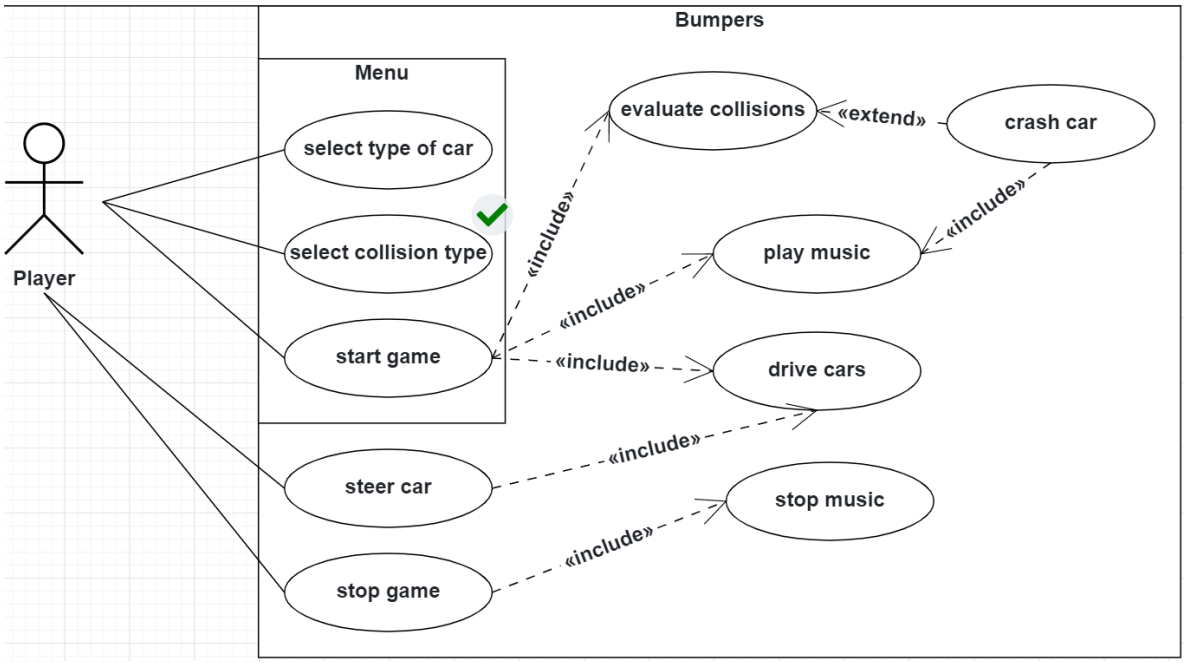
> ***Model the architecture of a chat system with a layered architectural style***
>
> - *2 computers are using a ChatClient with an application layer and a network layer*
> - *The network layer communicates with a ChatServer*
>
> ***Change: the messages should be end-to-end encrypted***
>
> - *Encryption should be implemented in a new presentation layer between the application layer and the network layer*



H02E01 Bumpers Functional Model

> ***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 functional model (UML Use Case Diagram of Lo2 slide 86) of Bumpers with the following requirements: In the future, when starting the game, the player can choose …*
>
> 1. *… the type of their car*
> 2. *… the collision type*



H02E01 Bumpers Functional Model

05

May 17

06

May 24

L05E01 Quiz

1.) Coupling and Cohesion

> *Which statements are correct?*

- ◯ Cohesion measures the dependency among subsystems
- ◯ Coupling measures the dependency among classes
- ✔ Low coupling and high cohesion indicate a good system design
- ◯ High coupling and low cohesion indicate a good system design

2.) Design goal trade-offs

> *What are typical design goal trade-offs?*

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

3.) From analysis to system design

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

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

4.) Architecture

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

- ◯ 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

5.) Decomposition

> *Which of the following statements about decomposition are correct?*

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

6.) Client server

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

- ◯ The client **cannot** call a service on the server
- ✔ The server **cannot** call a service on the client
- ✔ The client **does not** offer an interface to the server
- ◯ The server **does not** offer an interface to the client

07

May 31

L06E01 Quiz

1.) Access Control

> *Access on classes can be modeled by means of an access matrix. In this matrix …*

- [x] The rows represent the actors of the system
- [ ] The columns represent the actors of the system
- [x] An access right lists the operations that can be executed on instances of the class by the actor
- [ ] Performance is never an issue, no matter how big the table is

2.) Fork and Stair Diagrams

> *Which of the following statements about fork and stair diagrams are correct?*

- [x] Fork diagram: the control object uses other objects for direct questions and commands
- [x] Fork diagram: the control object knows all other objects
- [x] Stair diagram: each object can delegate responsibilities to other objects
- [ ] Stair diagram: the control object knows all other objects

08

June 7

09

June 14

L07E01 Quiz

1.) Discovering Inheritance

> *Which of the following statements about inheritance is correct?*

- [x] Generalization means that we first determine the subclass and then the superclass.
- [x] Specialization means that we find a subclass for an existing more general class.
- [ ] Specialization means that we discover the subclass first and then the superclass.
- [ ] Generalization is the same as delegation.

2.) Reuse

> *You have implemented a class `Car` with two methods `drive()` and `crash()`. Later in the development, you realize that you want to support multiple cars `FastCar` and `SlowCar` which have a common behavior as `Car`, but are slightly specialized. Which concept of reuse should you apply?*

- [ ] Specification inheritance
- [ ] Delegation
- [x] Implementation inheritance

10

June 21

L08E01 Quiz

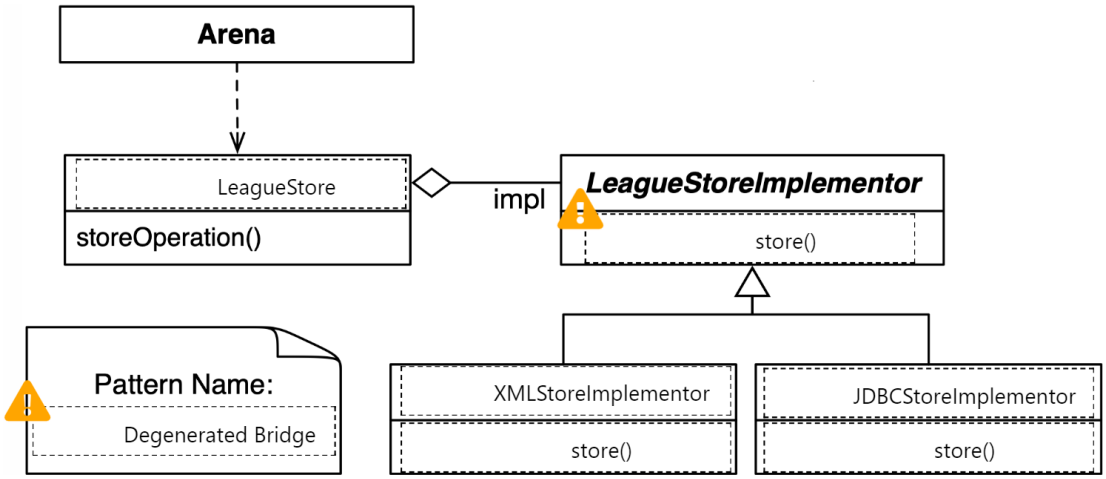
1.) Clues for Design Patterns

> *The following terms hint at using design patterns. Which of the patterns are correctly identified?*

- ✔ "interface to an existing API" >> Façade
- ✔ "complex", "variable length and height" >> Composite
- ✔ "set at startup time" >> Bridge

2.) UML Model of a Pattern

> *Drag and drop the following components into the diagram and name the following pattern*



11

June 28

L09E01 Quiz

1.) In object oriented testing...

   > *Please choose all correct answer options*

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

2.) Unit Tests

   > *Which of the following statements about unit tests are correct?*

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

3.) Integration Testing

   > *Which of the following answers are correct?*

   - [ ] Bottom up integration testing is useful for testing real time systems
   - [x] Horizontal integration testing strategies always test all layers of the system
   - [x] Top down integration testing can result in a lot of doubles if the lowest level of the system contains many methods
   - [x] Vertical integration testing allows to always have an executable version of the system

4.) Testing with JUnit 4

   > *Which of the following statements about testing with JUnit 4 are correct?*

   - [x] `assertEquals(expected, actual)` asserts that the two passed elements are equal
   - [x] `assertNull(object)` asserts that the object is null

- [x] `fail(String)` lets the test fail and outputs the string
- [ ] `fail(String)` checks if the test fails and outputs the string

12

July5

L1oE01 Quiz

1.) Process Control

> *What are differences between defined and empirical process control?*

- [x] In defined process control, everything is completely understood, which is not the case in empirical process control
- [ ] Empirical process control sees deviations as errors, while defined process control sees them as opportunities
- [x] Empirical process control deals better with changes than defined process control
- [x] Given a well defined set of inputs, defined processes create the same output every time, while the application of empirical process control can lead to different outputs

2.) Scrum Roles

> *Which statements about Scrum roles are correct?*

- [x] The Scrum Team includes members from different disciplines
- [ ] The Scrum Master decides everything
- [x] The Product Owner should have application domain knowledge
- [ ] The Product Owner should make sure that the team follows the Scrum process
- [ ] The Product Owner is always the end user

13

July12

L11E01 Quiz

1.) Distributed Version Control Systems

> *What are advantages of distributed version control systems?*

- [x] The repository can be restored from a programmer's computer if the server crashes
- [ ] It has a low learning curve
- [x] Branches are lightweight
- [ ] Programmers could exchange commits directly without the need to store the version on the remote repository

2.) git

> *What is correct about Git commands?*

- [x] `git clone` creates a local working copy from an existing git repository
- [ ] `git pull` promotes your changes to other developers
- [x] `git add` marks file changes in your existing directory to be committed
- [x] `git pull` is a compound command, composed of `git fetch` and `git merge`

3.) Change Management

> *Which of the following statements are correct?*

- [x] A change policy can, e.g., guarantee that each promotion conforms to commonly accepted criteria
- [ ] In Scrum, change requests only occur in the analysis phase
- [x] In certain cases, a change request from the customer can also be rejected
- [ ] The Waterfall Model encourages change requests from the customer in every phase

14

July 19

15

Jule 26