

Software Modeling 2019/20

doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Exam – additional date – December 16, 2019

Last name:		1b	
First name:		2b	
		3b	

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Write the answers to questions 1–12 into the table if the question itself does not prescribe a different way of answering. In multiple-choice questions (a–e) only one choice is correct. The procedure is not considered. Each correct answer has the value as indicated in the question.

Write the answer to question 13 on a separate paper containing this question.

Mark clearly possible corrections. An answer that is incorrect, ambiguous, or incomplete will be graded with 0 points. Only an undamaged paper will be accepted.

1. (2 b) On the level of the flows within one use case, an alternative use case flow is an analogy of:

- (a) the extend relationship
- (b) the inheritance relationship
- (c) the use case participation relationship
- (d) the include relationship
- (e) the exclude relationship

2. (1 b) How developers are organized:

- (a) affects only the organization of the model of the software system produced by these developers
- (b) does not affect the organization of the software system produced by these developers
- (c) affects only the organization of the code of the software system produced by these developers
- (d) affects only the organization of the documentation of the software system produced by these developers
- (e) affects the organization of the software system produced by these developers

3. (3 b) An algebraic specification of a computer game which, at the moment, enables to register only one player using the function:

$$addPlayer : Game \times Player \rightarrow Game$$

has to be extended by a possibility of adding another player anytime during the game. At most two players can participate in a game. Out of the activities introduced below, choose all those necessary

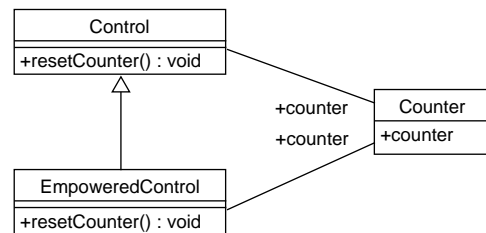
to be performed (answer by introducing the numeral designations of the corresponding possibilities):

1. add the function for reading and writing a variable which will hold the number of players
2. add the function for finding the number of registered players
3. add a variable which will hold the number of players
4. modify the signature of the player registration function
5. modify the preconditions of the player registration function

4. (1 b) Using a feature model:

- (a) it is possible to configure other code, but not other models
- (b) it is not possible to configure other models nor code
- (c) it is possible to configure other models, but not code
- (d) it is possible to configure other models and code
- (e) it is possible to configure only another feature model

5. (3 b) The following UML diagram is given:



The following OCL expression are provided with it:

```

context Control::resetCounter() : void
post reset:
    counter.counter = counter.counter@pre
    || counter.counter = 0
    
```

```

context EmpoweredControl::resetCounter() : void
post reset:
    counter.counter = 0
    
```

No other OCL expressions relevant to this question are defined. Answer the following partial questions by selecting one of the provided answers (in the form: *partial question number: answer*):

1. By this, the postconditions of the `resetCounter()` operation are being weakened, strengthened, or remain the same? [possible answers: weaken / strengthen / remain the same]
2. Is deriving the `EmpoweredControl` class from the `Control` class correct? [possible answers: yes / no]
3. By this, the `resetCounter()` operation preconditions are being weakened, strengthened, or remain the same? [possible answers: weaken / strengthen / remain the same]

6. (2 b) In composite structure diagrams in UML:

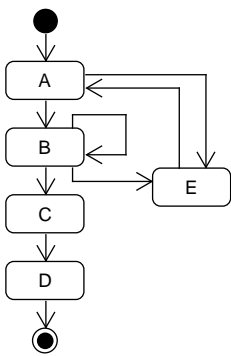
- (a) a part of component A can be connected to a part of component B by an interface, but without using ports
- (b) a part of component A can be connected to a part of component B only directly
- (c) a part of component A can be connected to a part of component B through the port of component A connected to the port of component B by an interface
- (d) a part of component A can be connected to a part of component B through the port of component A connected to the port of component B, but not by an interface
- (e) a part of component A can by no means be connected to a part of component B

7. (3 b) A learning support system enables to work with learning activities. A learning activity is first being created when any modifications. After creating it, a learning activity can be announced and the participants can be gradually registered. After the participant registration is closed, the learning activity realization starts, which lasts until its termination is ordered. Before its realization has begun, a learning activity can be deleted. A deleted learning activity can be renewed, which makes it modifiable again.

Assign the states from the following list:

1. Deleted
2. Provided
3. With the Participants Being Registered
4. Renewed
5. With a Closed Registration
6. Terminated
7. With an Open Registration
8. Announced
9. Being Realized
10. Being Created

to the states in the diagram sketched below:

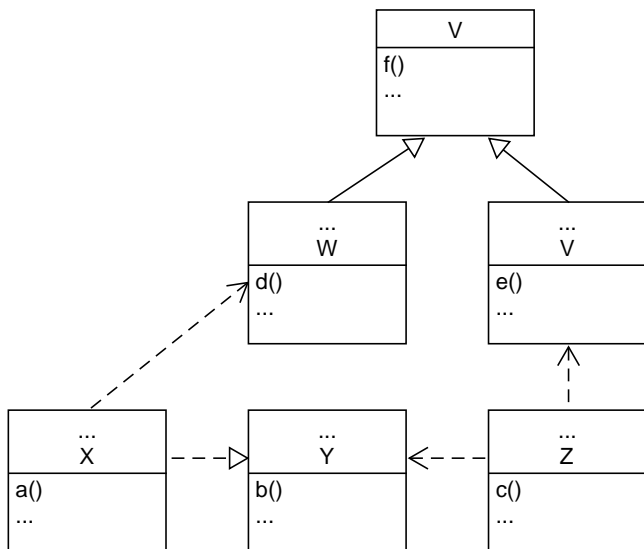


so that a learning activity life cycle in the learning support system would be properly comprised (designating the transitions is not the subject of this question). Provide your answer as an array of the following form:

Aa Bb Cc Dd Ee

where lower-case letters represent state numbers.

8. (3 b) The following UML diagram is given:



Add directly into the diagram the missing text inevitable for the diagram to be correct. An ellipsis can stand for several lines of text or no text at all. In generalization/specialization, express the operations received from the supertype.

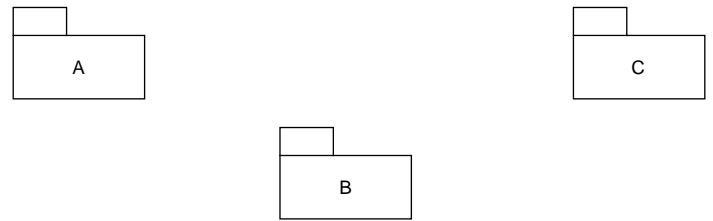
9. (1 b) The basic feature modeling notation (FODA or Czarnecki–Eisenecker, too) enables to express alternative features graphically:

- (a) only if these features have a common superfeature
- (b) only if these features don't have a common superfeature
- (c) only if these features don't have similar names
- (d) regardless of their position in a feature diagram
- (e) only if these features are in the superfeature–subfeature relationship

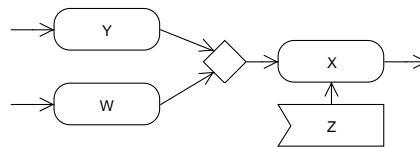
10. (2 b) The inner logic of a use case is implemented:

- (a) always by a single class that implements that use case exclusively
- (b) always by several classes that implement that use case exclusively
- (c) always by a single class that can also implement other use cases
- (d) always by several classes that can also implement other use cases
- (e) by a single class or several classes that can also implement other use cases

11. (2 b) A class from package A realizes an interface from package C. An operation of a class from package A is called by an operation of a class from package B. An operation of a class from package B changes the value of an attribute of a class from package C. Mark the corresponding relationships between the packages in a given UML package diagram:



12. (2 b) The following UML diagram is given:



For action X to be called:

- (a) actions Y and W must be activated and the corresponding event has to be captured by action Z
- (b) action Y or W must be activated and the corresponding event has to be captured by action Z
- (c) the corresponding event doesn't have to be captured by action Z, but action Y or W must be activated
- (d) the corresponding event doesn't have to be captured by action Z, but actions Y and W must be activated
- (e) actions Y and W don't have to be activated, but the corresponding event has to be captured by action Z

Last name:	
First name:	

13. (5 b) An internet banking system enables the clients to make periodical payments. On the webpage that is being displayed, a client enters a sum and target account. The client picks their own account from a drop-down list to make the payment from. Also, they provide the realization date or choose the date by clicking the desired date in the calendar being displayed.

Due to security, the internet banking system requires to confirm all the payments over EUR 100 by an authorization code sent as an SMS to the client's mobile phone number. The authorization code is valid for 180 seconds. The system displays a field into which the client can enter the authorization code. The client confirms entering the code by clicking the corresponding button. If the client enters a wrong code, they can request a new one by clicking the corresponding button.

Identify the use cases that imply directly from the description that has been provided and write their flows relevant to the description. The steps implied directly from the description that has been provided will be sufficient. Depict the corresponding use case diagram.

30 b

1 a

2 e

3 5 a 2

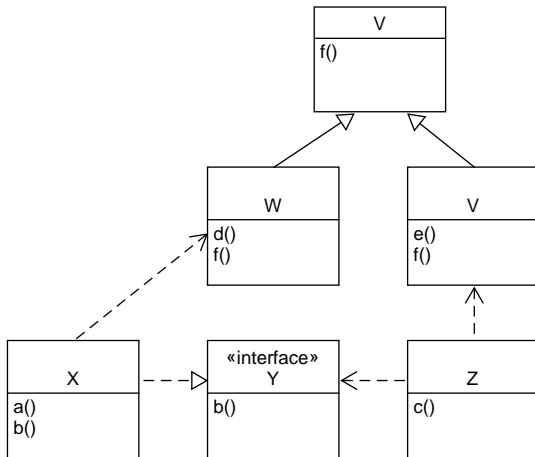
4 d

5 1: strengthen, 2: yes, 3: remain the same

6 c

7 A10 B8 C9 D6 E1

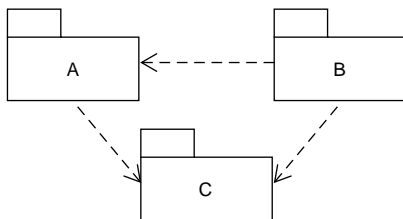
8



9 a

10 e

11



12 b

13 From the description, two use cases imply. They could be named *Enter a Periodical Payment* and *Authorize Payments*, where the *Authorize Payments* use case extends the *Enter a Periodical Payment* use case. The *Enter a Periodical Payment* use case comprises a basic flow, while the *Authorize Payments* use case comprises an alternative flow. The extension specification is acceptable using extension points (Jacobson) or declaratively (Cockburn). The use case description shouldn't contain user interface elements.