Final SE:

Question 1: An operation’s parameter can not be attached to an annotation:

Select one:

1. A.False
2. B.True

Câu hỏi: Một tham số của một hoạt động không thể được gắn vào một chú thích?

Giải thích: Đáp án là B. Đúng. Trong ngữ cảnh của câu hỏi, "annotation" được hiểu là một chú thích hoặc một thông tin bổ sung liên quan đến một hoạt động. Trong nhiều ngôn ngữ lập trình, tham số là một giá trị được truyền vào hoạt động để thực hiện một chức năng cụ thể. Tham số này không thể được gắn kết hoặc liên kết trực tiếp với chú thích, vì chúng là hai khái niệm khác nhau trong quá trình lập trình.

Question 2: What is the main motivation for using annotation in OOP design?

Select one:

1. A.help defines the design rules explicitly
2. B.design is not an OOP one without annotations
3. C.design must be defined using annotation
4. D.design is incorrect without annotations

Câu hỏi: Động lực chính để sử dụng chú thích trong thiết kế OOP là gì?

Giải thích đáp án: Đáp án là A. Giúp xác định rõ các quy tắc thiết kế.

- Chú thích trong lập trình hướng đối tượng (OOP) cung cấp thông tin bổ sung về các phần tử trong chương trình như lớp, phương thức, biến.

- Chúng giúp xác định rõ quy tắc thiết kế và tạo tài liệu cho chúng.

- Chú thích có thể được sử dụng để tạo mã, kiểm tra, áp dụng ràng buộc và kích hoạt hành vi.

- Chúng tăng cường sự rõ ràng và hiểu biết về mã nguồn, giúp truyền tải ý định thiết kế và tuân thủ các mô hình hoặc hướng dẫn thiết kế.

- Tuy nhiên, chú thích không bắt buộc trong thiết kế OOP và chỉ là một tính năng tùy chọn mang lại khả năng biểu đạt và linh hoạt trong quá trình thiết kế.

Question 3: Behavior differs from implementation (coding) because?

Select one:

1. A.it presents the code of the procedure
2. B.it describes how rather what a procedure performs.
3. C.it describes what rather what a procedure performs.
4. D.it formats the code in the Javadoc format.

According to your teacher, "behavior differs from implementation (coding) because it describes what rather than what a procedure performs." This means that behavior describes the intended outcome or functionality of a procedure, while implementation (coding) describes how that behavior is achieved.

Question 4: What does the optional constraint on an attribute mean?

Select one:

1. A.whether or not the attribute can be defined with the keyword “optional”
2. B.whether or not the attribute’s value must be initialised
3. C.whether or not the attribute can be turned off
4. D.whether or not the attribute’s can be defined in the class

Câu hỏi 4: Ý nghĩa của ràng buộc tùy chọn (optional constraint) trên một thuộc tính là gì?

Lựa chọn một trong những đáp án sau:

A. thuộc tính có thể được định nghĩa với từ khóa "optional" hay không.

B. giá trị của thuộc tính phải được khởi tạo hay không.

C. thuộc tính có thể bị tắt hay không.

D. thuộc tính có thể được định nghĩa trong lớp hay không.

Giải thích đáp án:

Đáp án đúng là A. Cho dù thuộc tính có thể được định nghĩa với từ khóa "optional" hay không.

Ràng buộc tùy chọn (optional constraint) trên một thuộc tính trong ngữ cảnh này chỉ ra xem liệu thuộc tính đó có thể được xác định một cách tùy ý hay không. Nếu thuộc tính được đánh dấu là tùy chọn (optional), có thể có trường hợp thuộc tính đó không được xác định hoặc không có giá trị nào được gán cho nó. Trong trường hợp này, từ khóa "optional" có thể được sử dụng để chỉ rõ rằng thuộc tính là tùy chọn và không bắt buộc phải có giá trị.

Các đáp án B, C và D không chính xác với ý nghĩa của ràng buộc tùy chọn trên một thuộc tính.

Question 5: Concrete data type means that

Select one:

1. A.it is defined by the lass named concrete
2. B.it is fixed but can be changed by the user
3. C.it realizes the format data type in an OOPL
4. D.it is fixed and can not be changed

Câu hỏi 5: "Concrete data type" có nghĩa là gì?

Lựa chọn một trong những đáp án sau:

A. Nó được xác định bởi lớp có tên "concrete".

B. Nó là cố định nhưng có thể được thay đổi bởi người dùng.

C. Nó thực hiện kiểu dữ liệu định dạng trong ngôn ngữ lập trình hướng đối tượng (OOPL).

D. Nó là cố định và không thể thay đổi.

Giải thích đáp án:

Đáp án đúng là D. Nó là cố định và không thể thay đổi.

"Concrete data type" (kiểu dữ liệu cụ thể) trong ngữ cảnh này đề cập đến một kiểu dữ liệu cụ thể và không thể thay đổi. Nó được sử dụng để mô tả kiểu dữ liệu mà giá trị của nó không thể được thay đổi sau khi được khởi tạo.

Ví dụ, trong một ngôn ngữ lập trình, nếu một biến được khai báo với một kiểu dữ liệu cụ thể, giá trị của biến đó sẽ không thể thay đổi trong quá trình chạy chương trình. Một khi giá trị đã được gán cho biến, nó sẽ cố định và không thể thay đổi.

Các đáp án A, B và C không phản ánh đúng ý nghĩa của "concrete data type" trong ngữ cảnh này.

Question 16: Which of the following is the name of a software engineering process model ?

Select one:

1. A.decomposition by abstraction
2. B.step-wise
3. C.development cycle
4. D.waterfall
5. E.spine

Câu hỏi 16: Trong các tùy chọn sau, cái nào là tên của một mô hình quy trình kỹ thuật phần mềm?

Lựa chọn một trong những đáp án sau:

A. Phân tách bằng trừu tượng (decomposition by abstraction)

B. Theo bước

C. Chu kỳ phát triển

D. Mô hình thác nước (waterfall)

E. Xương sống (spine)

Đáp án đúng là D. Mô hình thác nước (waterfall).

"Mô hình thác nước" (waterfall model) là một mô hình phát triển phần mềm dựa trên một quy trình tuần tự và tuyến tính. Theo mô hình này, các giai đoạn phát triển như phân tích, thiết kế, triển khai, kiểm thử và bảo trì được thực hiện theo một luồng tuần tự, tương tự như các bước trong một chuỗi thác nước.

Các đáp án A, B, C và E không phải là tên của một mô hình quy trình kỹ thuật phần mềm.

Đáp án A, B, C và E không phải là tên của các mô hình quy trình kỹ thuật phần mềm cụ thể.

- A. Phân tách bằng trừu tượng (decomposition by abstraction): Đây không phải là tên của một mô hình quy trình kỹ thuật phần mềm, mà là một khái niệm để mô tả quá trình phân rã một hệ thống phức tạp thành các phần nhỏ hơn thông qua việc trừu tượng hóa.

- B. Theo bước (step-wise): Đây không phải là tên của một mô hình quy trình kỹ thuật phần mềm cụ thể, mà chỉ đơn giản là một thuật ngữ để chỉ sự thực hiện công việc theo từng bước một.

- C. Chu kỳ phát triển (development cycle): Đây không phải là tên của một mô hình quy trình kỹ thuật phần mềm cụ thể, mà chỉ ám chỉ một chu kỳ phát triển tổng quát hoặc một quá trình lặp lại trong phát triển phần mềm.

- E. Xương sống (spine): Đây không phải là tên của một mô hình quy trình kỹ thuật phần mềm cụ thể. "Xương sống" có thể được sử dụng trong ngữ cảnh khác, nhưng không liên quan trực tiếp đến các mô hình quy trình phần mềm.

Mô hình thác nước (waterfall model) là mô hình quy trình kỹ thuật phần mềm duy nhất trong danh sách tùy chọn ban đầu.

Question 17: WHich of the following best describes the phase of a software engineering process?

Select one:

1. A.requirement analysis, implementation, testing
2. B.software requirement, design, implementation
3. C.requirement manufacturing, design, implementation
4. D.requirement capturing, design, testing

Câu hỏi 17: Trong các tùy chọn sau, cái nào mô tả tốt nhất giai đoạn của quy trình kỹ thuật phần mềm?

Lựa chọn một trong những đáp án sau:

A. Phân tích yêu cầu, triển khai, kiểm thử

B. Yêu cầu phần mềm, thiết kế, triển khai

C. Sản xuất yêu cầu, thiết kế, triển khai

D. Thu thập yêu cầu, thiết kế, kiểm thử

Đáp án đúng là B. Yêu cầu phần mềm, thiết kế, triển khai.

Giai đoạn trong quy trình kỹ thuật phần mềm thường bao gồm các hoạt động như phân tích yêu cầu, thiết kế và triển khai.

- A. Phân tích yêu cầu, triển khai, kiểm thử: Thiếu giai đoạn thiết kế trong quy trình.

- C. Sản xuất yêu cầu, thiết kế, triển khai: Thuật ngữ "sản xuất yêu cầu" không phù hợp với quy trình kỹ thuật phần mềm và thiếu giai đoạn phân tích yêu cầu.

- D. Thu thập yêu cầu, thiết kế, kiểm thử: Giai đoạn "kiểm thử" không được đề cập trong câu hỏi.

Do đó, đáp án B là đáp án chính xác nhất vì nó mô tả các giai đoạn yêu cầu phần mềm, thiết kế và triển khai trong quy trình kỹ thuật phần mềm.

Question 18: Which of the following best describes software engineering ?

Select one:

1. A.the application of a disciplined process to the development of software
2. B.produce large and complex software for important purpose
3. C.help programmers to effectively use a programming language to code a software
4. D.a method to automatically produce software using machinery

Câu hỏi 18: Trong các tùy chọn sau, cái nào mô tả tốt nhất kỹ thuật phần mềm?

Lựa chọn một trong những đáp án sau:

A. Áp dụng quy trình kỷ luật trong phát triển phần mềm.

B. Sản xuất phần mềm lớn và phức tạp cho mục đích quan trọng.

C. Giúp các lập trình viên sử dụng hiệu quả ngôn ngữ lập trình để viết mã phần mềm.

D. Một phương pháp tự động sản xuất phần mềm bằng máy móc.

Đáp án đúng là A. Áp dụng quy trình kỷ luật trong phát triển phần mềm.

Kỹ thuật phần mềm (software engineering) là việc áp dụng quy trình kỷ luật và phương pháp hệ thống để phát triển phần mềm. Nó bao gồm việc áp dụng nguyên tắc kỹ thuật, phương pháp và công cụ trong việc quản lý, thiết kế, xây dựng và kiểm thử phần mềm.

Các đáp án còn lại không phản ánh đúng khái niệm của kỹ thuật phần mềm:

- B. Sản xuất phần mềm lớn và phức tạp cho mục đích quan trọng: Mô tả không đầy đủ và hướng đến quy mô và phức tạp của phần mềm mà không tập trung vào quy trình kỹ thuật.

- C. Giúp các lập trình viên sử dụng hiệu quả ngôn ngữ lập trình để viết mã phần mềm: Mô tả hướng dẫn cho việc sử dụng ngôn ngữ lập trình hơn là kỹ thuật phần mềm tổng thể.

- D. Một phương pháp tự động sản xuất phần mềm bằng máy móc: Mô tả một phương pháp tự động hóa việc sản xuất phần mềm bằng các máy móc, nhưng không phản ánh đầy đủ khái niệm kỹ thuật phần mềm.

Vì vậy, đáp án chính xác là A: Áp dụng quy trình kỷ luật trong phát triển phần mềm.

Question 19: Which of the following is an essential characteristic of software?

Select one:

1. A.a system that is made up of soft materials
2. B.result of a creative process of the human brain
3. C.provide output for the human brain
4. D.is a manufactured part of a computer
5. E.provide input for the human brain

Câu hỏi 19: Trong các tùy chọn sau, cái nào là đặc điểm cần thiết của phần mềm?

Lựa chọn một trong những đáp án sau:

A. Hệ thống được tạo thành từ vật liệu mềm.

B. Kết quả của quá trình sáng tạo trong não bộ con người.

C. Cung cấp đầu ra cho não bộ con người.

D. Là một bộ phận được sản xuất của máy tính.

E. Cung cấp đầu vào cho não bộ con người.

Đáp án đúng là B. Kết quả của quá trình sáng tạo trong não bộ con người.

Phần mềm là một sản phẩm của quá trình sáng tạo trong não bộ con người. Nó được tạo ra thông qua việc thiết kế, lập trình và phát triển bởi những người lập trình viên và kỹ sư phần mềm. Phần mềm không phải là một hệ thống vật liệu mềm (đáp án A), không phải là một bộ phận được sản xuất của máy tính (đáp án D), không cung cấp đầu vào cho não bộ con người (đáp án E), và không phải là cung cấp đầu ra cho não bộ con người (đáp án C).

Vì vậy, đáp án chính xác là B: Kết quả của quá trình sáng tạo trong não bộ con người.

Question 20: The agile process model:

Select one:

1. A.emphasis on documenting the interaction with the customers
2. B.divides the development process into increments of 2-3 months
3. C.involves working closely with the clients to develop software
4. D.employs the client to work as an employee in the development team

Câu hỏi 20: Mô hình quy trình Agile (agile process model):

Lựa chọn một trong những đáp án sau:

A. Tập trung vào việc tài liệu hóa tương tác với khách hàng.

B. Chia quá trình phát triển thành các giai đoạn kéo dài 2-3 tháng.

C. Liên quan chặt chẽ với khách hàng để phát triển phần mềm.

D. Sử dụng khách hàng làm nhân viên trong nhóm phát triển.

Đáp án đúng là C. Liên quan chặt chẽ với khách hàng để phát triển phần mềm.

Mô hình quy trình Agile là một phương pháp phát triển phần mềm linh hoạt và tương tác, trong đó sự tương tác và phản hồi với khách hàng được coi là quan trọng. Mô hình này giúp tạo điều kiện để làm việc gần gũi với khách hàng, hiểu rõ nhu cầu của họ và phản hồi nhanh chóng để phát triển phần mềm theo yêu cầu. Mô hình Agile không phân chia quá trình phát triển thành các giai đoạn kéo dài 2-3 tháng (đáp án B), không tập trung vào việc tài liệu hóa tương tác với khách hàng (đáp án A), và không sử dụng khách hàng làm nhân viên trong nhóm phát triển (đáp án D).

Vì vậy, đáp án chính xác là C: Liên quan chặt chẽ với khách hàng để phát triển phần mềm.

Question 21: Which of the following is the name of an object-oriented modeling language?

Select one:

1. A.undefined modeling specification
2. B.uniform modeling language
3. C.unified modeling language’
4. D.universal object modeling
5. E.unified modeling system

Câu hỏi 21: Trong các tùy chọn sau, cái nào là tên của một ngôn ngữ mô hình hướng đối tượng?

Lựa chọn một trong những đáp án sau:

A. Mô hình đặc tả chưa được xác định

B. Ngôn ngữ mô hình thống nhất

C. Ngôn ngữ mô hình thống nhất

D. Mô hình đối tượng toàn cầu

E. Hệ thống mô hình thống nhất

Đáp án đúng là C. Ngôn ngữ mô hình thống nhất.

Unified Modeling Language (UML) là một ngôn ngữ mô hình hóa tiêu chuẩn được sử dụng trong kỹ thuật phần mềm để biểu diễn, chỉ định, xây dựng và tài liệu hóa các thành phần của một hệ thống phần mềm. Nó cung cấp một tập hợp các biểu đồ và ký hiệu đồ họa để biểu diễn cấu trúc, hành vi và tương tác của các hệ thống phần mềm.

Các tùy chọn còn lại không đúng:

- A. Mô hình đặc tả chưa được xác định: Tùy chọn này không đề cập đến bất kỳ ngôn ngữ mô hình hướng đối tượng cụ thể nào.

- B. Ngôn ngữ mô hình thống nhất: Không có ngôn ngữ mô hình hướng đối tượng nào được công nhận với tên gọi này.

- D. Mô hình đối tượng toàn cầu: Đây không phải là thuật ngữ công nhận cho một ngôn ngữ mô hình hướng đối tượng.

- E. Hệ thống mô hình thống nhất: Không có ngôn ngữ mô hình hướng đối tượng nào được công nhận với tên gọi này.

Vì vậy, đáp án đúng là C. Ngôn ngữ mô hình thống nhất.

Question 22: Why is it important to capture the non-functional requirements of a software?

Select one:

1. A.to understand other aspects of the software that can not be explained clearly
2. B.to understand the goal and objective of the software
3. C.to understand the non- essential services that the software performs
4. D.to understand the necessary constraints on the system functions

Câu hỏi 22: Tại sao việc ghi nhận các yêu cầu phi chức năng của phần mềm là quan trọng?

Lựa chọn một trong những đáp án sau:

A. Để hiểu các khía cạnh khác của phần mềm mà không thể giải thích một cách rõ ràng.

B. Để hiểu mục tiêu và mục đích của phần mềm.

C. Để hiểu các dịch vụ không cần thiết mà phần mềm thực hiện.

D. Để hiểu các ràng buộc cần thiết về chức năng hệ thống.

Đáp án đúng là D. Để hiểu các ràng buộc cần thiết về chức năng hệ thống.

Việc ghi nhận các yêu cầu phi chức năng của phần mềm là quan trọng vì chúng giúp hiểu các ràng buộc và yêu cầu về hệ thống mà không liên quan trực tiếp đến chức năng của phần mềm. Các yêu cầu phi chức năng thường liên quan đến hiệu suất, độ tin cậy, bảo mật, khả năng mở rộng, khả năng sử dụng, và các ràng buộc về quản lý và quy định. Việc hiểu các yêu cầu phi chức năng là cần thiết để đảm bảo rằng hệ thống phần mềm được xây dựng và triển khai đúng các yêu cầu này.

Các đáp án còn lại không chính xác:

- A. Để hiểu các khía cạnh khác của phần mềm mà không thể giải thích một cách rõ ràng: Các yêu cầu phi chức năng không nhất thiết liên quan đến việc giải thích các khía cạnh khác của phần mềm.

- B. Để hiểu mục tiêu và mục đích của phần mềm: Mục tiêu và mục đích của phần mềm thường được xác định bằng các yêu cầu chức năng và không nhất thiết phụ thuộc vào yêu cầu phi chức năng.

- C. Để hiểu các dịch vụ không cần thiết mà phần mềm thực hiện: Yêu cầu phi chức năng không liên quan đến các dịch vụ không cần thiết, mà thường tập trung vào các yêu cầu bắt buộc hoặc cần thiết.

Vì vậy, đáp án chính xác là D. Để hiểu các ràng buộc cần thiết về chức năng hệ thống.

Question 23: WHich of the following is the input for software design ?

Select one:

1. A.User requirements
2. B.requirement specification
3. C.software requirements
4. D.requirement analysis

Câu hỏi 23: Trong số các lựa chọn sau, cái nào là đầu vào cho thiết kế phần mềm?

Chọn một:

A. Yêu cầu của người dùng (User requirements)

B. Đặc tả yêu cầu (requirement specification)

C. Yêu cầu phần mềm (software requirements)

D. Phân tích yêu cầu (requirement analysis)

E. requirement engineering

Câu trả lời đúng là A. Yêu cầu của người dùng (User requirements).

Yêu cầu của người dùng là thông tin về nhu cầu, kỳ vọng và thông số kỹ thuật được cung cấp bởi người dùng cuối hoặc các bên liên quan đến phần mềm. Những yêu cầu này mô tả các chức năng và tính năng mà phần mềm nên có để đáp ứng nhu cầu và mục tiêu của người dùng. Yêu cầu của người dùng đóng vai trò là đầu vào cho quá trình thiết kế phần mềm, vì chúng cung cấp thông tin cần thiết để định nghĩa kiến trúc, các thành phần và cấu trúc tổng thể của hệ thống phần mềm.

Các lựa chọn còn lại không phải là đầu vào trực tiếp cho thiết kế phần mềm:

- B. Đặc tả yêu cầu: Đây là một tài liệu chi tiết mô tả yêu cầu phần mềm, nhưng nó được tạo ra từ yêu cầu của người dùng và không phải là đầu vào trực tiếp cho thiết kế phần mềm.

- C. Yêu cầu phần mềm: Yêu cầu phần mềm là một thuật ngữ rộng hơn bao gồm yêu cầu của người dùng, yêu cầu hệ thống và các đặc tả khác. Mặc dù yêu cầu phần mềm được xem xét trong quá trình thiết kế phần mềm, nhưng nó không phải là đầu vào trực tiếp.

- D. Phân tích yêu cầu: Phân tích yêu cầu là quá trình hiểu, làm rõ và xác định các yêu cầu. Nó là một bước trong vòng đời phát triển phần mềm, nhưng không phải là đầu vào trực tiếp cho thiết kế phần mềm.

Vì vậy, câu trả lời đúng là A. Yêu cầu của người dùng (User requirements).

Question 24: Which of the following best describes a statement about the input data of a software?

Select one:

a. A.every staff is employed by the book shop to work full-time

b. B.a customer asks a sales staff to search for documents about a book

c. C.a customer has name, date of birth, and address

d. D.all popular science-fiction books are entered into the system by data entry staff

Câu hỏi 24: Câu nào sau đây mô tả tốt nhất về dữ liệu đầu vào của một phần mềm?

Lựa chọn một trong những đáp án sau:

a. Mỗi nhân viên được thuê bởi cửa hàng sách làm việc toàn thời gian.

b. Một khách hàng yêu cầu nhân viên bán hàng tìm kiếm các tài liệu về một cuốn sách.

c. Một khách hàng có tên, ngày sinh và địa chỉ.

d. Tất cả các cuốn sách khoa học viễn tưởng phổ biến được nhập vào hệ thống bởi nhân viên nhập liệu.

Đáp án đúng là c. Một khách hàng có tên, ngày sinh và địa chỉ.

Câu này mô tả về dữ liệu đầu vào của một phần mềm, cụ thể là thông tin về một khách hàng bao gồm tên, ngày sinh và địa chỉ. Dữ liệu đầu vào là các thông tin, giá trị hoặc tập hợp các dữ liệu được cung cấp cho phần mềm để xử lý và thực hiện các chức năng cần thiết. Trong trường hợp này, dữ liệu đầu vào là thông tin cá nhân của khách hàng, có thể được sử dụng để tìm kiếm và xác định các yêu cầu của khách hàng.

Các đáp án còn lại không phù hợp với mô tả về dữ liệu đầu vào của một phần mềm:

- a. Mỗi nhân viên được thuê bởi cửa hàng sách làm việc toàn thời gian: Đây không phải là mô tả về dữ liệu đầu vào, mà là mô tả về nhân viên làm việc tại cửa hàng sách.

- b. Một khách hàng yêu cầu nhân viên bán hàng tìm kiếm các tài liệu về một cuốn sách: Đây là mô tả về một hoạt động cụ thể, không phải là mô tả về dữ liệu đầu vào.

- d. Tất cả các cuốn sách khoa học viễn tưởng phổ biến được nhập vào hệ thống bởi nhân viên nhập liệu: Đây là mô tả về một quá trình nhập liệu, không phải là mô tả về dữ liệu đầu vào.

Vì vậy, đáp án chính xác là c. Một khách hàng có tên, ngày sinh và địa chỉ.

Question 25: Which of the following describes the types of user interaction that could be captured in a use case?

Select one:

a. A.basic and advanced

b. B.normal and advanced

c. C.normal and extended

d. D.regular and erroneous

Câu hỏi 25: Các lựa chọn sau đây mô tả các loại tương tác người dùng có thể được ghi lại trong một trường hợp sử dụng (use case) như thế nào?

Chọn một:

a. A. Cơ bản và nâng cao (basic and advanced)

b. B. Bình thường và nâng cao (normal and advanced)

c. C. Bình thường và mở rộng (normal and extended)

d. D. Thông thường và lỗi (regular and erroneous)

Đáp án đúng là c. Bình thường và mở rộng (normal and extended).

Trong một trường hợp sử dụng, ta mô tả cách mà người dùng tương tác với hệ thống để đạt được mục tiêu cụ thể. Các loại tương tác người dùng có thể được ghi lại trong một trường hợp sử dụng bao gồm các tương tác bình thường (normal) và mở rộng (extended). Tương tác bình thường mô tả các hành động thông thường và tuần tự mà người dùng thực hiện để hoàn thành nhiệm vụ. Tương tác mở rộng mô tả các tương tác bổ sung hoặc biến thể mà có thể xảy ra trong một tình huống cụ thể.

Các đáp án khác không phù hợp với việc mô tả các loại tương tác người dùng trong một trường hợp sử dụng:

- a. Cơ bản và nâng cao: Đây không phải là mô tả chính xác về các loại tương tác người dùng trong trường hợp sử dụng.

- b. Bình thường và nâng cao: Tương tác "nâng cao" không phản ánh đầy đủ ý nghĩa của các tương tác người dùng có thể xảy ra trong một trường hợp sử dụng.

- d. Thông thường và lỗi: "Lỗi" không phải là một loại tương tác người dùng, mà là một trạng thái hoặc kết quả không mong muốn có thể xảy ra trong quá trình tương tác người dùng.

Vì vậy, đáp án chính xác là c. Bình thường và mở rộng (normal and extended).

Question 26:Which of the following are parts of a use case description?

Select one:

a. A.stimuli, user, database, reply

b. B.actor, data, response, stimuli

c. C.user, constraint, reply, input

d. D.database, response, description, data

Câu hỏi 26: Các phần sau đây nằm trong một mô tả trường hợp sử dụng (use case) là gì?

Chọn một:

a. A. Tác động, người dùng, cơ sở dữ liệu, phản hồi (stimuli, user, database, reply)

b. B. Diễn viên, dữ liệu, phản hồi, tác động (actor, data, response, stimuli)

c. C. Người dùng, ràng buộc, phản hồi, đầu vào (user, constraint, reply, input)

d. D. Cơ sở dữ liệu, phản hồi, mô tả, dữ liệu (database, response, description, data)

Đáp án đúng là b. Diễn viên, dữ liệu, phản hồi, tác động (actor, data, response, stimuli).

Một mô tả trường hợp sử dụng (use case description) bao gồm các phần sau đây:

- Diễn viên (actor): Đại diện cho các thực thể ngoại vi hoặc hệ thống khác tương tác với hệ thống mục tiêu.

- Dữ liệu (data): Các dữ liệu đầu vào và đầu ra được sử dụng hoặc tạo ra bởi trường hợp sử dụng.

- Phản hồi (response): Phản hồi của hệ thống hoặc diễn viên sau khi thực hiện trường hợp sử dụng.

- Tác động (stimuli): Những sự kiện hoặc hành động tác động đến hệ thống hoặc diễn viên để kích hoạt trường hợp sử dụng.

Các đáp án khác không phù hợp với các phần của một mô tả trường hợp sử dụng:

- a. Tác động, người dùng, cơ sở dữ liệu, phản hồi: "Người dùng" và "cơ sở dữ liệu" không phải là các phần chính của mô tả trường hợp sử dụng.

- c. Người dùng, ràng buộc, phản hồi, đầu vào: "Ràng buộc" và "đầu vào" không phải là các phần chính của mô tả trường hợp sử dụng.

- d. Cơ sở dữ liệu, phản hồi, mô tả, dữ liệu: "Mô tả" không phải là một phần chính của mô tả trường hợp sử dụng.

Vì vậy, đáp án chính xác là b. Diễn viên, dữ liệu, phản hồi, tác động (actor, data, response, stimuli).

Question 27: What are the first two design iterations called?

Select one:

a. A.beginning and preparation

b. B.preparation and starting

c. C.important and key abstractions

d. D.initial and top-level abstractions

Câu hỏi 27: Hai lần lặp thiết kế đầu tiên được gọi là gì?

Chọn một:

a. A. Bắt đầu và chuẩn bị (beginning and preparation)

b. B. Chuẩn bị và bắt đầu (preparation and starting)

c. C. Trừu tượng quan trọng và chủ chốt (important and key abstractions)

d. D. Trừu tượng ban đầu và cấp độ cao nhất (initial and top-level abstractions)

Đáp án đúng là d. Trừu tượng ban đầu và cấp độ cao nhất (initial and top-level abstractions).

Trong quá trình thiết kế phần mềm, các lần lặp đầu tiên được thực hiện để xác định và xây dựng các trừu tượng ban đầu và cấp độ cao nhất cho hệ thống. Đây là các trừu tượng quan trọng và chủ chốt trong thiết kế và thường được sử dụng làm cơ sở cho các lần lặp thiết kế tiếp theo.

Các đáp án khác không phù hợp với quá trình thiết kế ban đầu:

- a. Bắt đầu và chuẩn bị: Không mô tả đúng các giai đoạn thiết kế ban đầu.

- b. Chuẩn bị và bắt đầu: Không mô tả đúng các giai đoạn thiết kế ban đầu.

- c. Trừu tượng quan trọng và chủ chốt: Mô tả các trừu tượng quan trọng và chủ chốt, nhưng không phản ánh đầy đủ ý nghĩa của các lần lặp thiết kế đầu tiên.

Vì vậy, đáp án chính xác là d. Trừu tượng ban đầu và cấp độ cao nhất (initial and top-level abstractions).

Question 28: The spiral process model

Select one:

a. A.organise the phases in iterations with feedbacks between phases

b. B.organise the phases in a never-ending spiral shape

c. C.organise the phases in two pairs: two upper and two lower phases

d. D.organise the phases in four separate quarters

Question 29: Why is it difficult to build a good software?

Select one:

a. A.software has many interactions within it and with other systems

b. B.a good software requires a lot of natural resources to develop

c. C.programmers do not want to use software engineering standards

d. D.there is a lack of software engineering standards

e. E.there is a lack of good programmers in the industry

Câu hỏi 29: Tại sao việc xây dựng một phần mềm tốt khó khăn?

Chọn một:

a. A. Phần mềm có nhiều tương tác bên trong và với các hệ thống khác.

b. B. Một phần mềm tốt yêu cầu nhiều tài nguyên thiên nhiên để phát triển.

c. C. Các lập trình viên không muốn sử dụng các tiêu chuẩn kỹ thuật phần mềm.

d. D. Thiếu các tiêu chuẩn kỹ thuật phần mềm.

e. E. Thiếu các lập trình viên giỏi trong ngành công nghiệp.

Đáp án chính xác là a. Phần mềm có nhiều tương tác bên trong và với các hệ thống khác.

Việc xây dựng một phần mềm tốt đôi khi khó khăn do phần mềm có nhiều tương tác phức tạp bên trong nó và với các hệ thống khác. Phần mềm thường phải xử lý nhiều yêu cầu, luồng dữ liệu và tương tác với các thành phần khác, điều này làm tăng độ phức tạp và khó khăn trong việc xây dựng và duy trì một phần mềm tốt.

Các đáp án khác không phù hợp với lý do khó khăn xây dựng một phần mềm tốt:

- b. Một phần mềm tốt yêu cầu nhiều tài nguyên thiên nhiên để phát triển: Khó khăn trong việc xây dựng một phần mềm tốt không liên quan đến tài nguyên thiên nhiên.

- c. Các lập trình viên không muốn sử dụng các tiêu chuẩn kỹ thuật phần mềm: Một số lập trình viên có thể không muốn sử dụng tiêu chuẩn kỹ thuật phần mềm, nhưng điều này không phải lý do chính khiến việc xây dựng phần mềm khó khăn.

- d. Thiếu các tiêu chuẩn kỹ thuật phần mềm: Thiếu tiêu chuẩn kỹ thuật phần mềm có thể gây khó khăn trong việc phát triển phần mềm, nhưng không phải lý do chính khiến việc xây dựng phần mềm khó khăn.

- e. Thiếu các lập trình viên giỏi trong ngành công nghiệp: Dù có thiếu lập trình viên giỏi trong

Question 30: Which of the following lists the software design evaluation criteria?

Select one:

a. A.correctness, performance

b. B.simplicity, performance

c. C.effectiveness, correctness

d. D.simplicity, generality

e. E.performance, effectiveness

Question 31: A top-down implementation plan has the following features

Select one:

a. A.late detection of errors and decreased re-use in testing

b. B.late detection of errors and increased re-use in testing

c. C.early detection of errors and increased re-use in testing

d. D.early detection of errors and decreased re-use in testing

Question 32: A top-down implementation plan is

Select one:

a. A.coding a module before those that use it

b. B.implementing more important modules before those that are less important

c. C.coding a module before those that it uses

d. D.implementing the header before the body of a module

Question 33: A bottom-up implementation plan has the following features

Select one:

a. A.early protypes of the smaller system and more up-front resources

b. B.early protypes of the system and more up-front resources

c. C.early protypes of the system and less up-front resources

d. D.early protypes of smaller systems and less up-front resources

Question 34: Which of the following best defines an implementation plan?

Select one:

a. A.a plan for who will implement each abstraction and for how long

b. B.a plan for how to implement the software from the top down

c. C.a set of implementation iterations and the order in which they are performed

d. D.a plan for implementing the software from the bottom up

Question 35: Which of the following lists the types of non-functional requirements?

Select one:

a. A.accuracy, modifier

b. B.modifier, performance

c. C.performance, reusability

d. D.reusability, accuracy

Question 36: Why is it necessary to construct models of a software during requirement analysis?

Select one:

a. A.to define a suitable design from the requirements

b. B.to understand and specify the requirements in detail

c. C.to understand how the software operates

d. D.to visualize the requirements to see what the software would look like

Question 37: Which of the following describes the types of user interaction that could be captured in a use case?

Select one:

a. A.normal and extended

b. B.regular and erroneous

c. C.basic and advanced

d. D.normal and advanced

Question 38: Which of the following best describes the steps that are performed at each design iteration?

Select one:

a. A.select an abstraction to create a sequence diagram for it

b. B.select a requirement specification and identify the helper abstractions

c. C.select an abstraction to create a class diagram for it

d. D.select suitable abstractions to focus on and identify the helper abstractions

Question 39: Which of the following best describes the object-oriented design technique named "decomposition by abstraction"?

Select one:

a. A.decompose each software function and create/use abstractions as required to perform the functions

b. B.decompose a system into functions and to create/use procedures to perform them

c. C.decompose a system into classes and to create/use methods of these classes

d. D.iteratively decompose a system into smaller systems

Question 40: The primary purpose of the design note is to:

Select one:

a. A.record confidential design information that are not included in other documents

b. B.systematically analyse the requirements in order to make better design decisions

c. C.draw a picture of the system and briefly describes it

Question 41: What is the primary purpose of a use case diagram?

Select one:

a. A.to show a diagram of how to use the system

b. B.to show the user interactions in different scenarios

c. C.to show how the software operates in different situations

d. D.to describe the details of all the use cases of the software

Question 42: Which of the following is a constraint on an attribute?

Select one:

a. A.a word may appear in one or more documents

b. B.the frequency of each word in a document is the number of its

occurrences

c. C.a document matches a query if it contains all of the query's keywords

d. D.each document may have one or more words

Question 43: The requirement specification language

Select one:

a. A.is a programming language that the developer uses

b. B.is a natural language that the end user communicates

c. C.is adapted from the design specification language

d. D.is a mathematical language that makes precise the requirements

Question 44: The requirement specification of a software function

Select one:

a. A.defines the function with all the required information

b. B.describes the detailed input and output of the function

c. C.makes precise the pre- and post-conditions of the function

d. D.does not include the return type nor the pre-conditions

Question 45: Which of the following lists the basic modeling constructs of an UML class diagram?

Select one:

a. A.relationship, entity

b. B.class, association

c. C.class, entity

d. D.entity, association

e. E.class, relationship

Question 46:Which of the following is the goal of software design?

Select one:

a. A.to construct the precise specification for how the software will perform

b. B.to construct a detailed drawing of the software

c. C.to write a software specification for the designer to understand

d. D.to write a software specification for the users to understand

Question 47: Design specification of an object procedure consists of?

Select one:

a. A.header behaviour and description.

b. B.header and behaviour description.

c. C.header description and behaviour code.

d. D.header code and behaviour header.

Question 48: Which annotation is used to design the object representation?

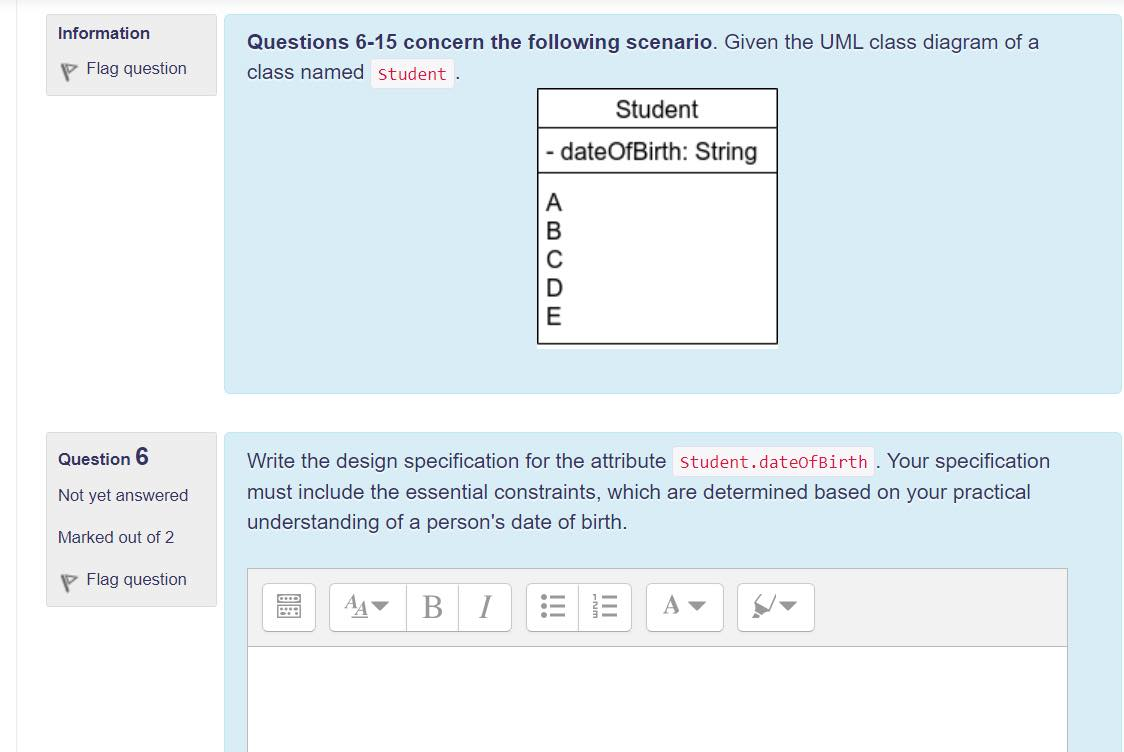
Select one:

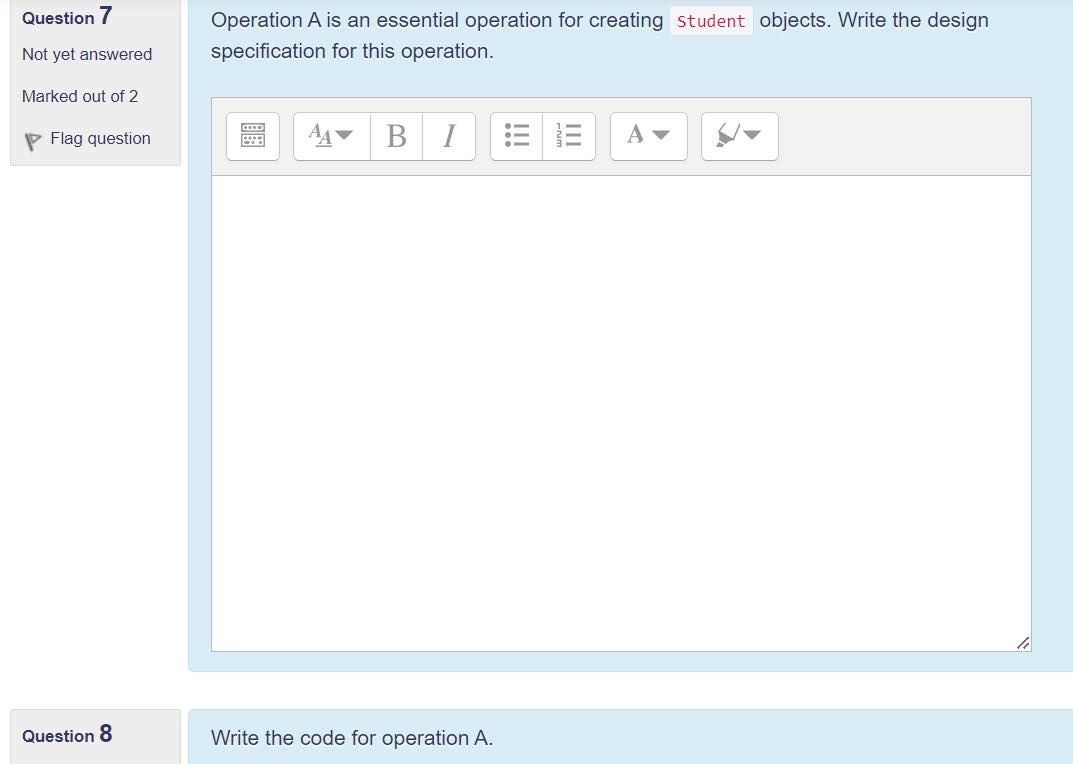
a. A.DomainObject

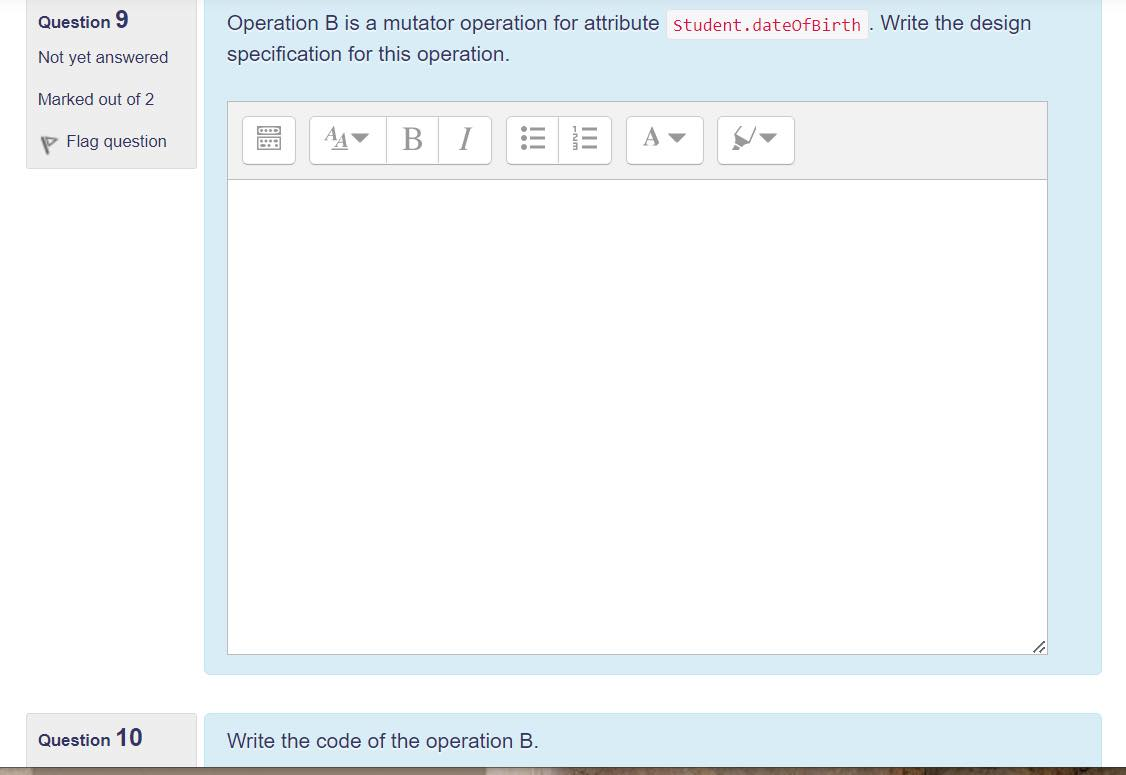
b. B.DomConstraint

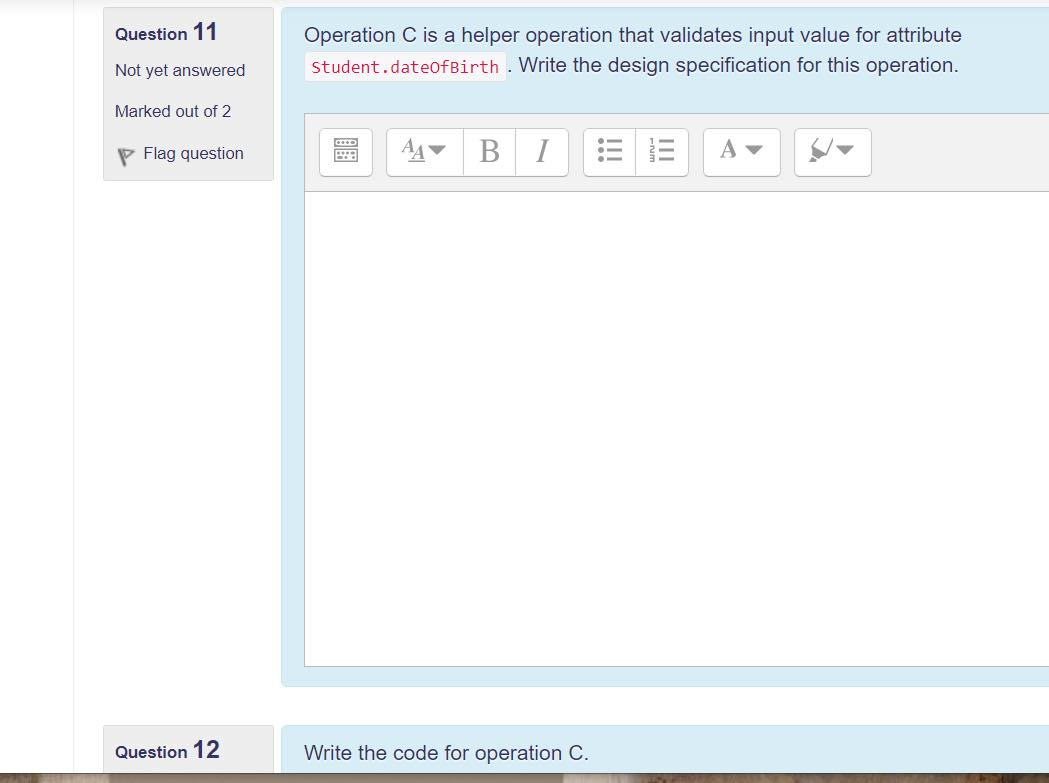
c. C.ObjectConstraint

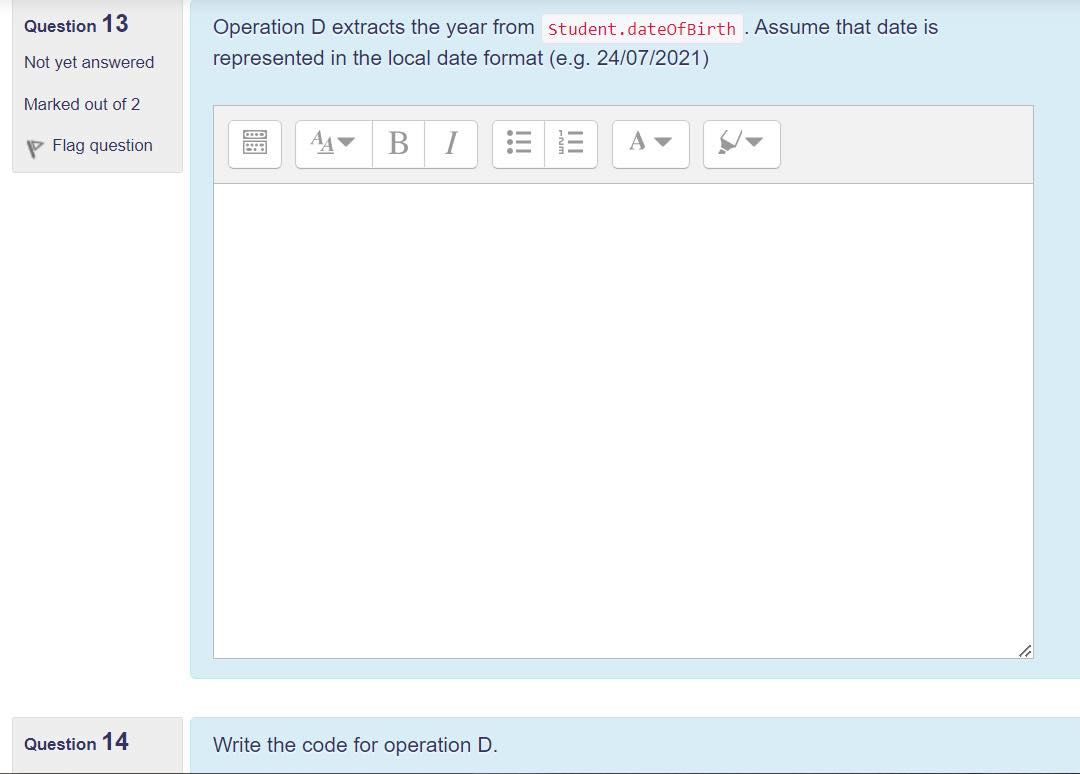
d. D.DomainConstraint

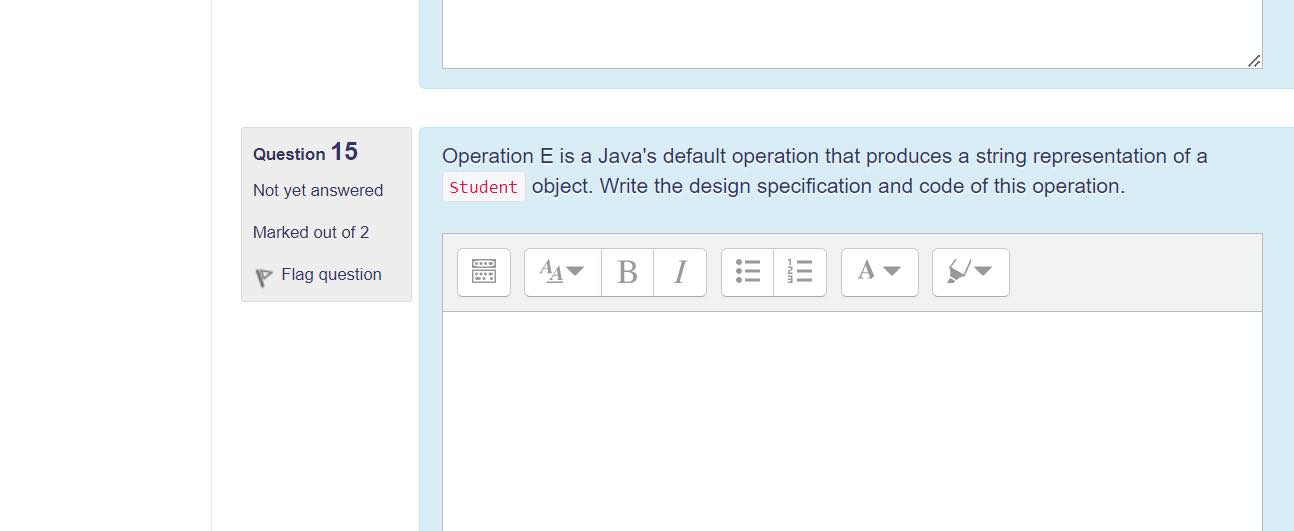


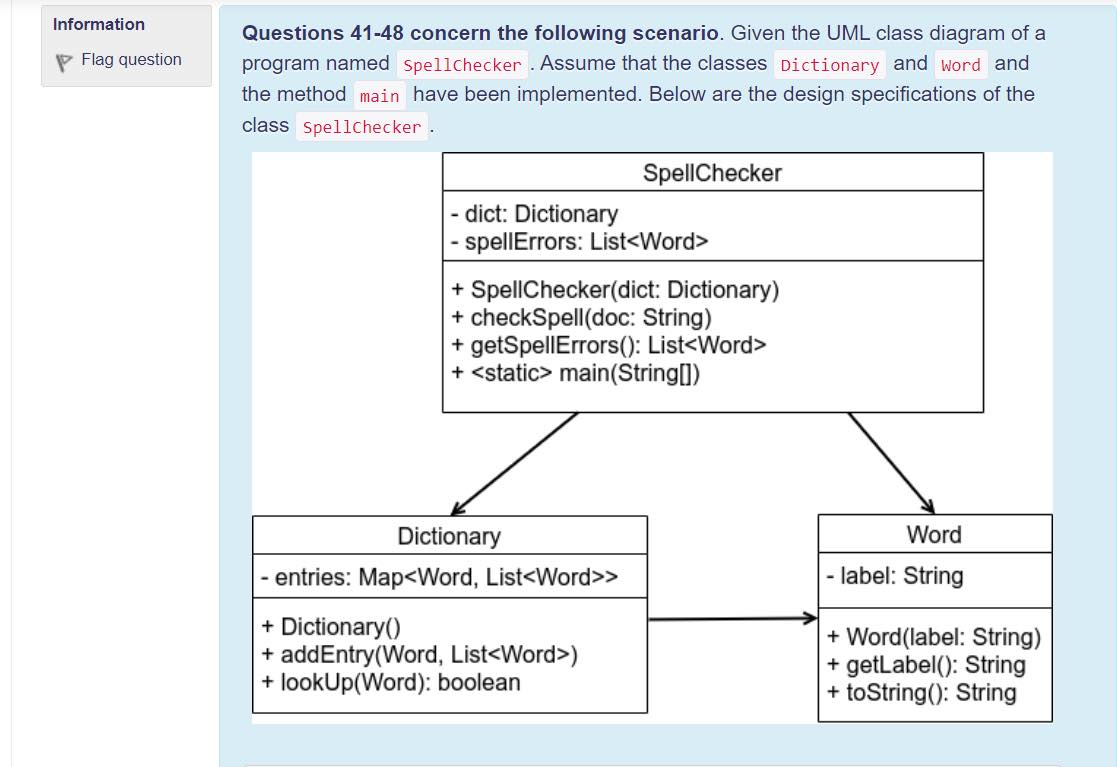


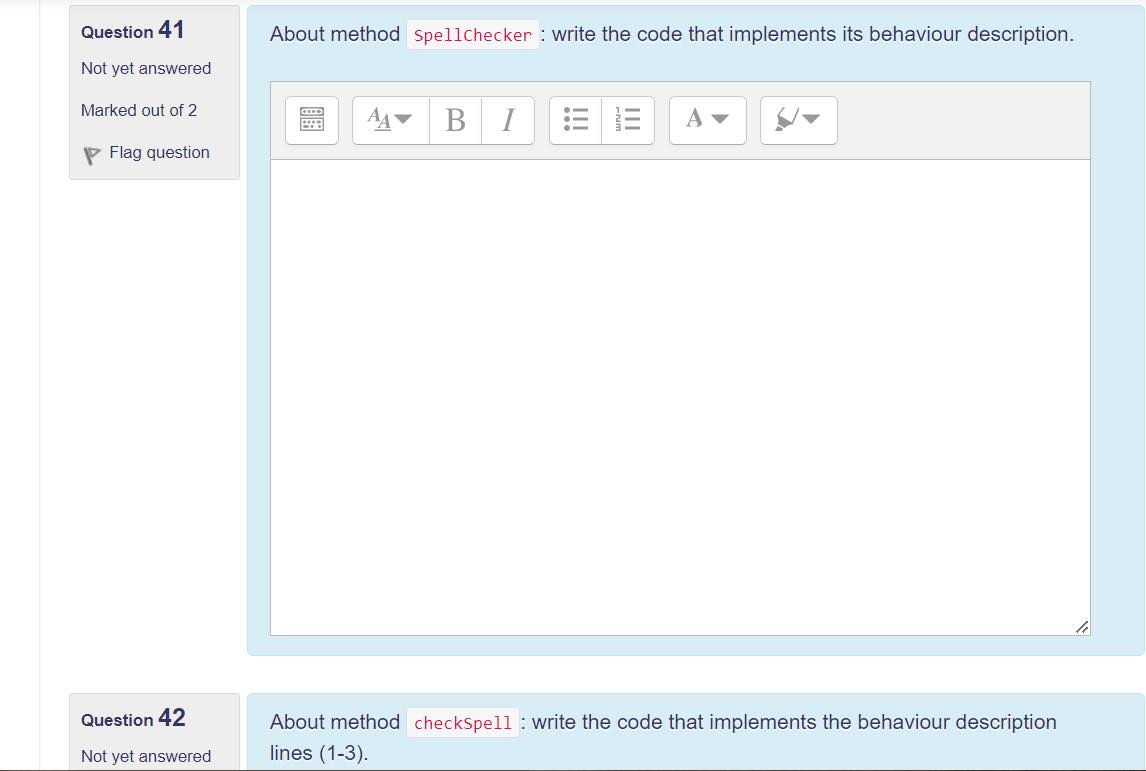


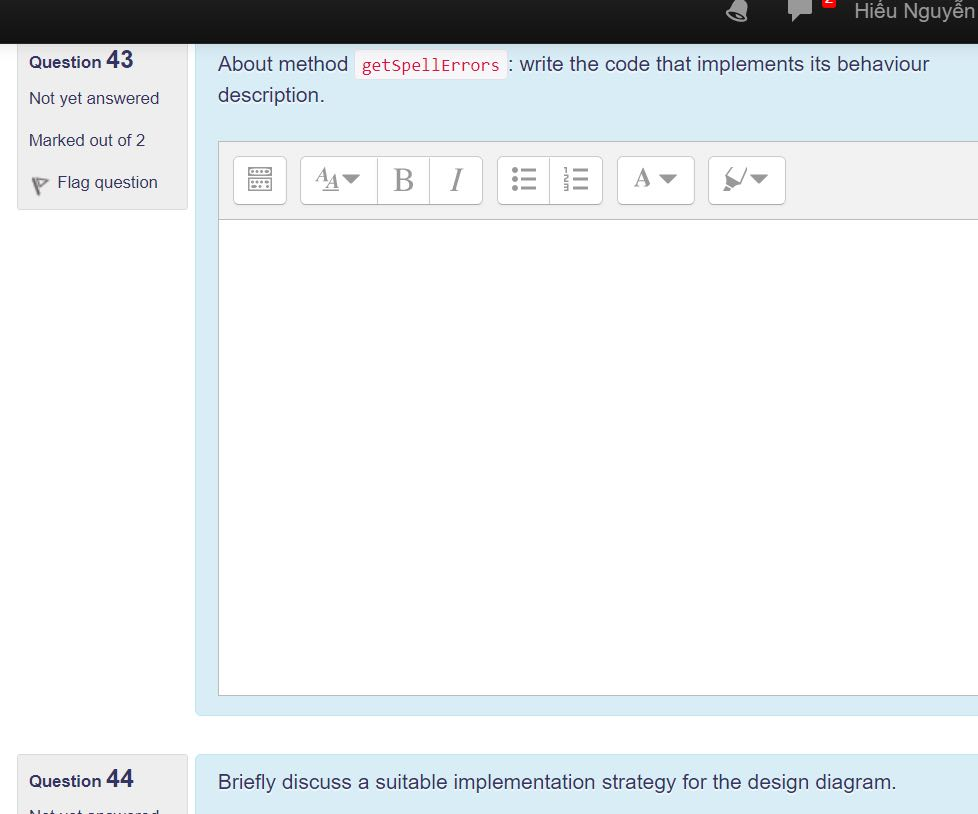


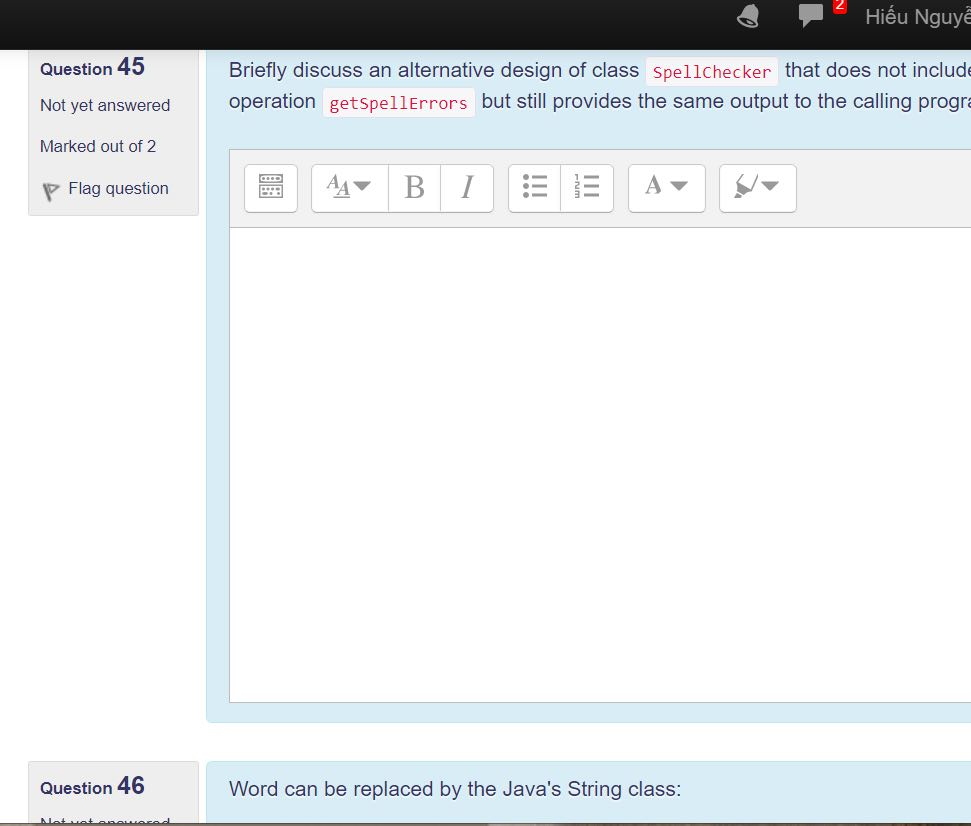




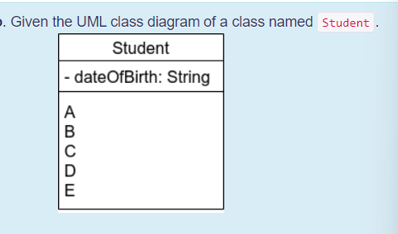








Q6: Given the UML class diagram of a class named Student:



Q6: Write the design specification for the attribute Student.dateOfBirth. Your specification must be include the essential constraints, which are determined based on your practical understanding of a person’s date of birth.

Design specification for the attribute Student.dateOfBirth with essential constraints:

Name: dateOfBirth

Type: string

Visibility: Private (-)

Constraints:

Must be a valid date in the format of "YYYY-MM-DD"

Must be a date before the current date

Must be at least 18 years old

Description: This attribute represents the date of birth of a student. It is implemented as a private string variable within the Student class.

Usage: The attribute can be accessed and modified only within the Student class methods. It can be used to retrieve and update the date of birth of a student object, subject to the above constraints.

Default Value: None specified

Example: "1999-03-15" (if the student was born on March 15, 1999)

Related Attributes/Methods: None specified

Dependencies: None specified

Note: This design specification includes essential constraints based on common practical understanding of a person's date of birth. The first constraint ensures that the date is in the correct format. The second constraint ensures that the date is not in the future. The third constraint ensures that the student is at least 18 years old, which is often the minimum age for many activities (e.g. voting, driving, legal contracts).

Q7: Operation A is an essential operation for creating Student objects. Write the design specification for this operation

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

Design specification for essential operation A for creating Student objects:

* Name: A
* Type: Constructor
* Visibility: Public (+)
* Parameters:
  + name: a string representing the name of the student (required)
  + id: a string representing the ID of the student (required)
  + dateOfBirth: a string representing the date of birth of the student (required)
* Description: This constructor is used to create a new Student object with the specified name, ID, and date of birth.
* Usage: The constructor can be called externally to create a new Student object. Once created, the object can be used to access and modify student information through its attributes and methods.
* Pre-conditions: The name, ID, and dateOfBirth parameters must all be provided and must meet the constraints specified for each attribute.
* Post-conditions: A new Student object is created with the specified name, ID, and date of birth, and its attributes are initialized accordingly. The new object is returned as the output of the constructor.
* Example:
* java
* Copy code
* Student newStudent = new Student("John Doe", "1234", "1999-03-15");
* Related Attributes/Methods:
  + name: a string representing the name of the student
  + id: a string representing the ID of the student
  + dateOfBirth: a string representing the date of birth of the student
  + Other attributes and methods of the Student class as needed
* Dependencies: None specified

Note: This design specification assumes that the Student class is implemented in a programming language with object-oriented features, and that the constructor is defined accordingly. The constructor takes three required parameters corresponding to the name, ID, and date of birth of the student, and initializes the corresponding attributes of the new Student object. The pre-conditions ensure that the provided parameters are valid according to the constraints specified for each attribute. The post-conditions ensure that a new Student object is created and returned as the output of the constructor, with its attributes initialized accordingly

Q8: Write the code for operation A:

public class Student {

private String name;

private String id;

private String dateOfBirth;

public Student(String name, String id, String dateOfBirth) {

this.name = name;

this.id = id;

this.dateOfBirth = dateOfBirth;

}

// getters and setters for name, id, and dateOfBirth attributes

// ...

// other methods for the Student class as needed

// ...

}

Q9: Operation B is a mutator for attribute Student.dateOfBirth. Write the design specification for this operation.

Operation B: setBirthDate(date: String) : void

Description: Sets the date of birth for the current Student object to the specified value.

Parameters:

* date: a String value representing the date of birth of the student in the format YYYY-MM-DD.

Preconditions:

* The Student object is not null.
* The date parameter is not null or empty.
* The date parameter represents a valid date of birth in the format YYYY-MM-DD.

Postconditions:

* The dateOfBirth attribute of the Student object is set to the specified date value.

Exceptions:

* IllegalArgumentException is thrown if the date parameter is null or empty, or does not represent a valid date in the format YYYY-MM-DD.

Q10: Write the code of the operation B.

public class Student {

// other attributes and methods for the Student class as needed

// ...

public void setBirthDate(String date) {

if (date == null || date.isEmpty()) {

throw new IllegalArgumentException("Date of birth cannot be null or empty");

}

// validate the date format and ensure it represents a valid date

// ...

this.dateOfBirth = date;

}

}

Q11: Operation C is a helper operation that validates input value for attribute Student.dateOfBirth. Write the design specification for this operation.

Design Specification for Operation C:

Operation Name: validateDateOfBirth

Input Parameter: dateOfBirth (string) - the date of birth to be validated

Output: boolean - indicating whether the input date of birth is valid or not

Description:

The helper operation validateDateOfBirth is used to validate the input value for the attribute Student.dateOfBirth. It takes a single parameter dateOfBirth which is a string representing the date of birth in the format DD/MM/YYYY. The operation returns a boolean value indicating whether the input date of birth is valid or not.

Constraints:

* The input date of birth must be a non-empty string.
* The input date of birth must be in the format DD/MM/YYYY.
* The input date of birth must be a valid date according to the Gregorian calendar.
* The input date of birth must be a date in the past (i.e., before the current date).

Algorithm:

1. Check if the input dateOfBirth is a non-empty string.
2. If the input is empty, return false.
3. If the input is not empty, parse the input string into day, month, and year components.
4. Check if the day, month, and year components are valid according to the Gregorian calendar.
5. If the components are invalid, return false.
6. Check if the input date is in the past.
7. If the date is in the future or today, return false.
8. If all checks pass, return true.

Q12: Write code for operation C

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class Student {

private String dateOfBirth;

// Constructor for creating a new Student object with a given date of birth

public Student(String dateOfBirth) {

if (validateDateOfBirth(dateOfBirth)) {

this.dateOfBirth = dateOfBirth;

} else {

throw new IllegalArgumentException("Invalid date of birth");

}

}

// Getter for the dateOfBirth attribute

public String getDateOfBirth() {

return dateOfBirth;

}

// Setter for the dateOfBirth attribute

public void setDateOfBirth(String dateOfBirth) {

if (validateDateOfBirth(dateOfBirth)) {

this.dateOfBirth = dateOfBirth;

} else {

throw new IllegalArgumentException("Invalid date of birth");

}

}

// Helper method for validating a date of birth string

public static boolean validateDateOfBirth(String dateOfBirth) {

if (dateOfBirth == null || dateOfBirth.isEmpty()) {

return false;

}

SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");

format.setLenient(false);

try {

Date dob = format.parse(dateOfBirth);

Date currentDate = new Date();

if (dob.after(currentDate) || dob.equals(currentDate)) {

return false;

}

} catch (ParseException e) {

return false;

}

return true;

}

}

Q13: Operation D extracts the year from Student.dateOfBirth. Assume that date is represented in the local date format (e.g 24/07/2021)

Here's a possible design specification for the operation D:

Operation D: extractYear

Description:

This operation extracts the year value from the date of birth attribute of a student object. The date of birth is represented in the local date format "dd/MM/yyyy".

Input:

* dateOfBirth (string): The date of birth of the student object in the format "dd/MM/yyyy".

Output:

* year (integer): The year value extracted from the input date string.

Behavior:

1. Parse the input dateOfBirth string using a SimpleDateFormat object with the pattern "dd/MM/yyyy".
2. Create a Calendar object and set its time to the parsed date.
3. Extract the year value using the Calendar.get(Calendar.YEAR) method.
4. Return the extracted year value.

Exceptions:

* ParseException: If the input dateOfBirth string is not in the expected format.

Preconditions:

* The input dateOfBirth string must be in the format "dd/MM/yyyy".

Postconditions:

* None.

Constraints:

* The input dateOfBirth string must represent a valid date in the local date format "dd/MM/yyyy".
* The extracted year value must be a valid integer value

Q14: Code for D:

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

public class Student {

// other attributes and operations

/\*\*

\* Extracts the year value from the date of birth attribute of this student object.

\*

\* @param dateOfBirth The date of birth of the student object in the format "dd/MM/yyyy".

\* @return The year value extracted from the input date string.

\* @throws ParseException If the input dateOfBirth string is not in the expected format.

\*/

public int extractYear(String dateOfBirth) throws ParseException {

SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");

Date parsedDate = dateFormat.parse(dateOfBirth);

Calendar calendar = Calendar.getInstance();

calendar.setTime(parsedDate);

return calendar.get(Calendar.YEAR);

}

// other attributes and operations

}

Q15: Operation E is a Java’s default operation that produces a string representation of a Student object. Write the design specification and code of this operation

Here's the design specification for operation E, which is to override the toString method of the Student class to produce a string representation of a Student object:

Design Specification:

Operation: E - toString

Description: Produces a string representation of a Student object.

Inputs: None.

Outputs: A string representation of the Student object.

Preconditions: None.

Postconditions: None.

Exceptions: None.

Design rationale: The toString method is a default method in Java that returns a string representation of an object. In this case, we want to override it in the Student class to produce a string that includes the values of all of its attributes.

public class Student {

private String dateOfBirth;

// other attributes

// other operations

/\*\*

\* Returns a string representation of this student object.

\*

\* @return A string representation of this student object.

\*/

@Override

public String toString() {

return "Student{" +

"dateOfBirth='" + dateOfBirth + '\'' +

// include other attributes in the string representation

'}';

}

}

Question : Given the UML class diagram