Question 1

Answer saved

Marked out of 2

Both Java and C# support some form of annotation.

Select one:

- a. False
- b. True

Question 2

Answer saved

Marked out of 2

What is the main motivation for using annotation in OOP design?

Select one:

- a. design is incorrect without annotations
- b. design must be defined using annotation
- c. design is not an OOP one without annotations
- d. helps define the design rules explicitly

Question 3

Answer saved

Marked out of 2

Which annotation(s) are typically used to design an object operation?

Select one:

- a. DomainConstraint, AttrRef.
- b. DomainConstraint, DOpt.
- c. DomOpt, AttribRef.
- d. DOpt, AttrRef.

Question 4

Answer saved

Marked out of 2

What does the `mutable` constraint on an attribute mean?

Select one:

- a. whether or not the attribute's data type can be changed
- b. whether or not the attribute's location can be changed
- c. whether or not the attribute's name can be changed
- d. whether or not the attribute's value can be changed

Question 5

Answer saved

Marked out of 2

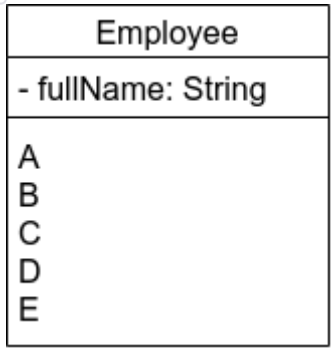
Domain constraints are written in?

Select one:

- a. the @object section
- b. the @attributes section
- c. the @abstract_properties section
- d. the @overview section

Information

Questions 6-15 concern the following scenario. Given the UML class diagram of a class named `Employee`.



Question 6

Answer saved

Marked out of 2

Write the design specification for the attribute `Employee.fullName`. Your specification must include the essential constraints, which are determined based on your practical understanding of a person's date of birth.

- Name: fullName
- Type: String
- Visibility: Private (-)
- Description: represents the full name of an employee.
- Constrain:
  - The attribute must not be null or empty.
  - The attribute must contain at least two space-separated parts, representing the first and last name of the employee.
  - Each part of the name must start with an uppercase letter, followed by one or more lowercase letters.
  - The attribute must not contain any special characters or numbers.
  - The maximum length of the attribute is 70 characters.

Question 7

Answer saved

Marked out of 2

Operation A is an essential operation for creating `Employee` objects. Write the design specification for this operation.

- Name: Employee
- Type: Constructor
- Visibility: Public (+)
- Parameters:
  - `fullName`: a string representing the name of the employee
- `Description`: This constructor is used to create a new `Employee` object with the specified fullName
- Usage: The constructor can be called externally to create a new `Employee` object. Once created, the object can be used to access and modify employee information through its attributes and methods.

Question 8

Answer saved

Marked out of 2

Write the code for operation A.

```
public Employee(String fullName) {
    this.fullName = fullName;
}
```

Question 9

Answer saved

Marked out of 2

Operation B is a mutator operation for attribute `Employee.fullName`. Write the design specification for this operation.

- Name: `NewFullName`
- Type: Method
- Visibility: Public (+)
- Parameters:
  - `newFullName`: A string representing the new value for the `Employee.fullName` attribute.
- Description: This method allows the modification of the `Employee.fullName` attribute. It takes a single string argument, representing the new value for the `Employee.fullName` attribute. The method validates the new value against the constraints defined for the `Employee.fullName` attribute before making any changes. If the new value is valid, the method updates the `Employee.fullName` attribute with the new value and returns `true`. If the new value is not valid, the method does not make any changes to the `Employee.fullName` attribute and returns `false`.

Question 10

Answer saved

Marked out of 2

Write the code of the operation B.

```
        return false;
      }
    }
    if (newFullName.length() > 70) {
        return false;
    }

    // Update the fullName attribute
    this.fullName = newFullName;
    return true;
}
```

Question 11

Answer saved

Marked out of 2

Operation C is a helper operation that validates input value for attribute `Employee.fullName`. Write the design specification for this operation.

- Name: `ValidateFullName`
- Type: Method
- Visibility: Private (-)
- Parameters:
  - `newFullName`: A string representing the new value for the `Employee.fullName` attribute.
- Description: This method is a helper operation that validates the input value for the `Employee.fullName` attribute. It takes a single string argument, representing the new value for the `Employee.fullName` attribute. The method checks if the new value meets the constraints defined for the `Employee.fullName` attribute. If the new value is valid, the method returns `true`. If the new value is not valid, the method returns `false`.
- Usage: This method can be called by other methods within the `Employee` class to validate the input value for the `Employee.fullName` attribute

Question 12

Answer saved

Marked out of 2

Write the code for operation C.

```
for (String part : parts) {
    if (!part.matches("[A-Z][a-z]+")) {
        return false;
    }
}
if (newFullName.length() > 70) {
    return false;
}

// Return true if the new value is valid
return true;
}
```

Question 13

Answer saved

Marked out of 2

Operation D extracts the `firstName` from `Employee.fullName`. Assume that name follows the Vietnamese naming convention (i.e. family name appears first). Write the design specification for the operation.

- Name: `ExtractFirstName`
- Type: Method
- Visibility: Public (+)
- Parameters: None
- Description: This method extracts the first name from the `Employee.fullName` attribute, assuming that the name follows the Vietnamese naming convention where the family name appears first. The method splits the `Employee.fullName` attribute by space and returns the last part of the split string, representing the first name of the employee.
- Usage: This method can be called on an instance of the `Employee` class to extract the first name from the `fullName` attribute.

Question 14

Answer saved

Marked out of 2

Write the code for operation D.

```
public String operationD() {
    // Split the fullName attribute by space
    String[] parts = this.fullName.split(" ");

    // Return the last part of the split string
    return parts[parts.length - 1];
}
```

Question 15

Answer saved

Marked out of 2

Operation E is a Java's default operation that produces a string representation of a `Employee` object. Write the design specification and code of this operation.

- Name: `toString`

- Type: Method

- Visibility: Public (+)

- Parameters: None

- Description: This method is a Java's default operation that produces a string representation of an `Employee` object. The method overrides the `toString()` method from the `Object` class and returns a string representation of the `Employee` object, including its attributes and their values.

- Usage: This method can be called on an instance of the `Employee` class to produce a string representation of the object.

```java
@Override
```

◀ Assignment 2 - Redo1

Jump to...

Question 16

Answer saved

Marked out of 2

Which of the followings is the name of a software engineering process model?

Select one:

- a. decomposition by abstraction
- b. step-wise
- c. waterfall
- d. spine
- e. development cycle

Question 17

Answer saved

Marked out of 2

Which of the followings best describes software engineering?

Select one:

- a. produce large and complex software for important purposes
- b. the application of a disciplined process to the development of software
- c. help programmers to effectively use a programming language to code a software
- d. a method to automatically produce software using machinery

Question 18

Answer saved

Marked out of 2

Which of the followings best describes the phases of a software engineering process?

Select one:

- a. requirement manufacturing, design, implementation
- b. requirement capturing, design, testing
- c. software requirement, design, implementation
- d. requirement analysis, implementation, testing
- e. requirement design, implementation, testing

Question 19

Answer saved

Marked out of 2

The agile process model

Select one:

- a. involves working closely with the clients to develop software
- b. emphasises on documenting the interaction with the customers
- c. employs the client to work as an employee in the development team
- d. divides the development process into increments of 2-3 months

Question 20

Answer saved

Marked out of 2

Which of the followings is the name of standard software engineering process?

Select one:

- a. software life cycle engineering
- b. standard software engineering cycle
- c. software system engineering process
- d. software development life cycle
- e. standard software engineering process

Question 21

Answer saved

Marked out of 2

Which of the followings best describes the concern of the functional requirement of a software?

Select one:

- a. function and data
- b. data and functional abstraction
- c. procedural and data specification
- d. end-user service and data

Question 22

Answer saved

Marked out of 2

Which of the followings lists the types of non-functional requirements?

Select one:

- a. performance, reusability
- b. reusability, accuracy
- c. modifier, performance
- d. accuracy, modifier

Question 23

Answer saved

Marked out of 2

Which of the followings describes the types of user interaction that could be captured in a use case?

Select one:

- a. normal and advanced
- b. normal and extended
- c. basic and advanced
- d. regular and erroneous

Question 24

Answer saved

Marked out of 2

Which of the followings best describes a statement about the input data of a software?

Select one:

- a. all popular science-fiction books are entered into the system by data entry staff
- b. a customer asks a sales staff to search for documents about a book
- c. every staff is employed by the book shop to work full-time
- d. a customer has name, date of birth, and address

Question 25

Answer saved

Marked out of 2

Why is it important to capture the non-functional requirements of a software?

Select one:

- a. to understand the necessary constraints on the system functions
- b. to understand the goal and objectives of the software
- c. to understand the non-essential services that the software performs
- d. to understand other aspects of the software that can not be explained clearly

◀ Assignment 2 - Redo1

Jump to...

Question 26

Answer saved

Marked out of 2

What is the primary purpose of a use case diagram?

Select one:

- (●) a. to show the user interactions in different scenarios
- ( ) b. to show a diagram of how to use the system
- ( ) c. to show how the software operates in different situations
- ( ) d. to describe the details of all the use cases of the software

Question 27

Answer saved

Marked out of 2

Which of the followings lists the basic modelling constructs of an UML class diagram?

Select one:

- ( ) a. class, entity
- ( ) b. entity, association
- (●) c. class, association
- ( ) d. relationship, entity
- ( ) e. class, relationship

Question 28

Answer saved

Marked out of 2

The requirement specification of a software function

Select one:

- ( ) a. does not include the return type nor the pre-conditions
- (●) b. makes precise the pre- and post-conditions of the function
- ( ) c. defines the function with all the required information
- ( ) d. describes the detailed input and output of the function

Question 29

Answer saved

Marked out of 2

Which of the followings is an essential characteristic of the step that specifies software requirement?

Select one:

- (●) a. the software operates as required by the user
- ( ) b. the components that make up the software are defined
- ( ) c. the requirement is precisely written in the programming language
- ( ) d. the top-level use cases become software functions

Question 30

Answer saved

Marked out of 2

Which of the followings is a constraint on an attribute?

Select one:

- a. a word may appear in one or more documents
- b. the frequency of each word in a document is the number of its occurrences
- c. a document matches a query if it contains all of the query's keywords
- d. each document may have one or more words

Question 31

Answer saved

Marked out of 2

Which of the following best describes the object-oriented design technique named "decomposition by abstraction"?

Select one:

- a. decompose each software function and create/use abstractions as required to perform the functions
- b. decompose a system into functions and to create/use procedures to perform them
- c. iteratively decompose a system into smaller systems
- d. decompose a system into classes and to create/use methods of these classes

Question 32

Answer saved

Marked out of 2

What are the first two design iterations called?

Select one:

- a. initial and top-level abstractions
- b. important and key abstractions
- c. beginning and preparation
- d. preparation and starting

Question 33

Answer saved

Marked out of 2

Which of the following is the goal of software design?

Select one:

- a. to construct a detailed drawing of the software
- b. to construct the precise specification for how the software will perform
- c. to write a software specification for the designer to understand
- d. to write a software specification for the users to understand

Question 34

Answer saved

Marked out of 2

The primary purpose of the design note is to:

Select one:

- a. record confidential design information that are not included in other documents
- b. draw a picture of the system and briefly describes it
- c. systematically analyse the requirements in order to make better design decisions
- d. make comments about the customer's design

Question 35

Answer saved

Marked out of 2

Design refinement means:

Select one:

- a. incrementally add detail and precision to the design
- b. incrementally produce a more refined product
- c. iteratively make the design size smaller
- d. iteratively cut the design into smaller elements

◀ Assignment 2 - Redo1

Jump to...

Question 36

Answer saved

Marked out of 2

A top-down implementation plan has the following features

Select one:

- a. late detection of errors and increased re-use in testing
- b. early detection of errors and decreased re-use in testing
- c. early detection of errors and increased re-use in testing
- d. late detection of errors and decreased re-use in testing

Question 37

Answer saved

Marked out of 2

A top-down implementation plan is

Select one:

- a. coding a module before those that use it
- b. coding a module before those that it uses
- c. implementing more important modules before those that are less important
- d. implementing the header before the body of a module

Question 38

Answer saved

Marked out of 2

A bottomp-up implementation plan has the following features

Select one:

- a. early protypes of the system and less up-front resources
- b. early protypes of the system and more up-front resources
- c. early protypes of the smaller system and more up-front resources
- d. early protypes of smaller systems and less up-front resources

Question 39

Answer saved

Marked out of 2

A bottom-up implementation plan is

Select one:

- a. coding a module before those that it uses
- b. implementing the header before the body of a module
- c. coding a module before those that use it
- d. implementing more important modules before those that are less important

Question 40

Answer saved

Marked out of 2

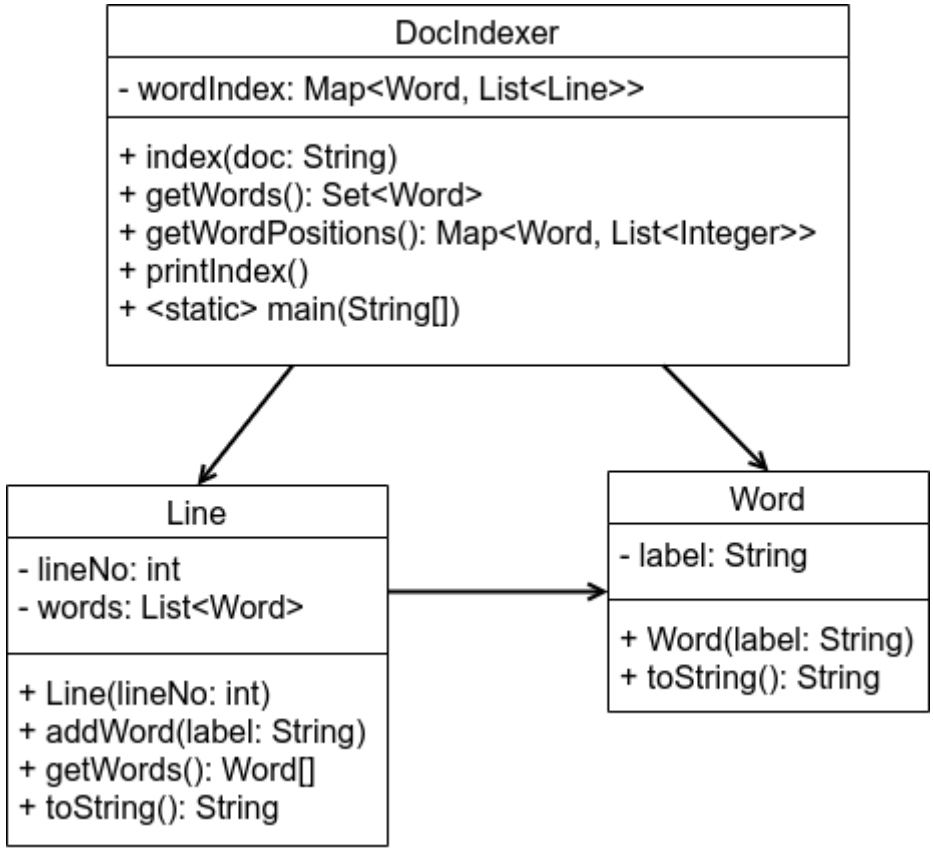
Which of the followings lists the software design evaluation criteria?

Select one:

- a. effectiveness, correctness
- b. correctness, performance
- c. performance, effectiveness
- d. simplicity, performance
- e. simplicity, generality

Information

Questions 41-48 concern the following scenario. Given the UML class diagram of a program named `DocIndexer`. Assume that the classes `Line` and `Word` and the method `main` have been implemented. Below are the design specifications of four methods `index`, `getWords`, `getWordPositions` and `printIndex`.



```
/**
* @requires doc is not null and a text document
* @effects
*   (1) if doc is not empty
*   (2)   for each line ls in doc, whose line number is lineNo
*   (3)     let l = Line(lineNo)
*   (4)     extract words in ls that have > 1 letter and add them t
o l as Words
*   (5)     for each Word w of l
*   (6)       update (w,l) into wordIndex
*/
public void index(String doc)

/**
* @effects
*   (1) if wordIndex is not empty
*   (2)  return the Set of Words in wordIndex
*   (3) else
*   (4)  return null
*/
public Set getWords()

/**
* @effects
*   (1) if wordIndex is not empty
*   (2)   return Map where each entry (w,l) is mapped to
*   (3)   a corresponding entry (w,lo) in wordIndex as follows:
*   (4)     l.get(i) = lo.get(i).lineNo, for all i = 0 to lo.size()
*   (5) else
*   (6)   return null
*/
public Map> getWordPositions()

/**
* @effects
*   (1) if this.wordIndex is null
*   (2)   print to the standard console "empty"
*   (3) else
*   (4)   for each tuple (w,lines) in this.wordIndex
*   (5)   print to the standard output w + ": "+ lines
*/
public void printIndex()
```

Question 41

Answer saved

Marked out of 2

Briefly describe an alternative design of class DocIndexer that does not need to include the two operations `getWords` and `getWordPositions`.

An alternative design of the `DocIndexer` class that does not include the `getWords` and `getWordPositions` operations could involve directly accessing the `wordIndex` data structure within the `printIndex` method. Instead of calling the `getWords` and `getWordPositions` methods to retrieve the necessary data, the `printIndex` method could iterate over the entries in the `wordIndex` map and extract the necessary information directly. This would eliminate the need for the two separate operations while still allowing for the printing of the index.

Question 42

Answer saved

Marked out of 6

About method `index`: write the code that implements the behaviour description.

```
public void index(String doc) {
    if (doc == null) {
        throw new IllegalArgumentException("doc must not be null");
    }
    String[] lines = doc.split("\n");
    for (int lineNo = 0; lineNo < lines.length; lineNo++) {
        Line l = new Line(lineNo);
        String[] words = lines[lineNo].split("\\s+");
        for (String word : words) {
            if (word.length() > 1) {
                Word w = new Word(word);
                l.addWord(word);
```

Question 43

Answer saved

Marked out of 2

About method `getWords`: write the code that implements the behaviour description line (2).

```
return new HashSet<>(wordIndex.keySet());
```

Question 44

Answer saved

Marked out of 2

About method `getWordPositions`: write the code that implements the behaviour description lines (2-4).

```
Map<Word, List<Integer>> wordPositions = new HashMap<>();
for (Map.Entry<Word, List<Line>> entry : wordIndex.entrySet()) {
    Word word = entry.getKey();
    List<Line> lines = entry.getValue();
    List<Integer> lineNumbers = new ArrayList<>();
    for (Line line : lines) {
        lineNumbers.add(line.lineNo);
    }
    wordPositions.put(word, lineNumbers);
}
return wordPositions;
```

Question 45

Answer saved

Marked out of 2

About method `printIndex`: write the code that implements the behaviour description lines (4) and (5).

```
for (Map.Entry<Word, List<Line>> entry : wordIndex.entrySet()) {
    Word word = entry.getKey();
    List<Line> lines = entry.getValue();
    System.out.println(word + ": " + lines);
}
```

Question 46

Answer saved

Marked out of 2

DocIndexer depends on Line because:

Select one:

- (●) a. It includes Line as part of its design.
- ( ) b. It references the attribute Line.words.
- ( ) c. Its main method takes a Line object as input.
- ( ) d. It needs to invoke a Line's method.

Question 47

Answer saved

Marked out of 2

Which of the following dependencies in the diagram can be replaced by a weak dependency:

Select one:

- ( ) a. association (WordIndex, Word).
- ( ) b. association (Line, Word).
- (●) c. association (WordIndex, Line).

Question 48

Answer saved

Marked out of 2

Word can be replaced by the Java's String class:

Select one:

- ( ) a. False.
- (●) b. True.

◀ Assignment 2 - Redo1

Jump to...