Software Systems Design - Diagnostic - 2022-12-01

Cursus: B-CS-MOD01-1A-202001022 B-CS Pearls of Computer Science Core 202001022

Gegenereerd op: 1 dec. 2022

Inhoud: Pagina's:

Software Systems Design - Diagnostic - 2022-12-01

Cursus: B-CS-MOD01-1A-202001022 B-CS Pearls of Computer Science Core 202001022

Welcome to the Software Systems Design digital diagnostic test:

- ***This is an open book exam -- You are allowed to use the slides which are digitally available on your Chromebooks under the URL: https://grammarware.net/slides/2022/ss/ (a separate tab should already be open).***
- ***Smartphones or personal notebooks are not allowed. Put those in your bag now (with the sound switched off).***
- ***For technical questions concerning the chromebooks, Remindo, log-in issues etc.: raise your mouse***
- ***For content questions: use the BBB chat.***
- ***Do not forget to save your answers once you are done.***
- ***Take your time to familiarize yourself with the Remindo environment.***
- ***The real exam is longer and will have more questions.***
- ***This is a diagnostic test. It is not graded! (Even though Remindo says so).***

Good luck and enjoy the test!

1 4 pt.

The classical phases of software development consist of 1. requirements, 2. design, 3. construction, 4. testing, 5. deployment and 6. maintenance phase.

Describe the key differences between the construction phase and the maintenance phase.

2

The core components of object-oriented design, according to how it was discussed during the course, are

1. hierarchy,
2. abstraction,
3. modularisation, and
4. encapsulation.

Explain in your own words what each is, and why it is important.

1 pt. **a.** Hierarchy:

1 pt. **b.** Abstraction:

1 pt. **c.** Modularisation:

1 pt. **d.** Encapsulation:

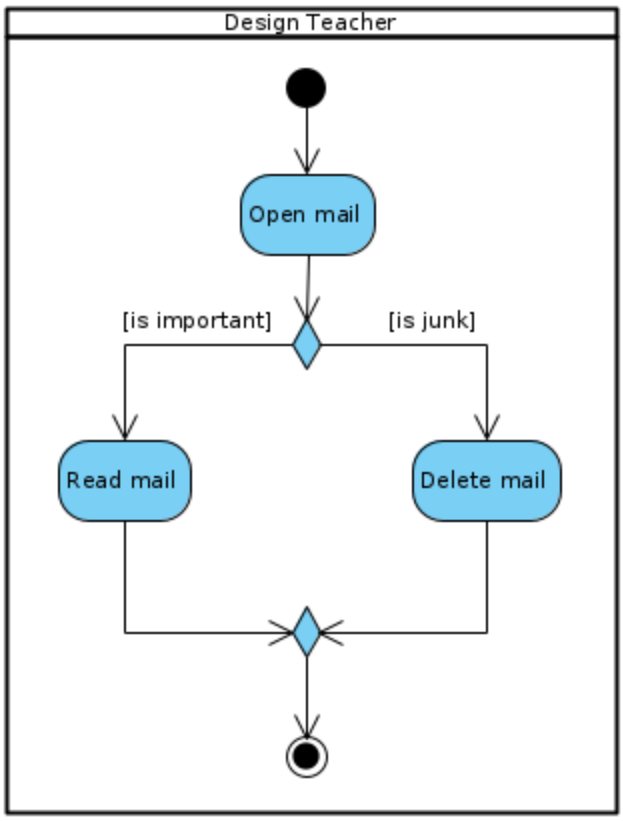
3

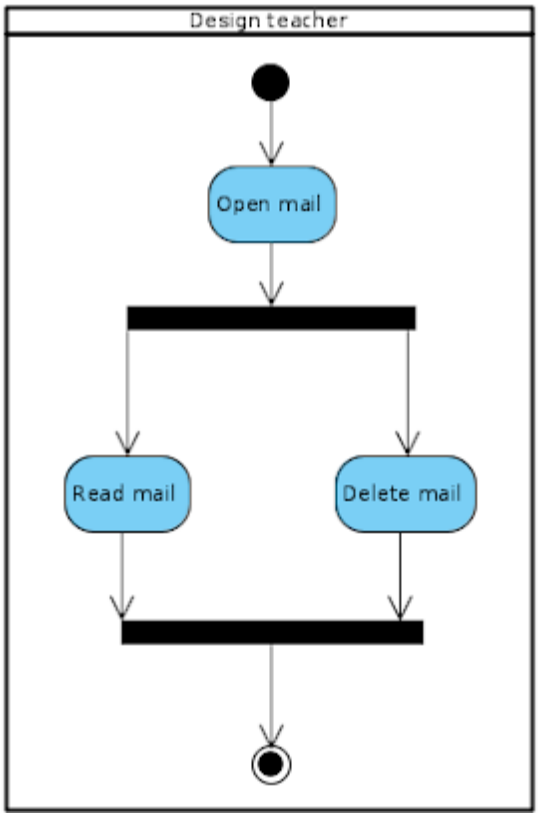
"During requirements elicitation it is your job to find out what the customer *really* wants" is a commonly repeated statement in software development.

4 pt. **a.** Explain what is meant by it, and describe how requirements elicitation can help to circumvent potential problems.

2 pt. **b.** Name two elicitation techniques and explain their relevance in the phase.

4 Given are the following two activity diagrams AD1 (left) and AD2 (right):





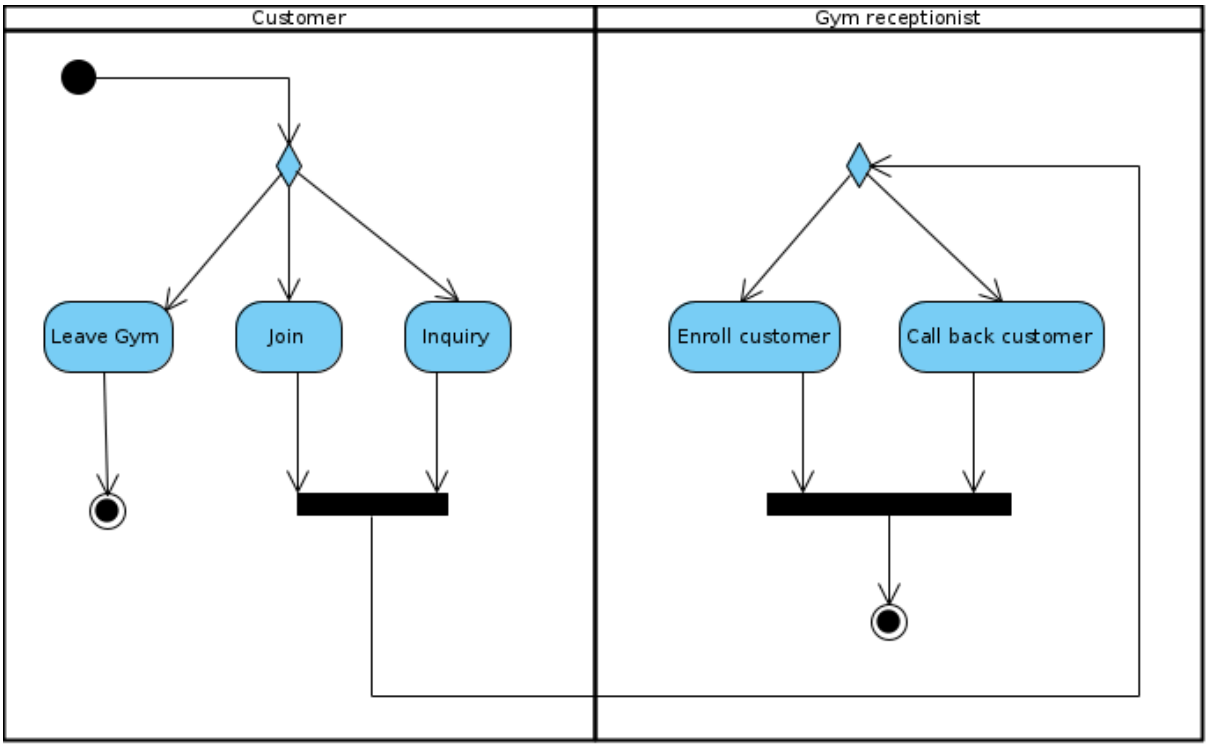
4 pt. **a.** Explain the difference between the two diagrams. Do both of them model the same behaviour?

2 pt. **b.** Which of the two activity diagrams models the behaviour of your Design teacher best.

Provide a short explanation of why you think that way.

5 Given is the following activity diagram modeling the enrollment of a customer in a gym:

6 pt.



Explain what is wrong with the presented diagram by pointing out **three issues** that can be improved.

Shortly describe how you would change them.

6 8 pt.

Given below are pairs of classes (eg. "House" and "Room").

For each pair:

1. **draw** a sensible association between the classes (use the drawing tools beneath the image),
2. specify meaningful association names when appropriate (eg. "lives in" or "owns"),
3. specify the multiplicity/cardinality of the association if appropriate (eg. "1 - 1" or "1..* - 0.. 5").

Do not draw associations that cross the black lines.

Hint: Your drawing must not be perfect as long as its intention is clear.

7 2 pt. You want to develop a software system for source code management as a direct competitor to *git*.

Name exactly four requirements you would use to ensure your success.

Thank you, your answers were saved.

Note that we will not correct your answers to this diagnostic test. You can find its on Canvas.

Correctiemodel

1.
4 pt.

Correction criterion	Points
Construction: The actual implementation of the software and writing of the necessary code. Maintenance: The application is done and is applied in the field. In this phase, users may encounter bugs that were not encountered during the testing phase. These bugs need to be resolved, which may lead to newer development cycles.	4 points
Total points:	*4 points*

2.
4 pt.

a.

Correction criterion	Points
Highlights common behaviour and makes relationship between objects explicit.	1 point
Total points:	*1 point*

b.

Correction criterion	Points
Reducing the amount of information in a particular model to focus on relevant parts.	1 point
Total points:	*1 point*

c.

Correction criterion	Points
Assumptions that modules make about each other.	1 point
Total points:	*1 point*

d.

Correction criterion	Points
Hide details of implementations that are not relevant at a certain level	1 point
Total points:	*1 point*

3.
6 pt.

a.

Correction criterion	**Points**
- Stakeholders express requirements in their own terms -> "Translate" from stakeholder to software designer - Different stakeholders may have conflicting requirements -> Lay open discrepancies and allow discourse - Organisational and political factors may influence system requirements -> Make stakeholders aware of restrictions - Requirements may have changed -> Get most recent take/new stakeholders etc.	4 points
Total points:	*4 points*

b.

Correction criterion	**Points**
Brainstorming, Interviews, Prototyping, Surveys etc. + a sensible reason	2 points
Total points:	*2 points*

4.
6 pt.

a.

Correction criterion	Points
No, the two diagrams model different behaviour. AD1, the activity diagram on the left, has a 'decision' and a 'merge' node. This serves to model a logical OR decision. That is, either the left branch is taken, or the right branch is taken -- Not both simultaneously. In the example, either the email is read, or it is deleted. AD2, the activity diagram on the right, has a 'fork' and a 'join' node. In contrary to AD1, this models that both actions need to be taken. This is comparable to a logical AND. In the example, the email will be read AND deleted. The 'join' node requires both preceeding actions to be concluded, before the final node is reached. For more information about this question, we refer to video topic 'L4T1 - Activity diagrams'.	4 points
Total points:	*4 points*

b.

Correction criterion	Points
AD1 is (ideally) the correct diagram to model the behaviour of your Design teacher. The dilligent teacher will find every student email to be of equal importance and read it carefully. This is best represented by the 'decision' node of AD1. AD2, on the other hand, makes the teacher delete every email simultaneously while reading it. We refer to video topic 'L4T1 - Activity diagrams' for this very example.	2 points
Total points:	*2 points*

5.
6 pt.

Correction criterion	Points
There are many things that can be pointed out here, we list some of them: * The 'decision' node is followed by a 'join' node. The 'join' node requires both preceeding actions to be fulfilled, while the 'decision' node explicity only ever takes one. The first thing that should be improved in the diagram is its syntactic correctness, i.e. 'decision' and 'merge' nodes vs 'fork' and 'join' nodes. * Same for the gym receptionist swim lane * No guards on transitions after 'decision' node. Actions can be taken without any pre-requisite. Can be fixed by adding guards, e.g. [Too expensive] on transition leading to 'leave gym' [good price] to transition leading to 'join' action * Vague action names, e.g. what is 'inquiry'. Looks like use cases were transferred verbatim to an activity diagram. Can be improved by having more meaningful action names * Aesthetic, but important nonetheless: Swimlanes are presented in parallel via very long arrow after 'join' node. gym receptionist 'decision' node should be moved further down. * The customer never hears back from the gym receptionist. Could be improved by modeling better their interaction.	6 points
Total points:	*6 points*

6.
8 pt.

Correction criterion	Points
2 points per association: a) Laptop "is a" Device Generalisation - multiplicity does not apply b) House "has a" landlord - Association or Aggregation - e.g., 1..* to 1 (more possible) c) Java program "has a" class - Aggregation or composition - 0..* to 1..* d) Woman "is a" human - Generalisation - multiplicity does not apply	8 points
Total points:	*8 points*

7.
2 pt.

Correction criterion	Points
This is a sanity check whether students know what git, VCS and requirements are. Literally anything sensible goes here. Wrong answers will be too abstract, unclear or simply be bare terms without meaning.	2 points
Total points:	*2 points*

Cesuur

Behaalde punten	Cijfer
36	10
35	9,7
34	9,4
33	9,2
32	8,9
31	8,6
30	8,3
29	8,1
28	7,8
27	7,5
26	7,2
25	6,9
24	6,7
23	6,4
22	6,1
21	5,8
20	5,6
19	5,3
18	5,1
17	4,9
16	4,6
15	4,4
14	4,2
13	4,0
12	3,7
11	3,5
10	3,3
9	3,0
8	2,8
7	2,6
6	2,4
5	2,1
4	1,9
3	1,7
2	1,5
1	1,2
0	1,0

Vraag-identificatiecodes

Deze identifiers kunnen worden gebruikt om de precieze vraag in de vragenbanken te identificeren. Gebruik deze code in combinatie met de documentcode wanneer u feedback doorgeeft, zodat precies duidelijk is op welke vraag en -versie uw feedback van toepassing is.

Documentidentificatiecode: 11568-15172

Vraagnummer	Vraag-identificatiecode	Versie-identificatiecode
1	41800	ef08d403-2029-b17e-6ad5-dec1f592a3fe
2	47655	dca9671d-e6e1-a21b-9141-94326355b8c3
3	47385	7d1fa1b2-7daa-48d6-19ce-5048cd11d3e1
4	41600	d087b4aa-28c2-17f0-17c2-ea02a59b9dc2
5	41605	6bd40f1f-c042-893f-b6de-7f11c364bec6
6	115407	d9b6c6c4-46c8-69c9-ec9e-8da9263100a3
7	43260	9d1c0216-8ce4-cbe1-46a3-5e23db1c3fa3