

KTest

高品質 より良いサービス



練習問題

<https://www.ktest.jp>

1年間無料アップデート

Exam : OMG-OCUP2-FOUND100

**Title : OMG Certified UML
Professional 2 (OCUP 2) -
Foundation Level**

Version : DEMO

1.Which modeling relationship allows instances of one class to substitute for instances of another?

- A. auxiliary
- B. association
- C. dependency
- D. replacement
- E. generalization

Answer: E

Explanation:

Generalization in UML is a modeling relationship that connects a general classifier (like a class) to a more specific classifier. It is akin to an "is a" relationship where the specialized element (subclass) inherits features from the general element (superclass), thus allowing instances of the subclass to substitute for instances of the superclass. For example, if "Bird" is a superclass and "Eagle" is a subclass, an instance of "Eagle" can substitute for an instance of "Bird". This relationship is fundamental in object-oriented modeling for representing inheritance. According to the UML 2.5 specification, generalization allows a subclass to inherit part or all of the structure and behavior of a superclass.

2.How is the abstract syntax of UML specified?

- A. using a MOF metamodel
- B. using the Backus-Naur Form (BNF)
- C. using natural language (e.g.. English)
- D. using UML structure and behavior diagrams

Answer: A

Explanation:

The abstract syntax of UML is specified using the Meta-Object Facility (MOF) metamodel. MOF is a modeling language that provides a meta-meta-model at the top layer of the four-layer metadata architecture, which is used to define the metamodels, like the UML. The MOF specification defines the structure and semantics for constructing metamodels, including the UML. By using MOF, UML ensures that its structure is well-defined and can be processed by tools that understand MOF-based metamodels. The use of MOF to specify UML abstract syntax ensures a clear, structured, and standardized method of describing the semantics of UML components, enabling consistent interpretation and implementation across different modeling tools and environments.

3.What represents the most appropriate use of UML during software development?

- A. forcing management decisions
- B. describing a planned or existing system to non-technical stakeholders
- C. capturing and clarifying the business-level concerns of a planned or existing system
- D. capturing the essential characteristics and design decisions of a planned or existing system

Answer: D

Explanation:

The most appropriate use of UML during software development is to capture the essential characteristics and design decisions of a planned or existing system. UML (Unified Modeling Language) is primarily utilized to visually represent the architecture, design, and behavior of a system, which includes detailing the components, relationships, and interactions within the system. This makes it a critical tool for

understanding complex systems and making informed design decisions that align with project requirements and constraints. UML facilitates clear communication among development team members and stakeholders, ensuring that design decisions are well-understood and accurately implemented.

4. For projects involving complex and strategic systems, what is a key advantage of developing models before starting implementation?

- A. Developing models ensures that all requirements will be addressed.
- B. Models are useful to provide proof of progress to project management.
- C. Models help to establish a consensus among all the project stakeholders.
- D. Modeling helps to convince developers that models are necessary for good design.

Answer: C

Explanation:

For projects involving complex and strategic systems, a key advantage of developing models before starting implementation is that models help to establish a consensus among all the project stakeholders. Creating UML models in the early stages of a project provides a visual and conceptual representation of the system that can be easily understood by various stakeholders, including developers, managers, and clients. This facilitates discussions and negotiations about the system's design and functionality, helping to ensure that all parties have a shared understanding and agreement on the project's objectives and solutions before significant resources are invested in implementation.

5. Why are abstractions in a model helpful?

- A. Abstractions add the full detail to the model.
- B. Abstractions can express or suppress detail as needed.
- C. Abstractions can be taken out and the model still makes sense.
- D. Abstractions are not helpful, but rather a distraction in models.

Answer: B

Explanation:

Abstractions in a model are helpful because they can express or suppress detail as needed. This capability is essential in managing complexity in a model by focusing on the high-level, essential aspects of the system while omitting or simplifying the less critical details. This selective detail management aids in understanding and analyzing the system's core functionality without getting overwhelmed by its intricacies. Abstractions facilitate clearer communication, more focused analysis, and more efficient system design by highlighting the most relevant aspects of the system in various contexts.