

Residência em Tecnologia da Informação Aplicada à Área Jurídica

Edital 002/2018

Prova de Conhecimentos Específicos

Área de Concentração 1 - Analista Desenvolvedor de Software

Candidato: _____

CPF: _____ **Telefone:** _____

Questões

1. Sobre programação estruturada e programação orientada a objetos, é INCORRETO afirmar que:

- a) No paradigma de programação estruturado, qualquer problema pode ser dividido em problemas menores, chamados de funções.
- b) A linguagem de programação C é um exemplo de linguagem de programação estruturada, compilada e procedural.
- c) O paradigma orientado a objetos entende o problema como um conjunto de objetos interagindo por meio de troca de mensagens.
- d) A linguagem de programação C++ é um exemplo de linguagem orientada a objetos, que não permite a utilização do paradigma estruturado na solução de um problema.

2. Na linguagem de programação C, as strings “%d”, “%f” e “%s” estão usualmente associadas ao uso da função:

- a) printf
- b) printf
- c) print
- d) printf

3. Considere a declaração abaixo, feita em C++:

for(«expressao1»; «expressao2»; «expressao3») { «comandos» };

Pode-se afirmar que o comando while equivalente a esse for é:

- a) while («expressao2»){ «expressao1»; «comandos»; «expressao3» };
- b) «expressao3»; while («expressao2») { «expressao1»; «comandos» };
- c) «expressao1»; while («expressao2») { «comandos»; «expressao3» };
- d) «expressao3»; while («expressao2») { «comandos»; «expressao1»};.

4. A respeito das tecnologias Java/Java EE, assinale a alternativa correta.

- a) EJB é uma arquitetura de componentes multi-plataforma para o desenvolvimento de aplicações distribuídas, escaláveis e orientadas a objetos.
- b) JTA é uma API da linguagem Java que permite a componentes baseados em Java/Java EE criar, enviar, receber e ler mensagens.

- c) JPA é uma especificação feita pela Sun que visa padronizar o uso de transações distribuídas feitas por aplicativos Java.
- d) JMS é um framework para desenvolver aplicações web de forma ágil, permite que o desenvolvedor crie UIs através de um conjunto de componentes UIs pré-definidos.

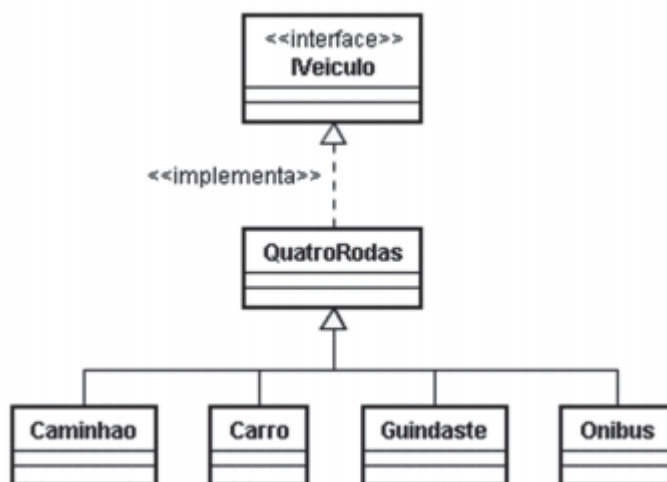
5. Analise as afirmativas abaixo sobre POO (Programação Orientada a Objetos).

- I. _____ são classes onde os objetos nunca são instanciados diretamente.
- II. _____ ocorre quando uma classe herda características de apenas uma superclasse.
- III. _____ permite a comunicação entre objetos.
- IV. _____ é o mecanismo utilizado para impedir o acesso direto aos atributos de um objeto.

Correspondem, respectivamente, às afirmativas I, II, III e IV:

- a) Classes Abstratas, Herança Múltipla, Polimorfismo, Encapsulamento.
- b) Superclasses, Herança Simples, Atributo, Instanciação.
- c) Classes Abstratas, Herança Simples, Método, Encapsulamento.
- d) Superclasses, Herança Múltipla, Polimorfismo, Instanciação.

7. Considere a hierarquia de classes mostrada na Figura a seguir:



Considere agora o seguinte fragmento de código Java baseado na hierarquia anterior:

```

public class Main {
    public static void main(String[] args) {
        Caminhao caminhao = new Caminhao();           // linha 1
        IVeiculo veiculo = caminhao;                   // linha 2
        QuatroRodas fw = new Guindaste();              // linha 3
        fw = veiculo;                                   // linha 4
        veiculo = fw;                                   // linha 5
    }
}

```

A linha marcada por um comentário que causará um erro em tempo de compilação é a linha

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

8. Considere a seguinte consulta SQL: **SELECT XXX A FROM TAB**. Para que não haja repetições nos valores obtidos como resposta para o atributo *A*, **XXX** deve ser substituído por:

- a) IN
- b) LIKE
- c) SINGLE
- d) DISTINCT

9. Considere as seguintes tabelas de um banco de dados relacional, sendo que os atributos *ID* e *Cod*, na tabela *Usa* são chaves estrangeiras, com origem, respectivamente, nas tabelas *Dpto* e *Eqpto*:

Depto (ID, Nome, Função)
Usa (ID, Cod)
Eqpto (Cod, Tipo, Valor)

Assinale a alternativa que apresenta a expressão SQL para obter o *Nome* do *Depto* e o *Tipo* de *Eqpto* utilizados.

- a) **SELECT Nome, Tipo FROM Usa WHERE USA.ID and Usa.Cod**
- b) **SELECT Nome, Tipo FROM Depto, Usa, Eqpto**
- c) **SELECT Nome, Tipo FROM Depto, Usa, Eqpto WHERE Depto.ID = USA.ID and Eqpto.Cod = Usa.Cod**
- d) **SELECT Nome, Tipo FROM Depto, Eqpto WHERE Depto.ID and Eqpto.Cod**

10. O padrão de arquitetura MVC é comumente adotado para construção de aplicações web. Sobre este padrão considere as seguintes afirmativas:

- I A View exibe para o usuário os dados fornecidos pelo Controller através de uma interface gráfica produzida por ele mesmo.

- II Em uma aplicação Java EE, arquivos JSP são considerados componentes do tipo Controller do MVC, já que estes são convertidos em servlet quando a aplicação é compilada e implantada.
- III As classes DAO (que acessam o banco de dados) são representadas no componente Model.
- IV O fluxo de comunicação entre os componentes View e Model no padrão MVC deve passar obrigatoriamente pelo componente Controller, pois o este último é o responsável pelo controle da comunicação entre os objetos.

As afirmativas corretas são:

- a) I e III
- b) I e IV
- c) II e IV
- d) II e III

11. PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional baseado em código aberto. Considerando as rotinas de atividades de manutenção desse Banco de Dados, é correto afirmar acerca do comando VACUUM:

- a) sua variante VACUUM FULL requer bloqueio exclusivo na tabela em que estiver trabalhando.
- b) o parâmetro FULL possibilita recuperação de maior espaço de disco, contudo toma pouco tempo para sua execução.
- c) não cria uma substancial quantidade de tráfego de Entrada e Saída (I/O), não afetando a performance de outras sessões ativas.
- d) é possível visualizar um relatório de atividade detalhado e indexar cada linha de registro para cada tabela, utilizando o parâmetro VERBOSE.

12. Analise as construções exibidas a seguir para definição da propriedade CSS "padding" de um elemento.

```
p { padding: 50px; }  
p { padding: 50px 20px; }  
p { padding: 50px 20px 30px; }  
p { padding: 50px 20px 30px 70px; }
```

Assinale o número de linhas que estão corretamente declarados.

- a) Uma.
- b) Duas.
- c) Três.
- d) Quatro.

13. Um desenvolvedor está utilizando Bootstrap 4 na criação de um site e deseja definir que um container deve usar toda a largura da tela. Assim, terá que utilizar na tag <div> o atributo class igual a:

- a) container.
- b) stretch.
- c) container-fluid.

d) full-width.

14. Em relação ao uso de JavaScript em browsers Web, considere as seguintes afirmativas:

- I** A execução do comando `var d = Date.currentTime()` atribui à variável “d” um objeto que representa a data e hora atual.
- II** O comando `document.getElementById(“xxx”).atrib` pode ser usado pra acessar o valor do atributo “atrib” de um elemento cujo *id* é “xxx”.
- III** Em JavaScript um vetor é criado de forma similar a Java, ou seja, usando o comando na forma `new Tipo[tamanho]`.
- IV** Quando uma página HTML é carregada pelo browser, cria-se uma representação dos elementos desta página na forma de uma árvore de objetos denominada DOM. Esses objetos podem ser acessados e modificados mesmo após a página estar renderizada e estar visível para o usuário.

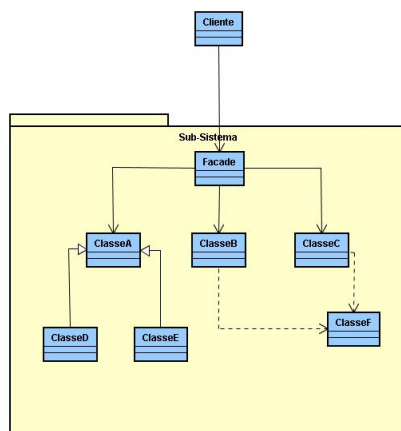
Estão corretas as afirmativas:

- a) II e III
- b) I e III
- c) II e IV
- d) I e IV

15. Em relação aos padrões de projeto do GoF, qual das alternativas abaixo apresentam apenas padrões comportamentais:

- a) Command, Iterator e Facade
- b) State, Visitor e Singleton
- c) Memento, Observer e Strategy
- d) Builder, Proxy e Chain of Responsibility

16. O diagrama a seguir ilustra o uso do padrão de projeto FACADE. Sobre tal padrão, marque a alternativa correta:



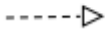



- a) Torna o sistema mais fácil de se usar, protegendo os clientes dos componentes do sistema, reduzindo o número de objetos que terão que lidar diretamente.
- b) Promove forte acoplamento entre os subsistemas e seus clientes

- c) Impede que as aplicações possam acessar as subclasses diretamente.
- d) Nenhuma das alternativas anteriores é correta.

17. A arquitetura da plataforma JAVA EE provê uma divisão tanto lógica quanto física de aplicações em camadas. Aplicações Java EE distribuídas são comumente compostas de uma camada cliente, que implementa a interface com o usuário; uma (ou algumas) camadas intermediárias, que processam a lógica do negócio e proveem serviços à camada cliente; e uma camada formada por sistemas legados. Em ambientes JAVA EE, essa ultima camada é também chamada de, chamada de:

- a) Enterprise Information System - EIS.
- b) Enterprise Service Bus - ESB.
- c) Java Persistence Tier - JPT.
- d) Java Database and Legacy Systems - JDLS

18. Considerando os tipos de relacionamentos existentes na UML (*Unified Modeling Language*), correlacione as colunas da figura a seguir:

Notação	Relacionamento
I. 	() Agregação
II. 	() Dependência
III. 	() Generalização
IV. 	() Realização

Qual a alternativa que apresenta a sequencia correta em ordem, de cima para baixo?

- a) I, II, IV, III.
- b) III, II, IV, I.
- c) III, I, IV, II.
- d) IV, II, III, I.

19. Analise as afirmativas abaixo sobre a estrutura de aplicações Android.

- I. Uma *Task* corresponde a uma unidade de interação do usuário ou uma unidade de execução.**
- II. Um *Intent* representa uma descrição abstrata de uma operação que uma atividade requer que outra desempenhe.**
- III. Uma *Activity* representa uma cadeia de atividades que podem se estender por mais de um aplicativo.**

Está(ão) CORRETA(S) a(s) afirmativa(s):

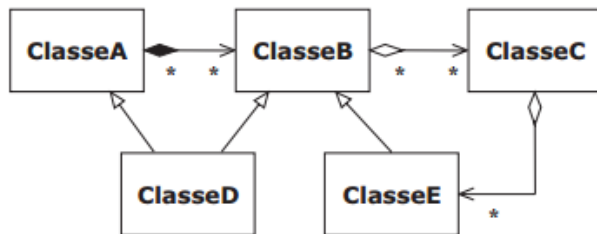
- a) II apenas.
- b) I e II, apenas.
- c) I e III, apenas.
- d) I, II e III.

20. Qual o diagrama da UML que define o comportamento do sistema através dos fluxos conduzidos por processamentos e/ou etapas do processo de negócio? Tal diagrama pode conter, a partir da versão 2.0 da UML, raias de natação para melhor definir quem é o ator responsável por cada etapa de cada fluxo.

- a) comunicação

- b) estados
- c) casos de uso
- d) atividades

21. Analise o diagrama de classes a seguir e marque a alternativa correta:



- a) A associação entre as classes *ClasseB* e *ClasseC* se chama composição, de modo que não obrigatoriamente deverão existir objetos da *ClasseC* que compõe a um objeto da *ClasseB*.
- b) A associação entre as classes *ClasseB* e *ClasseC* se chama agregação, de modo que não obrigatoriamente deverão existir objetos da *ClasseC* que compõe a um objeto da *ClasseB*.
- c) A herança múltipla (como exemplificada pela *ClasseD*) não é permitida na UML.
- d) Todas as alternativas estão erradas.

22. Com relação aos estilos arquiteturais de software, marque a alternativa correta:

- a) Arquitetura em camadas - Ajuda a estruturar aplicações que podem ser decompostas em grupos de sub tarefas no qual cada grupo de sub tarefas é um nível particular de abstração. O uso dessa arquitetura tem como vantagem a maior eficiência devido aos diversos níveis de abstrações.
- b) Arquitetura MVC – divide os componentes em três categorias, sendo elas: *Model*, *View* e *Controller*. Uma das responsabilidades do *Controller* é encapsular o estado da aplicação.
- c) Alguns dos benefícios da arquitetura em camadas são: a possibilidade de reuso de camadas e as dependências podem mais facilmente serem organizadas, evitando que fiquem espalhadas pelo código.
- d) Todas as alternativas estão erradas.

23. O padrão de projeto *Observer* pode ser aplicado quando:

- a) quando um objeto tem dois aspectos independentes um do outro. Encapsulando-se esses aspectos num único objeto, fica possível variá-los e reutilizá-los independentemente.
- b) quando um objeto deveria ser capaz de notificar outros objetos sem fazer hipóteses ou usar informações de quem são esses objetos. Em outras palavras, não se quer que esses objetos sejam fortemente acoplados.
- c) quando uma mudança em um objeto não exige mudanças em outros, tendo como consequência um forte acoplamento entre esses objetos.
- d) quando uma mudança em um objeto afeta todos os outros objetos independentes, de maneira que não é possível notificá-los e, dessa forma, não é possível também reutilizá-los.

24. Considere os dois itens a seguir e marque a opção que, respectivamente, apresenta dois dos componentes fundamentais do Java EE:

I. uma arquitetura gerenciada de componente do lado do servidor utilizada para encapsular a lógica de negócios de uma aplicação. Essa tecnologia permite o

desenvolvimento rápido e simplificado de aplicações distribuídas, transacionais, seguras e portáteis baseadas na tecnologia Java.

II. um framework que permite aos desenvolvedores gerenciar os dados utilizando o mapeamento relacional de objetos em aplicações construídas na plataforma Java.

Esses componentes são denominados, respectivamente:

- a) JPA (Java PersistenceAPI) e JSP (Java Server Pages).
- b) JSP (Java Server Pages) e JTA (Java Transaction API).
- c) JTA (Java Transaction API) e EJB (Enterprise JavaBeans).
- d) EJB (Enterprise JavaBeans) e JPA (Java PersistenceAPI).

25. Avalie as afirmações a seguir e marque a alternativa correta:

I. O uso de objetos em POO facilita o desenvolvimento de software, pois os objetos são divididos em propriedades e métodos, que representam, respectivamente, suas características e funções.

II. O encapsulamento é um mecanismo utilizado para impedir o acesso direto ao estado de um objeto, restando apenas os métodos públicos, que podem alterar esses estados.

III. A abstração é a propriedade do POO que permite a alteração do funcionamento interno de um método herdado de um objeto-pai.

- a) Todas as afirmações estão corretas.
- b) Apenas a afirmação *III.* está correta.
- c) Apenas as afirmações *I.* e *II.* estão corretas
- d) Apenas as afirmações *I.* e *III.* estão corretas

26. Avalie as afirmações a seguir sobre normalização em Banco de Dados e marque a alternativa correta:

I. A 1FN diz que as tabelas de um banco de dados não podem ter valores repetidos e nem atributos possuindo mais de um valor.

II. A 2FN define que os atributos normais, ou seja, os não chave, devem depender unicamente da chave primária da tabela.

III. A 3FN define que todos os atributos dessa tabela devem ser funcionalmente independentes uns dos outros, ao mesmo tempo que devem ser dependentes exclusivamente da chave primária da tabela.

- a) Todas as afirmações estão corretas.
- b) Apenas a afirmação *III.* está correta.
- c) Apenas as afirmações *I.* e *II.* estão corretas
- d) Apenas as afirmações *I.* e *III.* estão corretas

27. A arquitetura cliente-servidor é uma estrutura de aplicação distribuída em que as tarefas e cargas de trabalho estão distribuídas entre os fornecedores de recursos ou serviços, designados como servidores, e os requerentes dos serviços, designados como clientes. Marque a alternativa correta sobre o modelo cliente-servidor:

- a) Em uma arquitetura cliente-servidor, a aplicação utiliza comunicação direta entre duplas de servidores (hospedeiros) conectados alternadamente, denominados pares-clientes, tal como ocorre nas aplicações web, nas quais um servidor web fornece dados para vários pares.
- b) Na arquitetura cliente-servidor, as soluções são divididas de forma que, no servidor, fiquem todas as informações, os dados, as aplicações de transações com o banco de dados e a aplicação do usuário; e, no cliente, fique um terminal para interação com o usuário sem processamento
- c) A arquitetura cliente-servidor pode ser projetada para trabalhar em multicamadas distribuindo o processamento de aplicação entre vários servidores; além de serem escaláveis, os sistemas cliente-servidor multicamadas são voltados para aplicações nas quais os dados e a aplicação são voláteis e integrados a dados de várias fontes.
- d) Todas as alternativas são incorretas.

28. Quanto aos *design patterns*, é correto afirmar que:

- a) São projetos concretos que incluem detalhes de implementação.
- b) São conjuntos de classes abstratas e concretas que podem ser adaptadas e ampliadas para criar sistemas de aplicações.
- c) A UML pode ser utilizada para representar frameworks, mas não é tão ampla ao ponto de oferecer suporte à representação dos *design patterns*.
- d) Descrevem melhores práticas, bons projetos, e captam a experiência de uma maneira possível de ser usada por outros.

29. As propriedades que uma transação de banco de dados deve possuir são:

- a) Atualização, Concorrência, Invariabilidade, Durabilidade.
- b) Atomicidade, Concorrência, Independência, Delegação.
- c) Atualização, Coesão, Inserção, Deleção.
- d) Atomicidade, Consistência, Isolamento, Durabilidade.

30. Em uma aplicação Android, que utiliza um banco de dados SQLite, se deseja acessar os métodos *delete*, *update*, *insert*, *query* e *execSQL* para executar operações no banco de dados. Para acessar estes métodos será necessária uma variável da classe.

- a) SQLiteStatement
- b) SQLiteDatabase
- c) SQLiteExecutor
- d) SQLiteDatabaseManager

Boa prova!