

PROJETO – POO

1 – Que tipo de característica da POO, que demonstra que um determinado software se especializa a cada momento, nunca ficando naquela mesma versão encontrada no momento de sua criação:

- a) Oportuno – **Ciclo de desenvolvimento do sistema mais rápido**
- b) Confiável
- c) **Extensível**
- d) Reutilizável – **Reutilização de código em outros projetos**

2 – Que tipo de característica, a natureza modular dos objetos permite que você faça alterações em uma parte de seu programa, sem afetar outras partes. Os objetos isolam o conhecimento e a responsabilidade de onde pertencem.

- a) Natural – **Código mais natural**
- b) **Confiável**
- c) Manutenível – **Faz as subclasses se atualizarem de acordo com a manutenção de código da mãe**
- d) Oportuno

3 – O que significa um combinado de **Variáveis internas (Variáveis de Membro)** – **valor mantido dentro de um atributo** - de um objeto:

- a) Implementação - **código**
- b) **Estado de um objeto**
- c) Atributos
- d) Classe

4 – Quando se constrói um software que encapsula estado e comportamento, permitindo que você o modele (software) em termos reais de abstração, criamos:

- a) Comportamento – **Ação de um objeto**
- b) Variável
- c) Atributo
- d) **Objeto**

5 – O que define **os atributos e comportamentos** comuns compartilhados por um tipo de objeto.

- a) **Classe**
- b) Método – **está ligado a ação de um objeto, ou seja, o comportamento**
- c) O próprio objeto
- d) Variáveis

6 - São características de uma classe visíveis externamente:

- a) variáveis
- b) **atributo**
- c) método
- d) Classe abstrata

7 - Um objeto pode exercer o comportamento do outro, executando uma operação sobre esse objeto, a partir de uma, EXCETO:

- a) Chamada de Função
- b) Chamada de Método
- c) Passagem de Mensagem
- d) **Chamada de objeto**

8 - Chama-se criação de objetos de _____.

- a) **Instanciação**
- b) Criação
- c) Acesso
- d) Modelagem

9 – Que tipo de método permite que você mude ou recupere (**MUDANÇA TEMPORÁRIA**) o valor e têm ‘efeitos colaterais’ sobre o seu estado interno.

- a) Construtor – instanciar um objeto
- b) Mutante (modificadores ou setters) - Modificar o estado interno do objeto.
- c) NOARG – sem argumento
- d) Acessor (getters)

10 – Que tipo de característica da OO permite que você construa partes ocultas da implementação do software, que atinjam uma funcionalidade e ocultam os detalhes de implementação do mundo exterior.

- a) Encapsulamento
- b) Herança – Herda métodos e atributos de uma classe previamente existente.
- c) Polimorfismo – Um mesmo nome de método para vários comportamentos diferentes.
- d) Modularidade

11 – O que é utilizado para definir como um comportamento realmente fornece um serviço:

- a) Interface – Lista os serviços de um componente
- b) Implementação - codificação
- c) Classe
- d) Objeto

12 – Que nível de acesso garante o acesso à instância, ou seja, para aquele objeto, e para todas as subclasses.

- a) Público – para todas as classes (+)
- b) Privado – Somente para a classe que a possui (-)
- c) Protegido (#)
- d) Aberto

13 – São características do encapsulamento eficaz, EXCETO:

- a) Abstração – Cria objetos abstratos com o intuito de futuras implementações (Especialização)
- b) Divisão de Responsabilidade – Cada classe tem seu papel definido dentro da hierarquia de classes.
- c) Ocultação de Implementação – Proteger os objetos dos usuários e os usuários do objeto.
- d) Proteção de Implementação

14 – O que permite que se defina novos tipos na linguagem, ocultando dados internos e o estado, atrás de uma interface bem definida:

- a) Interface pública
- b) Interface Privada
- c) TAD – Tipo Abstrato de Dados
- d) Tipo – Espécie de valores (Domínio de um problema)

15 – Relacione a segunda coluna de acordo com a primeira:

I – Interface

II – Tipo

III - código fracamente acoplado (**Independência**)

IV - objeto de primeira classe

V - Encapsulamento Efetivo

VI - Métodos de Classe

(VI) pertencem à classe e não a uma instância específica

(III) independente da implementação de outros componentes.

(V) a abstração mais ocultação da implementação mais responsabilidade.

(IV) pode ser usado exatamente da mesma maneira que um tipo interno

(I) lista os serviços fornecidos por um componente

(II) define espécies de valores que você pode usar em seus programas

a) V-II-VI-IV-I-III

b) IV-III-V-VI-II-I

c) VI-III-V-IV-I-II

d) I-II-V-IV-VI-III

16 – O que seria um objeto cujo único propósito é conter outro objeto ou primitiva.

a) Objeto composto

b) Pacote

c) Componente

d) Objeto imutável

17 - Processo de um objeto passar uma mensagem para outro objeto, para atender algum pedido:

a) Operação

b) Método

c) Delegação

d) Composição

18 – Quando uma classe é implementada usando-se variáveis internas (chamadas de variáveis membro), que contêm instâncias de outras classes, chamamos de:

a) sobreposição

b) recursividade

c) Composição

d) pacote

19 – Que tipo de método estaria utilizando, quando uma nova classe simplesmente herda um método ou atributo da progenitora.

a) Nova – **implementação de um método que não se encontra na mãe.**

b) Recursiva

c) Sobreposto – **Modificação de um método herdado da mãe.**

d) Abstrato

20 – O que significa herdar uma classe e adicionar apenas o código que torne a nova classe diferente da classe herdada.

a) Recursividade

b) Sobreposição

c) Programação por diferença

d) Programação por reutilização

21 - Permite que você use o mesmo nome de método para muitos métodos diferentes. Cada método diferem apenas no número e no tipo de seus parâmetros.

- a) Polimorfismo Paramétrico – tratar métodos e tipos genéricos
- b) Polimorfismo Ad-Hoc – Sobrecarga / Overload – este é o único tipo de polimorfismo que pode ser feito dentro da mesma classe. Modifica a assinatura da mãe. A conversão e a sobrecarga andam juntas.
- c) Polimorfismo de Inclusão – Também pode ser chamado de puro – trata objetos relacionados genericamente.
- d) Polimorfismo Puro

* Sobreposição – (overriding) – inclui um apenas um código que diferencia da mãe, mas mantém a assinatura.

22 – O que define um procedimento para projeção de um software:

- a) Metodologia
- b) Planejamento
- c) Análise
- d) Implementação

23 - Elemento da UML que permite que se amplie o vocabulário da própria linguagem UML:

- a) Associação – tipo de relacionamento (composição ou agregação)
- b) Modelo de relacionamento
- c) Estereótipo << explicação>>
- d) Agregação

24 - Três tipos de alto nível de relacionamentos de objetos:

- a) Dependência, Associação, Generalização
- b) Agregação, Composição, Associação
- c) Generalização, Especialização, Associação
- d) Dependência, Especialização, Generalização

25 – Que tipo de associação não é um relacionamento entre pares (independência), ou seja, não são independentes uns dos outros:

- a) Agregação – modela um relacionamento entre pares.
- b) Especialização
- c) Generalização
- d) Composição

26 – Que tipo de relacionamento descreve o relacionamento entre objetos onde um objeto contém outro:

- a) tem um – Agregação - descreve o relacionamento em que uma classe contém uma instância de outra classe.
- b) é um – Descreve um relacionamento onde diz que um objeto é do mesmo tipo do outro. (composição)
- c) Todo/parte (entre pares – tem um) (é um)
- d) Associação

Todo/parte entre pares - Agregação

Todo/parte que não é entre pares - Composição

27 – O que indica um relacionamento entre o geral e o específico, ou seja, a herança. Se existe esse relacionamento, se sabe que pode substituir uma classe filha pela progenitora

- a) Especialização
- b) Generalização
- c) Agregação
- d) Associação

28 - Processo permite alterações em qualquer ponto do processo de desenvolvimento. Permite alteração adotando uma estratégia iterativa e incremental, para desenvolvimento do software:

- a) Processo Iterativo
- b) Processo em cascata – desenvolvimento onde a alteração é feita no final do processo somente.
- c) Sistema
- d) Análise de requisitos

29 – Etapas de um processo em cascata ou iterativo, EXCETO:

- a) Manutenção
- b) Análise de requisitos
- c) Projeto

c) Implementação e teste

30 – O processo iterativo utiliza duas estratégias para desenvolvimento do software:

- a) Iterativo e Especializado
- b) Iterativo e Requisitos
- c) Iterativo e Incremental
- d) Incremental e Especializado

31 - Uma sequência ou fluxo de eventos entre o usuário e o sistema:

- a) Ator – qualquer um que interage com o sistema (pessoa ou um outro sistema)
- b) Variante – Versão de um caso de uso
- c) Cenário
- d) Condições (prévias ou posteriores)

Caso de uso – descreve a interação do usuário com o sistema.

32 – Os diagramas de interação ajudam a modelar os relacionamentos entre casos de uso. Ajudam também a capturar as interações entre os vários atores participantes do sistema, e divide-se em:

- a) Sequência e Colaboração
- b) Atividade e Domínio
- c) Sequência e Atividade
- d) Domínio e Colaboração

33 – Que tipo de Diagrama que ajudam a modelar processos que podem ser executados em paralelo.

- a) Caso de uso
- b) Atividade
- c) Domínio
- d) Colaboração

34 – É o relacionamento onde os objetos interagem para realizar o mesmo propósito:

- a) Projeto Orientado a Objetos
- b) Domínio
- c) Colaboração
- d) CRC – diagrama utilizado para especificar as colaborações entre objetos.

35 – Quem faz a mediação entre dois ou mais objetos para atingir algum objetivo.

- a) Ator
- b) Analista
- c) Projetista
- d) Agente

36 – Que tipo de padrão tem um substituto ou lugar reservado que intermedia o acesso ao objeto de interesse real:

- a) Iterator
- b) Proxy
- c) Adapter
- d) Case

37 – Que tipo de Padrão tem um dispositivo que é um objeto que transforma a interface de outro objeto:

- a) Iterator – fornece uma interface genérica
- b) Proxy
- c) Adapter
- d) Case

38 – Que tipo de teste, testa se o sistema funciona conforme o esperado:

- a) Teste de Caixa preta
- b) Teste de Caixa branca – código (100% código)
- c) Unidade
- d) Integração

* Sistema – Testa o sistema se está de acordo com o caso de uso, e também se comporta com situações inesperadas.

39 – Que tipo de teste, envia uma mensagem para um objeto e depois verifica se ele recebe o resultado esperado do objeto:

- a) Teste de Caixa preta
- b) Regressão
- c) Unidade
- d) Integração – **Testa de dois objetos se comunicam perfeitamente.**

40 – Que tipo de teste examina as alterações nas partes do sistema que já foram validadas:

- a) Teste de Caixa preta
- b) **Regressão**
- c) Unidade
- d) Integração

41 - Prepara o conjunto de objetos sobre os quais um caso de teste atuará:

- a) **Acessório de teste**
- b) Objeto falsificado
- c) Classe Anônima
- d) Stubs

42 – **Objetos** que realizam apenas trocas de mensagens enquanto que os objetos falsificados se diferem no sentido de que eles realmente executam alguma função, em vez de simplesmente aceitarem uma chamada e retornarem algum valor prévio.

- a) Acessório de teste
- b) Estrutura
- c) Classe Anônima
- d) **Stubs – utilizados para fazer teste de unidade ou de integração**

43 - Durante o processo de desenvolvimento de software podem ocorrer erros. Felizmente, através de testes de software, podemos tomar medidas preventivas contra esses erros, nas quais é possível validar a análise, o projeto e a implementação de seu software. Entre os diversos tipos de teste, qual verifica se dois ou mais objetos funcionam em conjunto corretamente?

- a) Teste de Sistema
- b) Teste de Unidade
- c) **Teste de Integração**
- d) Teste de Regressão

44 - Objeto de _____ é aquele que pode ser usado exatamente da mesma maneira que um tipo interno.

- a) classe externa
- b) classe interna
- c) **primeira classe**
- d) segunda classe

45 - Correlacione os conceitos utilizados pela Programação Orientada a Objetos e, em seguida, assinale a alternativa que contém a sequência correta.

- (1)- Atributo
- (2)- Classe
- (3)- Comportamento
- (4)- Domínio
- (5)- Objeto

(3) é uma ação executada por um objeto quando passada uma mensagem ou em resposta a uma mudança de estado.

(2) define os atributos e comportamentos comuns compartilhados por um tipo de objeto.

(1) é a característica de uma classe visível externamente.

(5) é uma construção de software que encapsula estado e comportamento.

(4) é o espaço onde um problema reside.

- a) 3, 2, 4, 5, 1
- b) **3, 2, 1, 5, 4**
- c) 2, 3, 1, 5, 4
- d) 2, 4, 3, 1, 5

46 - O polimorfismo paramétrico permite que você

- a) trate objetos relacionados genericamente.
- b) **crie métodos e tipos genéricos.**
- c) use o mesmo nome de método para muitos métodos diferentes.
- d) escreva vários métodos para somar todos os tipos.

47 Um dos três pilares da Programação Orientada a Objetos (POO) é a Herança. A seguir estão alguns termos relacionados com este pilar da POO. Relacione a coluna da esquerda com a da direita, e, em seguida, assinale a alternativa que contém a sequência correta.

- I- Herança
- II- Delegação
- III- Composição
- IV- Programação por diferença
- V- Classe Raiz / **base / metaclasses**

(IV) Significa herdar uma classe e adicionar apenas o código que torne a nova classe diferente da classe herdada.

(III) Significa que uma classe é implementada usando-se variáveis internas (chamadas de variáveis membro), que contêm instâncias de outras classes.

(I) É um mecanismo que permite a você basear uma nova classe na definição de uma classe previamente existente.

(II) É o processo de um objeto passar uma mensagem para outro objeto, para atender algum pedido.

(V) É a classe superior da hierarquia de herança.

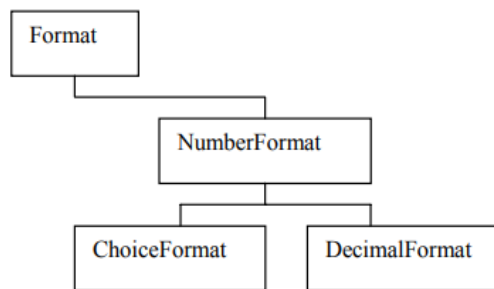
a) IV - III - I- II - V

b) I - V- II - IV - III

c) IV - III - I - V- II

d) II - I - IV - III - V

48 – Observe o diagrama a seguir e assinale a alternativa correta, no que diz respeito à classe “Decimal Format”.



a) Qualquer classe dada será ascendente de Format.

b) DecimalFormat é descendente de Format.

c) DecimalFormat é ancestral de Format.

d) ChoiceFormat é “filha” de DecimalFormat

49 – Relacione a coluna da direita com a da esquerda, a seguir, assinale a alternativa que contém a sequência correta.

- (1)- Estado
- (2)- Objeto
- (3)- Comportamento

(2) Instância de uma classe.

(1) É o significado combinado das variáveis internas do objeto.

(3) É algo que um objeto faz.

a) 1, 2, 3

b) 2, 1, 3

c) 2, 3, 1

d) 3, 1, 2

50 – Assinale a alternativa que completa correta e respectivamente as lacunas da assertiva a seguir. Quando se utiliza o conceito de encapsulamento da programação orientada a objetos, enquanto a _____ define os detalhes internos do componente, a _____ lista os serviços fornecidos por ele.

a) interface – implementação

b) classe – implementação

c) interface – classe

d) implementação - interface

51 – De acordo com os conceitos da programação orientada a objetos, relacione as colunas e, a seguir, assinale a alternativa com a sequência correta.

- | | |
|----------------------|---|
| 1- Classe | () Define exatamente o que uma entidade externa pode fazer com o objeto. |
| 2- Interface | () Instanciação de uma classe. |
| 3- Método Construtor | () Define os atributos e comportamentos comuns compartilhados por um tipo de objeto. |
| 4- Objeto | () Utilizado para inicializar objetos durante sua instanciação. |

a) 1,2,4,3

b) 1,4,2,3

c) 2,3,1,4

d) 2,4,1,3

52 – Informe se é falso (F) ou verdadeiro (V) o que se afirma abaixo sobre os conceitos da programação orientada a objetos. A seguir, assinale a alternativa que apresenta a sequência correta.

- () A especialização permite que você remova da classe filha comportamentos e atributos herdados.
- () A superclasse herda os atributos e comportamentos da subclasse.
- () Os métodos de acesso privado não são herdados pela classe filha.
- () Protegido é o nível de acesso no qual se restringe o acesso à classe e às classes filhas.

a) F,F,V,F

B) F,F,V,V

C) V,V,F,F

D) V,F,V,V

53 – De acordo com os conceitos da programação orientada a objetos, relacione as colunas e, a seguir, assinale a alternativa com a sequência correta.

- | | |
|--------------------|---|
| 1- Classe abstrata | () Não pode ser sobrescrito, ou seja, não pode ser redefinido na classe-filha. |
| 2- Classe final | () Nunca será instanciada na forma de objeto. |
| 3- Método Abstrato | () Não pode ser uma superclasse. |
| 4- Método Final | () Deverá conter uma implementação na classe-filha. |

a) 3,1,4,2

b) 4,2,1,3

c) 4,1,2,3

d) 3,1,2,4

54 – Marque V para verdadeiro e F para falso a respeito das afirmações sobre Programação Orientada a Objetos. Em seguida, assinale a alternativa que corresponde à sequência correta.

- () Encapsulamento é a característica da Orientação a Objetos que permite construir partes ocultas da implementação do *software*, que atinjam uma funcionalidade.
- () O uso do mecanismo da Herança permite que sua nova classe herde todos os atributos e comportamentos presentes na classe previamente existente.
- () Todas as linguagens suportam o polimorfismo.
- () Uma linguagem orientada a objetos considera tudo um objeto.
- () Os métodos de classe pertencem à classe e não a uma instância específica.

- a) V-F-F-V-F
- b) F-V-V-V-F
- c) **V-V-F-F-V**
- d) F-F-V-F-V

55 – A seguir estão alguns termos relacionados à “herança”, um dos três pilares da Programação Orientada a Objetos (POO). Relacione a coluna da esquerda com a da direita e, em seguida, assinale a alternativa que contém a sequência correta.

- | | |
|-----------------|---|
| 1- Herança | () Trata-se do relacionamento em que uma classe contém uma instância de outra classe. |
| 2- “Tem um” | () É a classe que está herdando. |
| 3- Classe filha | () É um mecanismo que permite basear uma nova classe na definição de uma classe previamente existente. |
| 4- Classe Raiz | () É a classe superior da hierarquia de herança. |
| 5- Classe Folha | () É uma classe sem filhas. |

- a) 2-1-4-3-5
- b) 3-2-1-4-5
- c) 4-3-2-5-1
- d) **2-3-1-4-5**

56 – Considerando os conceitos utilizados em Programação Orientada a Objetos (POO), relacione as colunas e, em seguida, assinale a alternativa que apresenta a sequência correta.

- | | |
|------------------|--|
| 1- Classe | () Dão acesso aos dados internos de um objeto. |
| 2- Atributos | () É a forma através da qual os objetos se comunicam uns com os outros. |
| 3- Comportamento | () São as características de uma classe visíveis externamente. |
| 4- Mensagem | () Define todas as características comuns a um tipo de objeto. |
| 5- Acessores | () É algo que um objeto faz. |

- a) 5-4-3-2-1
- b) **5-4-2-1-3**
- c) 3-4-2-5-1
- d) 4-2-5-1-3

57 – Assinale a alternativa que completa correta e respectivamente as lacunas do texto a seguir.

Em Programação Orientada a Objetos, pode-se afirmar que a herança é um mecanismo que permite basear uma nova classe na definição de uma outra previamente existente. Quando isso ocorre, a nova classe herda automaticamente todos os _____, _____ e _____ presentes na classe previamente existente.

- a) atributos, comportamentos e implementações
- b) Códigos, Instâncias e implementações
- c) comportamentos, instâncias e códigos
- d) termos, atributos e instâncias

58 — Assinale a alternativa que completa correta e respectivamente as lacunas da assertiva a seguir relacionada à programação orientada a objetos. O nível de acesso que você escolhe é muito importante para seu projeto. O acesso _____ garante o acesso a todos os objetos. Para garantir o acesso à instância, você precisa ter acesso _____ ou _____. Lembrando que o acesso _____ garante o acesso apenas para a instância, ou seja, para aquele objeto.

- a) público – protegido – privado – privado
- b) privado – privado – protegido – protegido
- c) protegido – público – privado – privado
- d) público – privado – protegido – protegido

59 – Dentre as opções abaixo, qual representa uma das maneiras de se basear casos de teste?

- a) Caixa preta.
- b) Livre de erro.
- c) Teste unitário reverso.
- d) Teste de espera.

60 – Segundo Anthony Sintès, quais são as armadilhas que precisam ser evitadas no aprendizado, pela primeira vez, da Orientação a Objetos?

- a) Pensar na POO simplesmente como uma linguagem; medo da reutilização; pensar na OO como uma solução para tudo; programação egoísta.
- b) Pensar natural; pensar de forma confiante; pensar na reutilização; pensar na manutenção.
- c) Pensar natural; pensar na POO simplesmente como uma linguagem; pensar na manutenção; programar de forma egoísta.
- d) Pensar natural; medo da reutilização; pensar na OO como uma solução para tudo; programar de forma egoísta.

61 – Segundo Anthony Sintès, quais são as três características do encapsulamento eficaz na Orientação a Objetos?

- a) Abstração, ocultação da implementação e divisão de responsabilidade.
- b) Implementação de classes com atributos públicos, herança simples e abstração.
- c) Abstração, a não divisão de responsabilidade e a ocultação da implementação.
- d) Abstração, ocultação da implementação dependendo da linguagem e a negação da divisão de responsabilidade.

62 – Marque (V) para verdadeiro ou (F) para falso. Em seguida, assinale a alternativa com a sequência correta.

- (F) Objeto é um elemento da UML que permite a ampliação do vocabulário da própria linguagem UML.
- (V) Na UML um relacionamento é uma conexão entre dois ou mais elementos da notação.
- (V) A implementação define como algo é feito. Em termos de programação, implementação é o código.
- (F) Estereótipo é uma construção de software que encapsula estado e comportamento.

- a) F – V – V – V
- b) V – V – F – V
- c) V – F – V – F
- d) F – V – V – F

63 – Considerando os conceitos de introdução a UML, assinale a alternativa cuja assertiva é verdadeira.

- a) Um estereótipo descreve um relacionamento ‘é um’ entre duas classes.
- b) Uma linguagem de modelagem é uma notação hexadecimal para descrever projetos de software.
- c) Uma agregação é um tipo especial de associação que modela o relacionamento ‘tem um’ entre pares.
- d) Uma composição é menos rigorosa do que a agregação, pois é um relacionamento entre pares completamente independentes

64 – Indique a opção que completa corretamente a lacuna da assertiva a seguir.

“Considerando-se as definições de herança na programação orientada a objetos, um método ou atributo _____ é simplesmente herdado de uma classe progenitora.”

- a) novo
- b) recursivo
- c) construtor
- d) sobreposto

65 – Levando em consideração os conceitos básicos de análise orientada a objetos (AOO) e a introdução à UML, assinale a alternativa correta.

- a) Requisitos **CASO DE USO** descrevem a interação entre o usuário e o sistema.
- b) Um processo de desenvolvimento de software mostra exclusivamente o estágio de implementação do código.
- c) Uma variante de caso (**VERSÃO DE UM CASO DE USO**) de uso é um atributo que deve ser usado em todas as classes que envolvem o caso de uso no sistema.
- d) **Denomina-se ator tudo que interage com o sistema. Pode ser um usuário humano ou outro sistema de computador.**

66 - Tratando-se dos tipos de teste de software, assinale a alternativa que fala sobre testes de integração.

- a) Examinam todo o sistema.
- b) Verificam apenas um recurso por vez.
- c) **Verificam se dois ou mais objetos funcionam em conjunto corretamente.**
- d) Examinam as alterações nas partes do sistema que foram validadas anteriormente.

67 - Um recurso poderoso da OO é o encapsulamento que possibilita, entre outras funcionalidades, dividir responsabilidades e ocultar a implementação. Os níveis de acesso são responsáveis por garantir tais recursos. Quando se deseja que o acesso seja disponível apenas à instância, ou seja, só para o objeto, aplica-se o nível de acesso

- a) público.
- b) **privado.**
- c) recursivo.
- d) protegido.

68 - Complete o espaço da frase abaixo e assinale a alternativa correta. Em desenvolvimento de sistemas, o termo _____ é utilizado quando uma classe não tem filhas.

- a) classe base
- b) **classe folha**
- c) classe estéril
- d) classe anamorfa

69 - Os objetos se comunicam uns com os outros através de _____. Os objetos podem existir _____ uns dos outros e ademais, um objeto poderia conter outros _____. Assinale a alternativa que corresponde respectivamente ao correto preenchimento das lacunas.

- a) acessores – independentemente – objetos.
- b) acessores – dependentemente – agregados.
- c) **mensagens – independentemente – objetos.**
- d) mensagens – independentemente – modelos.

70 – Sobre o POO, é correto afirmar que

- a) não permite que você explore cada solução e a decida antecipadamente.
- b) os erros no projeto têm um alto custo para serem corrigidos, facilitando sua criação.
- c) o modelo de objeto não contera as classes principais do projeto, suas responsabilidades e uma definição. **CRC**
- d) **é o processo que constrói o modelo de objeto de uma solução e que divide uma solução em vários objetos constituintes.**

71 – Sobre UML (Unified Modeling Language), marque V para verdadeiro e F para falso. Em seguida, assinale a alternativa com a sequência correta.

- (V) A UML faz diferença entre operação e métodos
- (V) Na UML uma operação é um serviço que você solicita de qualquer objeto de uma classe, enquanto um método é uma implementação específica da operação.
- (V) A UML consiste na notação para descrever cada aspecto de um projeto de software.
- (F) A UML é única linguagem de modelagem reconhecida e aceita.

- a) V – V – F – F

- b) F – V – F – V
- c) V – F – V – V
- d) V – V – V – F

72 - – Qual alternativa está correta sobre a importância das UIs (User Interface) desacopladas em um único sistema?

- a) Um projeto flexível não suporta alterações em um único sistema.
- b) Um projeto desacoplado permite que você aponte os erros que são da UI ou que são do sistema; o segredo é incorporar o código da UI dentro do próprio sistema.
- c) Um projeto cujos recursos do sistema sejam dependentes torna os testes seguros, reduz o índice de alterações indesejáveis e facilita a criação e atualização de novos recursos.
- d) Um projeto desacoplado permite que você adicione qualquer UI no sistema e faça alterações nas UIs existentes, sem ter de fazer alterações correspondentes no sistema em si.

73 - – Com relação à Programação orientada a objetos, marque V para verdadeiro ou F para falso. Em seguida, assinale a alternativa com a sequência correta.

- (V) Os objetos se comunicam uns com os outros através de mensagens.
- (F) Uma variável interna é um comportamento mantido dentro de um objeto.
- (V) Um objeto é uma construção de software que encapsula estado e comportamento.
- (V) O domínio é o conjunto de conceitos que representam aspectos importantes do problema que um desenvolvedor precisa resolver.

- a) V – V – F – F
- b) V – F – V – V
- c) F – V – V – F
- d) F – F – V – V

74 - – Considere as representações UML das classes abaixo e assinale a alternativa que apresenta uma sobrecarga de métodos.

a) Comparador Numerico
+ max(x : int, y : int) : int
+ max(x : long, y : long) : long

b) ComparadorNumerico
+ max(x : int, y : int) : int
+ min(x : int, y : int) : int

c) ComparadorNumerico
+ max(x : int, y : int) : int
+ min(x : int) : void

d) ComparadorNumerico
+ max(x : int, y : int) : int
+ maior(x : int, y : int) : int

75 - – Considerando os conceitos de introdução a UML, assinale a alternativa cuja assertiva é verdadeira.

- a) Um estereótipo descreve um relacionamento ‘é um’ entre duas classes.
- b) Uma linguagem de modelagem é uma notação hexadecimal para descrever projetos de software.
- c) Uma agregação é um tipo especial de associação que modela o relacionamento ‘tem um’ entre pares.
- d) Uma composição é menos rigorosa do que a agregação, pois é um relacionamento entre pares completamente independentes