Análise e Projeto OO

TRT-23 (FCC 2011)

41. Sobre os conceitos de orientação a objetos, considere:

I. Classe encapsula dados para descrever o conteúdo de alguma entidade do mundo real.

II. Objetos são instâncias de uma classe que herdam os atributos e as operações da classe.

III. Superclasse é uma especialização de um conjunto de classes relacionadas a ela.

IV. Operações, métodos ou serviços fornecem representações dos comportamentos de uma classe.

Está completo e correto o que consta em

a) I, II, III e IV.
b) I, II e IV, apenas.
c) II, III e IV, apenas.
d) I e II, apenas.
e) II e IV, apenas.

43. No projeto de arquitetura modelo-visão-controle (MVC), o controlador

a) renderiza a interface de usuário a partir da visão, o modelo encapsula funcionalidades e objetos de conteúdo e a visão processa e responde a eventos e invoca alterações ao controlador.

b) encapsula funcionalidades e objetos de conteúdo, o modelo processa e responde a eventos e invoca alterações ao controlador e a visão renderiza a interface de usuário a partir do modelo.

c) encapsula funcionalidades e objetos de conteúdo, o modelo renderiza a interface de usuário a partir da visão e a visão processa e responde a eventos e invoca alterações ao controlador.

d) processa e responde a eventos e invoca alterações ao modelo, o modelo encapsula funcionalidades e objetos de conteúdo e a visão renderiza a interface de usuário a partir do modelo.

e) processa e responde a eventos e invoca alterações ao modelo, o modelo renderiza a interface de usuário a partir da visão e a visão encapsula funcionalidades e objetos de conteúdo.

NOSSA CAIXA DESENVOLVIMENTO (FCC 2011)

45. Em relação a projetos orientados a objetos, considere:

I. É uma estratégia de projeto em que o projetista pensa em termos de coisas em vez de funções.

II. A funcionalidade do sistema é expressa em termos de serviços oferecidos pelos objetos.

III. Objetos se comunicam por passagem de mensagem, eliminando áreas de dados compartilhados.

IV. O objetivo é implementar os requisitos, a partir do desenvolvimento de um modelo orientado a objeto de um sistema de software.

Está correto o que se afirma em

a) I e III, apenas.
b) II e IV, apenas.
c) I, III e IV, apenas.
d) II, III e IV, apenas.
e) I, II, III e IV.

TRT-14 (FCC 2011)

43. A classe Veiculo contém alguns atributos de interesse da classe Aeronave. Todavia, as aeronaves também demonstram interesse em captar atributos e também operações da classe Elemento Turbinado. O enunciado enfatiza o conceito OO de

a) polimorfismo.

b) herança múltipla.

c) dependência funcional.

d) realização.

e) encapsulamento.

TRE/AP (FCC 2011)

25. Considere:

I. A classe Veículo possui as subclasses Carro e Trem.

II. Os objetos das subclasses Carro e Trem herdam a operação acelerar.

III. A aceleração do Carro é no pedal e, do Trem, é na manivela.

I, II e III mostram um conceito OO aplicável à implementação da aceleração, que é

a) encapsulamento.

b) derivação.

c) polimorfismo.

d) herança múltipla.

e) estereotipagem.

55. Sobre orientação a objetos, é correto afirmar:

a) Uma classe é o projeto do objeto. Ela informa à máquina virtual como criar um objeto de um tipo específico. Cada objeto criado a partir da classe terá os mesmos valores para as variáveis de instância da classe.

b) Um relacionamento de herança significa que a superclasse herdará as variáveis de instância e métodos da subclasse.

c) Uma interface é uma classe 100% abstrata, ou seja, uma classe que não pode ser instanciada.

d) Os objetos têm seu estado definido pelos métodos e seu comportamento definido nas variáveis de instância.

e) A principal regra prática do encapsulamento é marcar as variáveis de instância como públicas e fornecer métodos de captura e configuração privados.

TRE/SP (FCC 2012)

41. Nos conceitos de orientação a objetos, ..I... é uma estrutura composta por ...II..;. que descrevem suas propriedades e também por ...III.... que moldam seu comportamento. ....IV.... são ....V.... dessa estrutura e só existem em tempo de execução.

Para completar corretamente o texto as lacunas devem ser preenchidas, respectivamente, por

a) objeto, métodos, assinaturas, Classes, cópias.

b) polimorfismo, funções, métodos, Herança, cópias.

c) classe, atributos, operações, Objetos, instâncias.

d) multiplicidade, símbolos, números, Classes, herdeiros.

e) domínio, diagramas, casos de caso, Diagramas de classe, exemplos

TRE/CE (FCC 2012)

41. Sobre orientação a objetos, é INCORRETO afirmar:

a) os conceitos de generalização e especialização da orientação a objetos estão diretamente associados ao conceito de herança.

b) um objeto pode existir mesmo que não exista nenhum evento a ele associado.

c) um construtor visa inicializar os atributos e pode ser executado automaticamente sempre que um novo objeto é criado.

d) polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma assinatura e mesmo comportamento.

e) uma classe define o comportamento dos objetos através de seus métodos, e quais estados ele é capaz de manter através de seus atributos.

TJ/PE (FCC 2012)

53. Sobre a arquitetura cliente-servidor em camadas é correto afirmar:

a) Na camada de dados da arquitetura em três camadas devem ser representados os componentes que cuidam da lógica de negócios (business logic).

b) Sistemas que usam a arquitetura, cliente-servidor em duas camadas geralmente possuem problemas de falta de escalabilidade, dificuldade de manutenção e dificuldade de acessar fontes heterogêneas.

c) A arquitetura centralizada foi dominante até a década de 90 como arquitetura corporativa e disponibilizava uma interface amigável.

d) Na arquitetura cliente-servidor em duas camadas, a camada cliente trata da Interface do Usuário, enquanto a camada servidor trata exclusivamente da lógica de negócio.

e) A arquitetura em três camadas permite representar os componentes da aplicação nas camadas de negócio, aplicação e dados.

54. Com relação à arquitetura MVC, considere:

I. O MODEL representa os dados da empresa e as regras de negócio que governam o acesso e atualização destes dados.

II. O VIEW acessa os dados da empresa através do MODEL e especifica como esses dados devem ser apresentados. É de responsabilidade do VIEW manter a consistência em sua apresentação, quando o MODEL é alterado.

III. O CONTROLLER traduz as interações do VIEW em ações a serem executadas pelo MODEL. Com base na interação do usuário e no resultado das ações do MODEL, o CONTROLLER responde selecionando uma VIEW adequada.

IV. Permite uma única VIEW para compartilhar o mesmo modelo de dados corporativos em um fluxo de comunicação sequencial.

Está correto o que se afirma em

a) I, II, III e IV.
b) I, II e III, apenas.
c) II e III, apenas.
d) II, III e IV, apenas.
e) I e II, apenas.

BANESE (FCC 2012)

67. Em relação ao MVC, considere:

I. Como o modelo MVC gerencia múltiplos visualizadores usando o mesmo modelo é mais complicado manter, testar e atualizar sistemas múltiplos.

II. É simples incluir novos clientes apenas incluindo seus visualizadores e controles.

III. Torna a aplicação escalável.

IV. Não é possível ter desenvolvimento em paralelo para o modelo, visualizador e controle pois são interdependentes.

Está correto o que se afirma APENAS em

(A) I e II.
(B) II e III.
(C) II e IV.
(D) I, II e IV.
(E) I, III e IV.

METRO/SP (FCC 2012)

36. Com relação à característica dos componentes na engenharia de software baseada em componentes (CBSE – componente based software engineering), considere:

I. A padronização de componentes significa que um componente usado em um processo CBSE precisa obedecer a um modelo de componentes padrão. Esse modelo pode definir as

interfaces de componentes, metadados de componente, documentação, composição e implantação.

II. Um componente deve ser independente, deve ser possível compor e implantá-lo sem precisar usar outros componentes específicos.

III. Para um componente ser composto, todas as interações externas devem ter lugar por meio de interfaces privadas e restritas. Além disso, ele deve proporcionar acesso externo a informações sobre si próprio, como seus métodos e atributos.

Está correto o que consta em

(A) II e III, apenas.
(B) I, II e III.
(C) I e III, apenas.
(D) I e II, apenas.
(E) III, apenas.

43. A engenharia de software baseada em reuso é uma estratégia da engenharia em que o processo de desenvolvimento é orientado para o reuso de softwares existentes. Dentre os benefícios do reuso de software, é INCORRETO afirmar:

(A) Preencher uma biblioteca de componentes reusáveis e garantir que desenvolvedores de software possam utilizar essa biblioteca são ações não onerosas, pois processos de desenvolvimento não precisam ser adaptados para utilizar essa biblioteca.

(B) Devido ao custo do software existente já ser conhecido, o risco de processo é reduzido.

(C) Especialistas em aplicações podem desenvolver softwares reusáveis que encapsulem seu conhecimento, tornando seu uso mais eficaz.

(D) Muitas vezes os custos gerais de desenvolvimento não são tão importantes quanto entregar um sistema ao mercado o mais rápido possível. O reuso de um software pode acelerar a produção do sistema.

(E) Alguns padrões, como os de interface de usuário, podem ser implementados como um conjunto de componentes reusáveis. O uso de interfaces de usuário-padrão melhora a confiança, pois os usuários cometem menos erros quando são apresentados a interfaces familiares.

Testes e Qualidade de Software

INFRAERO (FCC 2011)

47. Analise os itens a seguir sobre as estratégias de teste para softwares convencionais:

I. Uma estratégia de teste que é escolhida normalmente por uma boa parte das equipes de software adota uma visão incremental do teste, começando com o teste de unidades individuais de programa, avançando para testes projetados a fim de facilitar a integração das unidades e culmina com testes que exercitam o sistema construído.

II. O teste de unidade focaliza o esforço de verificação na maior unidade de projeto do software: o componente ou módulo de software.

III. O teste de unidade enfoca a lógica interna de processamento e as estruturas de dados dentro dos limites de um componente.

IV. No teste de unidade, a interface do módulo é testada para garantir que a informação flui adequadamente para dentro e para fora da unidade de programa que está sendo testada.

Está correto o que consta em

a) I, II, III e IV.
b) I e II, apenas.
c) I, II e III, apenas.
d) II, III e IV, apenas.
e) I, III e IV, apenas.

TRT-4 (FCC 2011)

52. O teste de componentes compostos concentra-se, principalmente, em verificar se

a) a funcionalidade dos códigos do componente atendem aos requisitos de negócio.

b) o sistema está atingindo a carga projetada, quando processados todos os componentes em conjunto.

c) a interface de componente se comporta de acordo com sua especificação.

d) a versão dos componentes do sistema está de acordo com os requisitos não funcionais.

e) a saturação sistêmica aborta a execução de um ou mais componentes.

TRT-20 (FCC 2010)

42. No contexto da estratégia para o teste de um projeto, os estágios de teste desempenham um papel importante. O teste que é aplicado a componentes do modelo de implementação para verificar se os fluxos de controle e de dados estão cobertos e funcionam conforme o esperado, é o teste

a) do desenvolvedor.

b) independente.

c) de integração.

d) de sistema.

e) unitário.

BAHIAGÁS (FCC 2010)

56. Na direção dos tipos de teste focados pela engenharia de software, os testes de integração cuidam dos tópicos associados com os problemas de verificação.

a) da engenharia de sistemas.

b) do projeto do software.

c) dos códigos de programa.

d) dos requisitos funcionais.

e) dos requisitos não funcionais.

TRT-9 (FCC 2010)

32. O teste de sistema que força o software a falhar de diversos modos e verifica o retorno do processamento dentro de um tempo pré-estabelecido é um tipo de teste de

a) Integração.

b) Estresse.

c) Recuperação.

d) Desempenho.

e) Segurança.

TJ/RJ (FCC 2012)

53. No que se refere a testes de software, é correto afirmar que

(A) o teste de operação é a fase onde é testada a ergonomia da interface de uso do software.

(B) o teste da caixa preta (teste funcional), baseia-se em analisar os arquivos de log do sistema procurando por mensagens de funcionamento inconsistente.

(C) um teste bem sucedido é um teste que não encontra nenhum erro no software.

(D) o teste da caixa branca (teste estrutural), baseia-se em testar as estruturas do código fonte, como comandos condicionais e de repetição.

(E) um caso de teste é uma categoria de possíveis resultados na execução de testes

UML

TRT-19 (FCC 2011)

33. Na versão 2.0 da UML, costuma conter elementos tais como: ações, bifurcações, ramificações e fluxos. Trata-se do diagrama de

a) máquina de estados

b) implantação.

c) sequência.

d) atividades.

e) artefatos.

36. Considere: E = estruturais e C = comportamentais. Os diagramas de comunicação, pacotes, implantação e componentes são, respectivamente,

a) C; E; E; E.

b) C; C; E; E.

c) C; E; E; C.

d) E; C; C; C.

e) E; C; E; C.

INFRAERO (FCC 2011)

43. Quanto à classificação de tipo de bloco de construção, de acordo com Grady Booch et al, o vocabulário UML abrange

a) 1 tipo.

b) 2 tipos.

c) 3 tipos.

d) 4 tipos.

e) 5 tipos.

44. Considere:

I. estruturais,

II. comportamentais,

III. de agrupamentos,

IV. anotacionais.

Quanto aos itens da UML que podem constituir blocos de construção básicos OO, está correto o que consta em

a) I e II, apenas.
b) I e III, apenas.
c) I, II e III, apenas.
d) II e IV, apenas.
e) I, II, III e IV.

45. Na notação UML, um nome entre ângulos (ex. <<nome>>), colocado acima do nome de outro elemento, é utilizado para a representação gráfica de

a) objeto.

b) função.

c) multiplicidade.

d) operação.

e) estereótipo.

46. Qualquer descendente do classificador é capaz de usar a característica; sua especificação é antecedida pelo símbolo #. A definição trata da visibilidade usada na notação UML, de nível

a) público.

b) privado.

c) pacote.

d) protegido.

e) dependente.

47. Como exemplo, a classe CarroImportado (em itálico) é escrita desta forma na UML para especificar que tal classe

a) é concreta.

b) pode não apresentar instâncias diretas.

c) herda características de mais de uma classe mãe.

d) herda características de apenas uma classe mãe.

e) se relaciona com ela mesma.

50. No diagrama de colaboração da UML (2.0), as linhas cheias que ligam os objetos e as setas de pontas cheias representam, respectivamente,

a) vínculo e dependência.

b) vínculo e mensagem.

c) mensagem e dependência.

d) mensagem e vínculo.

e) dependência e vínculo.

51. Ramificação sequencial, bifurcação concorrente e união concorrente são

a) meios de associação entre classes na UML ou entidades no MER.

b) componentes do Diagrama de Atividades da UML.

c) componentes do Diagrama Entidade-Relacionamento.

d) componentes do Diagrama de Sequência da UML.

e) possibilidades de associação entre atores e casos de uso da UML.

52. Para captar os requisitos funcionais de um sistema pode- se utilizar a UML. O diagrama mais adequado para essa finalidade é o diagrama de

a) casos de uso.

b) atividades.

c) colaboração.

d) classes.

e) comunicações.

60. Em UML,

a) diagramas de componentes são diagramas de comportamento com a função de visualizar um conjunto de componentes e as suas relações.

b) diagramas de classes têm como função visualizar um conjunto de objetos e as suas relações num determinado instante de tempo.

c) os requisitos identificam as funcionalidades pretendidas no sistema para cada perfil de usuário, com base nos diagramas de iteração.

d) diagramas de instalação são diagramas estruturais e têm a função de visualizar a configuração de um conjunto de nós de processamento e dos componentes em execução em cada nó.

e) uma classe abstrata é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações e relações.

MPE/PE (FCC 2012)

44. Atributos estáticos são atributos da classe em vez de serem atributos de uma instância da classe. Em UML um atributo estático é representado ao se utilizar em sua transcrição o:

a) modo sublinhado
b) símbolo #
c) símbolo /
d) modo itálico
e) símbolo ~

TRE/CE (FCC 2012)

34. A UML oferece mecanismos para a extensão da sintaxe e da semântica da linguagem. Considere:

I. Novos atributos de modelagem.
II. Novos elementos de modelagem.
III. Nova semântica de modelagem. I, II e III são, respectivamente,

a) estereótipos, diagramas e relacionamentos.
b) dependências, associações e generalizações.
c) valores atribuídos, estereótipos e restrições.
d) valores discretos, diagramas e protótipos.
e) relacionamentos, dependências e associações.

TJ/PE (FCC 2012)

51. É INCORRETO afirmar que em UML,

a) o diagrama de sequência pode ser usado para descrever como alguns objetos de um caso de uso colaboram em algum comportamento ao longo do tempo.

b) é comum no diagrama de caso de uso, a utilização de atores para representar entidades do mundo real e suas respectivas ações.

c) o diagrama de classes define as classes de que o sistema necessita e é a base para a construção dos diagramas de sequência e comunicação.

d) visão geral e atividades, casos de uso e de sequência são diagramas que descrevem uma visão dinâmica de um sistema.

e) o diagrama de sequência é tido como o menos funcional entre os diagramas da UML, em função de sua não integração com os sistemas de gerência.

TRT/AM (FCC 2012)

41. São técnicas úteis na fase de projeto de software orientado a objetos, EXCETO criar diagramas de

a) estados para classes com históricos de vida complexos.

b) distribuição para mostrar o layout físico do software.

c) pacote para mostrar a organização em larga escala do software.

d) atividade para descrever como as pessoas interagem com o sistema.

e) classes a partir de uma perspectiva de software para mostrar as classes presentes no software e seus relacionamentos.

42. Sobre o diagrama de classe da UML é correto afirmar:

a) Quando se utiliza diagramas de classe deve-se focar exclusivamente na estrutura do software e ignorar seu comportamento.

b) Dependência com classes não são adequadas para ilustrar um relacionamento transitório, como quando um objeto é passado para outro como parâmetro.

c) A UML permite representar dependência apenas de classes. Utilizam-se dependências quando se deseja mostrar que as mudanças em uma classe não afetam a outra classe.

d) Suporta quatro abreviações de visibilidade: + (público), - (privado), ~ (pacote) e # (protegido).

e) Uma classe abstrata é uma classe que pode ser instanciada diretamente. A maneira mais comum de identificar uma classe abstrata na UML é colocar o nome em negrito.

BANESE (FCC 2012)

65. A UML define Estereótipos Padrão que se aplicam às classes. Assim, aquele que especifica uma classe cujos atributos e operações pertencem ao escopo de estática é o

(A) extend.
(B) utility.
(C) powertype.
(D) metaclass.
(E) stereotype.

Padrões de Projeto

TRT-4 (FCC 2011)

59. O catálogo de padrões de projeto (design patterns) do GoF contém

a) 20 padrões e está basicamente dividido em duas seções: Structural e Behavioral.

b) 21 padrões e está basicamente dividido em duas seções: Creational e Behavioral.

c) 23 padrões e está basicamente dividido em duas seções: Structural e Behavioral.

d) 23 padrões e está, basicamente, dividido em três seções: Creational, Structural e Behavioral.

e) 24 padrões e está basicamente dividido em três seções: Creational, Spectral e Behavioral.

INFRAERO (FCC 2011)

56. Para fazer a modelagem de um padrão de projeto (design pattern) utilizando a UML é DESNECESSÁRIO

a) entender o padrão de projeto como uma colaboração representada com suas partes estruturais e comportamentais.

b) fazer a modelagem do mecanismo como uma colaboração, fornecendo seus aspectos estruturais, assim como os aspectos comportamentais.

c) identificar as soluções específicas e particulares para o problema básico e reificá-la como um mecanismo.

d) identificar os elementos do padrão de projeto que devem ser vinculados aos elementos em um contexto específico e representá-los como parâmetros para colaboração.

e) identificar as soluções comuns para o problema básico.

TRT-14 (FCC 2011)

55. No contexto dos padrões de projeto:

I. Oferecer uma interface simples para uma coleção de classes.

II. Desacoplar uma abstração de sua implementação para que ambas possam variar independentemente.

Correspondem respectivamente a

a) Façade e Bridge.
b) Adapter e Façade.
c) Composite e Bridge.
d) Façade e Composite.
e) Bridge e Adapter.

TRT-24 (FCC 2011)

54. Considere:

I. Fornecer uma interface para criação de famílias de objetos relacionados ou dependentes, sem especificar suas classes concretas. Possibilitar o adiamento da instanciação para as subclasses.

II. Garantir a existência de apenas uma instância de uma classe, mantendo um ponto global de acesso ao seu objeto.

III. Possibilitar o armazenamento do estado interno de um objeto em um determinado momento, para que seja possível retorná-lo a este estado, caso necessário.

I, II e III são, respectivamente, objetivos dos design patterns intitulados:

a) Interpreter, Iterator e Memento.
b) Command, Singleton e Iterator.
c) Factory Method, Singleton e Memento.
d) Iterator, Factory Method e Flyweight.
e) Singleton, Flyweight e Command.

TRT-9 (FCC 2010)

36. Sobre design pattern considere:

I. No framework pode incluir código de programação e conter vários design patterns.

II. No design pattern pode incluir código de programação e conter vários frameworks.

III. Os design patterns são bastante abstratos e os frameworks menos abstratos.

Está correto o que consta em

a) I e III, apenas.
b) I e II, apenas.
c) II e III, apenas.
d) III, apenas.
e) I, II e III.

Gabaritos

Análise e Projeto OO

TRT-23 (FCC 2011)
41 E, 43 D

NOSSA CAIXA DESENVOLVIMENTO (FCC 2011)
45 E

TRT-14 (FCC 2011)
43 B

TRE/AP (FCC 2011)
25 C, 55 C

TRE/SP (FCC 2012)
41 C

TRE/CE (FCC 2012)
41 D

TJ/PE (FCC 2012)
53 B, 54

BANESE (FCC 2012)
67 B

METRO/SP (FCC 2012)
36 D, 43 A

Testes e Qualidade de Software

INFRAERO (FCC 2011)
47 E

TRT-4 (FCC 2011)
52 C

TRT-20 (FCC 2010)
42 E

BAHIAGÁS (FCC 2010)
56 B

TRT-9 (FCC 2010)
32 C

UML

TRT-19 (FCC 2011)
33 D, 36 A

INFRAERO (FCC 2011)
43 C, 44 E, 45 E, 46 D, 47 B, 50 B, 51 B, 52 A, 60 D

MPE/PE (FCC 2012)
44 A

TRE/CE (FCC 2012)
34 C

TJ/PE (FCC 2012)
51 E

TRT/AM (FCC 2012)
41 D, 42 D

BANESE (FCC 2012)
65 B

Padrões de Projeto

TRT-4 (FCC 2011)
59 D

INFRAERO (FCC 2011)
56 C

TRT-14 (FCC 2011)
55 A

TRT-24 (FCC 2011)
54 C

TRT-9 (FCC 2010)
36 A