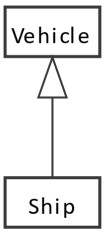
Exercises

Question 1 (single choice):

- Topic: UML basics
- Question Text: What is the relationship between “Vehicle” and “Ship” and how do we express that in a UML class diagram?
- Create 5 answers with the attached SVG files, and let option 6 be “None of the above”
- Answer 2 is correct:



- Explanation: Ship is a subclass of vehicle. We can say “A ship IS-A vehicle.”

Question 2 (single choice):

- Topic: UML basics
- Question Text: What is the relationship between “Book” and “Title” and how do we express that in a UML class diagram?
- Create 5 answers with the attached SVG files, and let option 6 be “None of the above”
- Answer 4 is correct:

- Explanation: A title is an attribute (aka instance variable) of the class book.

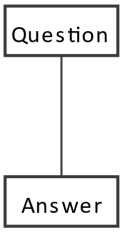
Question 3 (single choice):

- Topic: UML basics
- Question Text: What is the relationship between “Person” and “Albert Einstein” and how do we express that in a UML class diagram?
- Create 5 answers with the attached SVG files, and let option 6 be “None of the above”
- Answer 6 is correct – “None of the above”
- Explanation: This is a typical case where “Albert Einstein” is an object (aka instance) of the type “Person”. One way to convince ourselves that this is not a case for inheritance is that saying “An Albert Einstein IS-A person” – as in “*Every* Albert Einstein IS-A person” doesn’t make sense.

Question 4 (single choice):

- Topic: UML basics

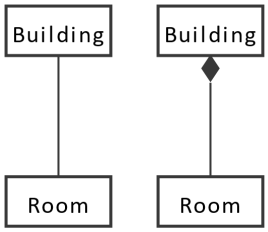
- Question Text: What is the relationship between “Question” and “Answer” and how do we express that in a UML class diagram?
- Create 5 answers with the attached SVG files, and let option 6 be “None of the above”
- Answer 1 is correct



- Explanation: The classes “Answer” and “Question” are associated. We can easily imagine that a question class contains an array of answers. While it is not completely wrong to say that answer is an attribute of question, we probably want answer to have some attributes of itself (answer title, answer text, …) and thus make it its own class.

Question 5 (multiple choice):

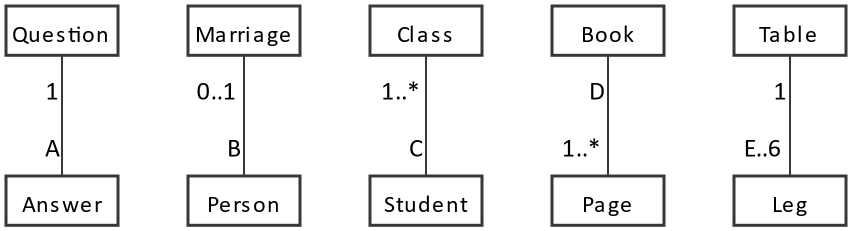
- Topic: UML basics
- Question Text: What is the relationship between “Building” and “Room” and how do we express that in a UML class diagram?
- Create 7 answers from the attached SVG files
- The correct answers are 1 and 5



- Explanation: It is correct to say “A room IS-PART-OF-A building.” Also, when the building gets destroyed the room gets destroyed with it, so this is a case of composition. Since composition is a special case of association, answer 1 is also correct.

Question 6 (fill-in):

- Topic: UML basics
- Question Text: For each of the letters A, B, C, D, E, find the best match – either a number or the symbol * (star).
- Show question6.svg *with* the question:



- Correct answers:
  - A = *
  - B = 2
  - C = *
  - D = 1
  - E = 3

Question 7 (multiple choice):

- Topic: UML basics
- Question Text: Which UML class diagrams can be used to model a binary tree?
- Create 5 answers from the attached SVG files
- Correct Answers: 2, 3, and 4
- Explanation: It is correct to draw UML diagrams to the level of detail that communicates the intent. Leaving out a class for tree or an attribute for value does not mean that they don't exist.
  - Answer 1 is wrong because left and right are not classes, but attributes (and objects at run-time).
  - Answer 2 is the simplest and most straight forward answer.
  - Answer 3 is using the composite pattern. This is how a binary tree would be modelled in many functional languages (e.g. Scala).
  - Answer 4 is similar to the correct answer 2 but gives more detail – it let's us know that one child node association is called "Left" and the other one "Right".
  - Answer 5 is wrong because it allows more than two child nodes and does thus not model a binary tree.

Question 8 (single choice):

- Topic: Patterns
- Question Text: Which pattern would you use in the situation described in the text below? Draw a UML class diagram before you look at the answer.

  You are writing a drawing program in which users can draw circles and rectangles, combine these shapes to form more complicated shapes, save them under a name and reuse them. In the example below a user combined two circles and called that shape "eye", then used that to create a shape "face", and that to create the final drawing.
- Show question8.svg with the question text.
- Answers to choose from: Builder, Command, Composite, Decorator, Façade, Flyweight, Proxy, Observer, Prototype, State, Strategy, Visitor
- The correct answer is "Composite".

- Show question8_answer.svg as an explanation.

Question 9 (single choice):

- Topic: Patterns
- Question Text: Which pattern would you use in the situation described in the text below? Draw a UML class diagram before you look at the answer.

  An application you are designing has to allow the user to authenticate using one of the following three methods:
  1. With their corporate account using the Security Assertion Markup Language (SAML)
  2. With their Google account using OAuth
  3. With their phone number, using a onetime token you send them via text message
- Answers to choose from: Builder, Command, Composite, Decorator, Façade, Flyweight, Proxy, Observer, Prototype, State, Strategy, Visitor
- The correct answer is “Strategy”
- Show question9_answer.svg as an explanation.

Question 10 (single choice):

- Topic: Patterns
- Question Text: An augmented reality gaming platform allows users to create little monsters of different configurations and pin them to geographical locations to interact with other players. They are running into memory problems keeping all the monster objects in memory. What pattern would you recommend investigating?
- Answers to choose from: Builder, Command, Composite, Decorator, Façade, Flyweight, Proxy, Observer, Prototype, State, Strategy, Visitor
- The correct answer is “Flyweight”
- Explanation: The question is kept vague, but of the given choices the only pattern that helps cutting down the memory footprint of a large collection of similar object is the “Flyweight” pattern. In implementing this pattern, we need to decide which part of each monster’s state needs to be stored with the monster object and which part can be externalized to save memory.

Question 11 (single choice):

- Topic: Patterns
- Question Text: Which pattern would you use in the situation described in the text below? Draw a UML class diagram before you look at the answer.

  A text message needs to be sent in different formats:
  - Clear Text
  - Clear Text + Signature
  - Encrypted Text
  - Encrypted Text + Signature
- Answers to choose from: Builder, Command, Composite, Decorator, Façade, Flyweight, Proxy, Observer, Prototype, State, Strategy, Visitor
- The correct answer is “Decorator”

- Explanation: Notice that special attention needs to be paid so that the order of application of the decorators doesn't change the result. It is probably acceptable if applying the "encrypted message decorator" twice also encrypts the message twice. On the other hand, it is (probably) not acceptable if the rendered message is different for these two cases:
    - ("concrete message" decorated with "encrypted message") decorated with "signed message"
    - ("concrete message" decorated with "signed message") decorated with "encrypted message"

  How can we model this and make it clearer in the diagram?
- Show question11_answer.svg as an explanation.

Question 12 (single choice):

- Topic: Patterns
- Question Text: Which pattern would you use in the situation described in the text below? Draw a UML class diagram before you look at the answer.

  A server is accepting and processing requests. In "normal mode" it is accepting every request and puts it in a queue. Once the queue grows too big, it enters "panic mode" in which it drops a certain percentage of requests. If the queue is reaching critical size, it enters "DoS prevention mode" in which it drops every request.
- Answers to choose from: Builder, Command, Composite, Decorator, Façade, Flyweight, Proxy, Observer, Prototype, State, Strategy, Visitor
- The correct answer is "State"
- Show question12_answer.svg as an explanation.

Question 13 (single choice):

- Topic: Patterns
- Question Text: Which pattern would you use in the situation described in the text below? Draw a UML class diagram before you look at the answer.

  A weather station is keeping track of the temperature and humidity. It displays the last measurements on a display and sends them to a mobile phone application.
- Answers to choose from: Builder, Command, Composite, Decorator, Façade, Flyweight, Proxy, Observer, Prototype, State, Strategy, Visitor
- The correct answer is "Observer"
- Show question13_answer.svg as an explanation.

Question 14 (single choice):

- Topic: Patterns
- Question Text: Which pattern would you use in the situation described in the text below? Draw a UML class diagram before you look at the answer.

A web application connects to a server to fetch search results. While the results are loading, it displays the text “Please Wait”. Once all data has been downloaded, it replaces this text with the real result.

- Answers to choose from: Builder, Command, Composite, Decorator, Façade, Flyweight, Proxy, Memento, Prototype, State, Strategy, Visitor
- Note that the options are different.
- The correct answer is “Proxy”
- Show question14_answer.svg above the explanation.
- Explanation: If your intuition was to use the observer pattern (or futures) you are not wrong – we also need a mechanism for notifying the web application when the results have arrived. However, in this diagram we only focus on where the text is coming from. For that purpose, the proxy pattern is ideal.

Question 15 (single choice):

- Topic: Patterns
- Question Text: Which pattern would you use in the situation described in the text below? Draw a UML class diagram before you look at the answer.

  You are writing a chess playing application, in which two players take turns to make moves. Now you want to add an option to take the last move back.
- Answers to choose from: Builder, Command, Composite, Decorator, Façade, Flyweight, Proxy, Memento, Prototype, State, Strategy, Visitor
- The correct answer is “Command”
- Show question15_answer.svg as with the explanation.

Question 16 (single choice):

- Topic: Patterns
- Question Text: Which pattern is described in the text below?

  You are writing a phone application that controls a collection of smart home devices. For example, when the user presses the “Wake Up” button, it opens the blinds, turns on the coffee maker, and changes the radio station to the news.
- Answers to choose from: Builder, Command, Composite, Decorator, Façade, Flyweight, Proxy, Memento, Prototype, State, Strategy, Visitor
- The correct answer is “Façade”

Question 17 (single choice):

- Topic: Patterns
- Question Text: Which pattern would you use in the situation described in the text below? Draw a UML class diagram before you look at the answer.

  You have a collection of geometric shapes (Triangles, Squares, Circles, ...) for which you have written classes in your favorite programming language. You find yourself modifying a lot of

different files, every time you add a new feature (calculate area, calculate circumference, find centroid, …)

- Answers to choose from: Builder, Command, Composite, Decorator, Façade, Flyweight, Proxy, Observer, Prototype, State, Strategy, Visitor
- The correct answer is “Visitor”
- Show question17_answer.svg as an explanation.

Question 18 (single choice):

- Topic: Measuring the quality of a design
- Question Text: Which of the SOLID principles is violated in this design?
- Show question18.svg *with* the question.
- Answers to choose from:
  - Single responsibility principle
  - Open/closed principle
  - Liskov substitution principle
  - Interface segregation principle
  - Dependency inversion principle
- The correct answer is “Liskov substitution principle”
- Explanation: Even though we have learnt in school that every square is (a special kind of) rectangle, this doesn’t mean that we can use the square everywhere we can use the rectangle. A trivial example is, that we cannot use a square to fill space where width and height are different.

  As another example, imagine that the rectangle class has a method to set the width and height (at the same time). If these two values are not the same, we would have to throw an exception in the square subclass. This however is not allowed by the Liskov substitution principle.

Question 19 (single choice):

- Topic: Measuring the quality of a design
- Question Text: Which of the SOLID principles is violated in this design?
- Show question19.svg *with* the question.
- Answers to choose from:
  - Single responsibility principle
  - Open/closed principle
  - Liskov substitution principle
  - Interface segregation principle
  - Dependency inversion principle
- The correct answer is “Liskov substitution principle”
- Explanation: We have seen in the last question that we should not model the square class to be subclass of rectangle. It might be tempting to just reverse the inheritance relationship, but this is also violating the Liskov substitution principle: An invariant of the superclass square is that width and height always have the same value. This invariant however would not be preserved in the subclass rectangle.

Question 20 (single choice):

- Topic: Measuring the quality of a design
- Question Text: Below is the design for a class that loads a file and verifies that it has the correct fingerprint (think MD5 or SHA checksum). Which of the SOLID principles is violated in this design?
- Show question20.svg *with* the question.
- Answers to choose from:
    - Single responsibility principle
    - Open/closed principle
    - Liskov substitution principle
    - Interface segregation principle
    - Dependency inversion principle
- The correct answer is “Single responsibility principle”
- Explanation: Loading a file should be separated from checking the fingerprint. One situation under which we will see why this is not ideal is, when we want to have full (unit) test coverage for this class. We then must write data to the filesystem in our unit test. Without violating the SRP it would be enough to just pass hardcoded test data to our class.