Concurso Público

Programador JAVA

Caderno de Questões
Prova Objetiva

2015

SRH SUPERINTENDÊNCIA DE RECURSOS HUMANOS DA UERJ

Cópia Internet - http://concursos.srh.uerj.br/

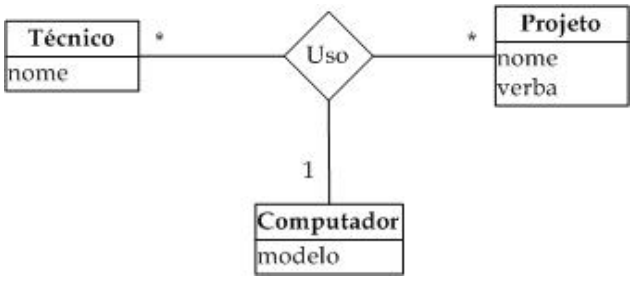
01|

O modelo de referência *Open Systems Interconnection* (ISO OSI) tem sete camadas. A camada que trata da transmissão de bits brutos por um canal de comunicação, garantindo que, quando um lado enviar um bit 1, o outro lado o receberá como um bit 1, não como um bit 0, é chamada de:

a) camada física
b) camada de rede
c) camada de transporte
d) camada de enlace de dados

02|

A associação entre as classes representada na figura abaixo é do tipo:



a) composição
b) agregação
c) reflexiva
d) ternária

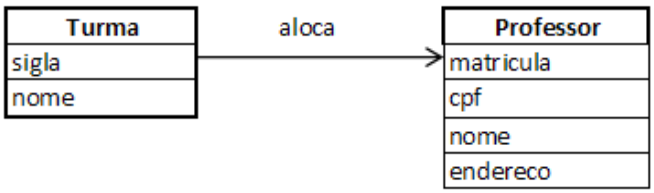
03|

Dentre as formas de abstração utilizadas na orientação a objetos, aquela que indica a capacidade de abstrair várias implementações diferentes na mesma interface é:

a) composição
b) polimorfismo
c) generalização
d) encapsulamento

04|

Considerando o conjunto de classes e associações apresentados abaixo, o modelo relacional que o representa é:



a) Turma (id, sigla, nome, idProfessor)
Professor (id, matricula, cpf, nome)
b) Turma (sigla, nome, idProfessor)
Professor (matricula, nome, endereco)
c) Turma (sigla, nome)
Professor (matricula, cpf, nome, endereco)
d) Turma (id, sigla, nome, idProfessor)
Professor (id, matricula, cpf, nome, endereco)

05|

A banda de transmissão via satélite que apresenta como principal problema a interferência terrestre é:

a) S
b) L
c) C
d) Ka

Coda - http://concursos.srh.uerj.br/

06|

Na terminologia da UML, qualquer elemento externo ao sistema que interage com ele mesmo é, por definição, denominado de:

a) ator
b) cenário
c) caso de uso
d) relacionamento

07|

O DBA, como um dos usuários do ambienhte de banco de dados, interage com as seguintes interfaces:

a) consulta interativa e instruções DDL
b) instruções DDL e comandos privilegiados
c) programas de aplicação e consulta interativa
d) comandos privilegiados e programas de aplicação

08|

Considerando as diversas abordagens para a prevenção de impasses, a abordagem adequada a ser adotada para a condição de espera circular é:

a) tirar os recursos
b) fazer *spool* de tudo
c) ordenar os recursos numericamente
d) solicitar todos os recursos inicialmente

09|

Considerando as definições do padrão XHTML, deve-se respeitar a seguinte característica na sua implementação:

a) os atributos devem estar contidos entre colchetes
b) todo documento deve especificar seu tipo de documento
c) as *tags* de fechamento são obrigatórias, exceto para </p>
d) todas as *tags* e atributos devem estar em letras maiúsculas

10|

Nos cabos de fibra ótica, dois tipos de fontes de luz são geralmente utilizados para fazer a sinalização: os LEDs e os lasers semicondutores. Em relação aos LEDs, sua taxa de dados, distância e duração são, respectivamente:

a) alta / curta / curta
b) alta / longa / longa
c) baixa / longa / curta
d) baixa / curta / longa

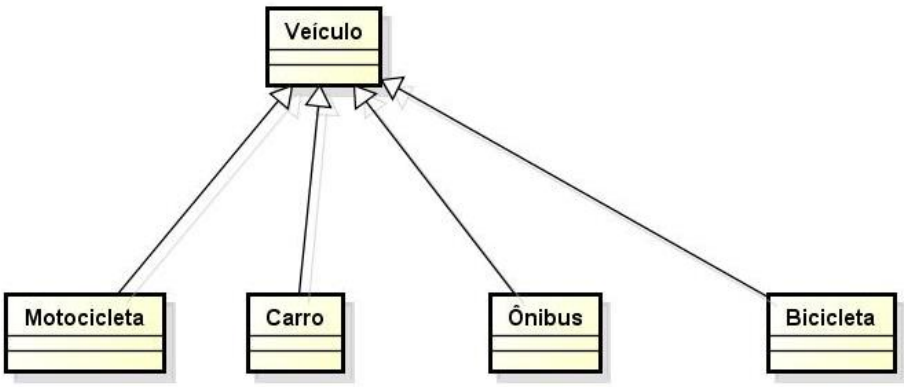
11|

O estado de um processo, no qual mesmo que a CPU não tenha nada a fazer, o processo não pode executar é:

a) pronto
b) bloqueado
c) executando
d) aguardando

Cópia Internet: http://concursos.srh.uerj.br/

12|
A relação entre a classe Veículo e as demais classes do diagrama abaixo expressa o seguinte conceito do paradigma de orientação a objeto:



a) herança
b) capacitação
c) polimorfismo
d) encapsulamento

13|
A cláusula utilizada no Diagrama de Transição de Estados da UML para declarar ações que são executadas sempre que o objeto sai de um estado é:

a) go
b) do
c) exit
d) entry

14|
Diversos diagramas são utilizados na UML para construir modelos de várias perspectivas do sistema. Dois exemplos de Diagramas Comportamentais utilizados na UML são:

a) diagrama de componentes e diagrama de classes
b) diagrama de atividades e diagrama de componentes
c) diagrama de transição de estados e diagrama de classes
d) diagrama de atividades e diagrama de transição de estados

15|
O algoritmo requer que o *software* mantenha 1 bit: o bit da direção atual, UP ou DOWN. Quando uma solicitação termina, o *driver* do disco verifica o bit. Se é UP, o braço é movido para a próxima solicitação acima. Se nenhuma solicitação está pendente em posições mais altas, o bit de direção é invertido. Quando o bit é configurado para DOWN, o movimento do braço é para a próxima solicitação abaixo, se houver.

Essa descrição corresponde ao algoritmo conhecido como:

a) elevador
b) busca mais curta primeiro
c) primeiro a entrar, último a sair
d) primeiro a entrar, primeiro a sair

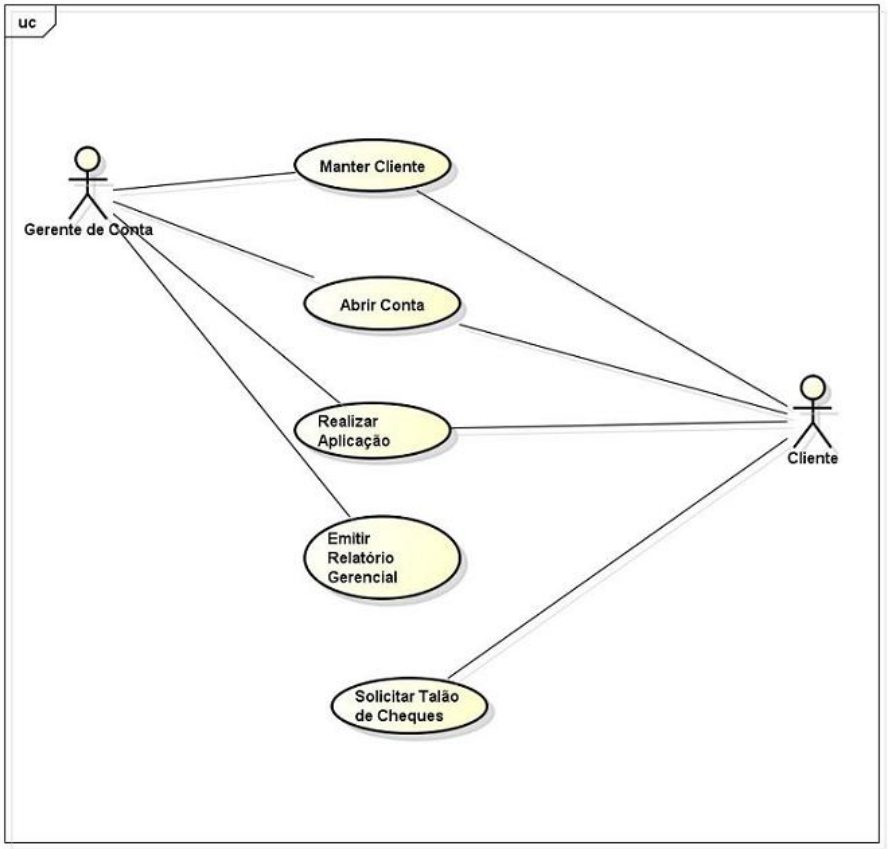
16|
Um banco de dados, em geral, possui grupos de entidades que são semelhantes. O termo que define uma coleção (ou conjunto) de entidades que tem os mesmos atributos é:

a) conjunto de entidades
b) tipo de entidade
c) atributo-chave
d) valor NULL

Cópia internet: http://concursos.srh.uerj.br/

17|

Em relação ao diagrama de casos de uso, abaixo, verifica-se que os gerentes de conta podem utilizar, com exclusividade no sistema bancário, a seguinte funcionalidade:



a) abrir conta
b) manter cliente
c) emitir relatório gerencial
d) solicitar talão de cheques

18|

No diagrama de atividade da UML, o elemento que recebe uma transação de entrada e cria dois ou mais fluxos de controle paralelos é:

a) ponto de união
b) barra de junção
c) barra de bifurcação
d) ponto de ramificação

19|

Observe a relação de itens abaixo, referente à modelagem de casos de uso de um sistema acadêmico:

- As turmas não podem ter mais que 30 alunos matriculados.
- Um aluno poderá se inscrever em, no máximo, oito disciplinas por semestre.
- O número total de créditos que o aluno cursará em um semestre não poderá ser superior a 30 créditos.
- Um professor não poderá ser alocado em mais de três disciplinas distintas no semestre.
- Um professor não poderá ser alocado em mais de três Unidades da Instituição de Ensino em um semestre.

Essa relação corresponde à seguinte documentação suplementar ao modelo de casos de uso:

a) regras de negócio
b) requisitos de interface
c) requisitos não funcionais
d) requisitos de desempenho

Cópia Internet - http://www.srh.uerj.br/

20|

O tempo para realizar a transferência de um setor em um disco rígido pode ser medido em:

a) milissegundos
b) picossegundos
c) nanossegundos
d) microssegundos

21|

Um trecho de uma página *Java ServerPages* (JSP) é apresentado a seguir. Apenas as partes relevantes são apresentadas (as reticências indicam partes deliberadamente omitidas). O propósito da página é listar, para cada produto, seu nome e o nome de sua categoria. Além disso, ao fim da lista de produtos, deve ser exibido o somatório dos valores dos produtos da lista.

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
…
<c:set var="total" value="0" scope="page" />
<c:forEach items="${requestScope.produtos}" var="produto" >
        <tr>
                <td><c:out value="${produto.nome}" /></td>
                <td><c:out value="${produto.categoria.nome}" /></td>
                <c:set var="total" value="${total + produto.valor}" />
        </tr>
</c:forEach>
...
```

Para apresentar o somatório dos valores dos produtos, posicionado após a lista na página HTML resultante, o programador deve usar a seguinte expressão:

a) <fmt:formatNumber type="currency" value="total" />
b) <fmt:formatNumber type="currency" value="${total}" />
c) <fmt:formatNumber type="currency" value="${page.total}" />
d) <fmt:formatNumber type="currency" value="${produto.total}" />

22|

Em aplicações WEB desenvolvidas em Java, é comum o uso do protocolo de interação denominado requisição-resposta, no qual diversos componentes interagem para produzir a resposta correspondente a uma requisição.

Quando uma requisição direcionada a uma *servlet* X chega no servidor WEB, o componente responsável por instanciar objetos das classes javax.servlet.http.HttpServletRequest e javax.servlet.http.HttpServletResponse é:

a) a *servlet* X
b) o servidor WEB
c) o contêiner WEB
d) o objeto da classe ServletConfig

Cópia Internet - http://concursos.srh.uerj.br/

23|

A especificação EJB define três tipos de componentes: *session beans*, *entity beans* e *message-driven beans*. Nesse contexto, considere as duas listas a seguir.

Tipos de *beans*	Características
1. *Entity Beans*	I. Implementação da persistência de dados da aplicação
2. *Message-Driven Beans*	II. Processamento de lógica de negócios, processos e fluxo de trabalho
3. *Session Beans*	III. Recebimento de dados assíncronos provenientes de outros sistemas

A correta associação dos tipos de *beans* às respectivas características é:

a) 1-I; 2-III; 3-II
b) 1-I; 2-II; 3-III
c) 1-II; 2-I; 3-III
d) 1-III; 2-II; 3-I

24|

Considere a classe em Java apresentada a seguir.

```
public class SwitchTeste {
    public static void main(String[] args) {
        int valor = 1;
        switch (valor) {
        case 0:
            System.out.print("X");
            break;
        case 1:
            System.out.print("P");
        case 2:
            System.out.print("T");
        default:
            System.out.print("O");
        }
    }
}
```

A correta execução da classe *SwitchTeste* resulta na impressão do seguinte valor:

a) P
b) PO
c) PT
d) PTO

25|

Durante a transformação de um arquivo com código fonte em Java (extensão .java) em um arquivo de *bytecodes* (extensão .class) pela JVM (Java Virtual Machine), a atribuição da atividade denominada compilação JIT é:

a) verificar que todos os *bytecodes* são sintática e semanticamente válidos
b) carregar o arquivo de *bytecodes* da memória secundária para a memória principal
c) localizar trechos de *bytecodes* frequentemente executados com vista à otimização da execução
d) verificar que todos os *bytecodes* estão de acordo com restrições de segurança predeterminadas

Copia Interna ... rsos.srh.uerj.br/

26|

Considere que uma aplicação denominada ACME, construída com a tecnologia JSP, possui a funcionalidade de cadastro de empresas. Essa funcionalidade permite a inclusão, exclusão, alteração e visualização de registros com informações sobre empresas. Considere ainda que essa aplicação contém duas páginas JSP, listadas a seguir:

- ACME/srh/DetalhesEmpresa.jsp
- ACME/srh/Empresas.jsp

A partir da página DetalhesEmpresa.jsp, é necessário fazer o despacho (*dispatch*) para a página Empresas.jsp.

Dentre as opções abaixo, aquela que apresenta a expressão a ser usada para obter o objeto da classe javax.servlet.RequestDispatcher necessário para fazer o despacho acima descrito é:

a) servlet.getRequestDispatcher("/srh/Empresas.jsp")
b) request.getRequestDispatcher("/srh/Empresas.jsp")
c) servlet.request.getRequestDispatcher("/srh/Empresas.jsp")
d) getServletContext().request.getRequestDispatcher("/srh/Empresas.jsp")

27|

Em uma aplicação que utiliza *Java Server Faces* (JSF), um programador está implementando uma funcionalidade que lista todos os municípios brasileiros. Visto que essa lista é muito grande, por questões de desempenho, esse programador decide manter uma única instância dela no servidor, de tal forma que todos os usuários da aplicação, que está sendo desenvolvida, façam acesso a essa mesma instância.

Nesse cenário, a anotação que define o escopo mais adequado para usar na classe que irá manter a lista dos municípios é:

a) javax.faces.bean.ApplicationScoped
b) javax.faces.bean.RequestScoped
c) javax.faces.bean.SessionScoped
d) javax.faces.bean.UnifiedScoped

28|

Considere a classe em Java apresentada a seguir.

```
public class ExcecaoTeste {
    public static void main(String[] args) {
        System.out.println(outroMetodo("5") + outroMetodo("0"));
    }

    private static String outroMetodo(String v) {
        String r = v;
        try {
            umMetodo(v);
        } catch (IllegalArgumentException e) {
            r = e.getMessage();
        } catch (Exception e) {
            return "2";
        } finally {
            if (r.equals(v)) {
                r = "3";
            }
        }
        return r;
    }

    private static void umMetodo(String v) {
        if (v.equals("0"))
            throw new IllegalArgumentException("4");
    }
}
```

A correta execução da classe ExcecaoTeste resulta na impressão do seguinte valor:

a) 30
b) 33
c) 34
d) 50

29|
Dentre as opções abaixo, aquela que apresenta o padrão do catálogo J2EE, que pode ser usado para minimizar o acoplamento entre a camada de apresentação e os serviços de negócio de uma aplicação, é:
a) *data access object*
b) *business delegate*
c) *transfer object*
d) *view helper*

30|
Os acrônimos JAXP, JNDI e JTA são siglas de especificações existentes na plataforma Java.

Especificações	Capacidade
1. JAXP	I. Permite vincular um objeto a um nome
2. JNDI	II. Permite transações distribuídas sobre vários recursos.
3. JTA	III. Permite a validação de documentos em formato semiestruturado.

A correta associação das especificações às respectivas capacidades é:
a) 1-I; 2-II; 3-III
b) 1-II; 2-I; 3-III
c) 1-III; 2-II; 3-I
d) 1-III; 2-I; 3-II

31|
Em uma aplicação WEB em Java, é possível definir um elemento <session-timeout> dentro de um elemento <session-config> no arquivo web.xml, com o propósito de definir a duração de cada sessão.

Para definir que cada sessão dura dois minutos, um desenvolvedor deve especificar para o elemento <session-timeout> o valor numérico igual a:
a) 2
b) 120
c) 2/60
d) 7200

32|
Considere a classe em Java apresentada a seguir.

```
public class ArrayTeste {
    public static void m(int umArray[]) {
        for (int i = 0; i < umArray.length; i += 2)
            umArray[i] *= 2;
    }

    public static void main(String[] args) {
        int array[] = { 1, 2, 3, 4, 5 };

        int m = 0;
        for (int i : array) {
            m *= i;
        }

        m(array);

        int n = 0;
        for (int i : array) {
            n += i;
        }

        System.out.println(n - m);
    }
}
```

A correta execução da classe ArrayTeste resulta na impressão do seguinte valor:
a) 0
b) 15
c) 24
d) 30

Cópia Inter... http://concursos.srh.uerj.br/

33|

O catálogo de padrões J2EE está organizado de acordo com a camada na qual cada padrão é usado.

Dentre as opções abaixo, aquela que lista padrões pertencentes, respectivamente, às camadas de apresentação, negócio e integração é:

a) *View Helper* / *Front Controller* / *Intercepting Filter*
b) *Context Object* / *View Helper* / *Data Access Object*
c) *Transfer Object* / *Intercepting Filter* / *Context Object*
d) *Front Controller* / *Transfer Object* / *Data Access Object*

34|

Considere a classe em Java apresentada a seguir.

```
class AtribuicaoTeste {
    Integer i = 0;
    static Integer j = 1;
    static final Integer k = 2;
    static Integer[] v;

    static void umMetodo() {
        k = 3;
        i = 4;
        j = new Integer(5);
        v = Integer[10];
    }
}
```

A instrução de atribuição que poderia ser definida no método umMetodo da classe AtribuicaoTeste sem gerar erro de compilação é:

a) i = 4
b) k = 3
c) v = Integer[10]
d) j = new Integer(5)

35|

Dentre as opções a seguir, aquela em que cada componente corresponde ao nome de uma interface definida na API *Java Database Connectivity* (JDBC) é

a) PreparedStatement, ResultSet, RowSet
b) Connection, ConnectedStatement, RowSet
c) ConnectedStatement, PreparedStatement, ResultSet
d) Connection, ConnectedStatement, PreparedStatement

Cópia Internet http://www.cv.uerj.br/

36|

Java Persistence API (JPA) é uma especificação para *frameworks* de mapeamento objeto-relacional. Nesse contexto, considere que em um programa em Java existam duas classes, denominadas Pessoa e Projeto. Considere ainda que a classe Pessoa contém a declaração a seguir.

```
@ManyToMany
@JoinTable(name = "PESSOA_PROJETO",
      joinColumns = { @JoinColumn(name = "C1", referencedColumnName =
"C2") },
      inverseJoinColumns = { @JoinColumn(name = "C4",
referencedColumnName = "C3") })
private List<Projeto> projetos;
```

Dentre as opções abaixo, aquela que lista corretamente as colunas de chaves estrangeiras que devem ser definidas na tabela PESSOA_PROJETO é:

a) C1 e C2
b) C1 e C4
c) C2 e C3
d) C3 e C4

37|

O pacote *Java Developers Kit* (JDK) contém um programa denominado javadoc, um gerador de documentação em formato HTML para código fonte escrito em Java. Esse programa analisa o código fonte de uma aplicação à procura de marcadores especiais (*tags*) que são empregados para indicar conteúdo a ser usado na geração da documentação em HTML.

Dentre as opções abaixo, aquela que apresenta marcadores do javadoc que se aplicam apenas a métodos é:

a) @link, @return, @since
b) @link, @param, @return
c) @param, @value, @throws
d) @deprecated, @exception, @return

38|

Desde a versão 2.0 do JSF, é possível passar valores de parâmetros em expressões que envolvem a chamada de um método. Nesse contexto, considere a classe *FormBean* apresentada a seguir, na qual as reticências indicam partes irrelevantes.

Repare que a classe está marcada com a anotação javax.faces.bean.ManagedBean.

```
@ManagedBean
public class FormBean {
    ...
    public String mover(int deslocamento) {
        ...
    }
    ...
}
```

Na página correspondente ao *bean* apresentado, uma forma correta de invocar o método mover é:

a) <h:commandButton value="Anterior" action="{formBean.mover(-1)}"/>
b) <h:commandButton value="Anterior" action="{FormBean.mover(-1)}"/>
c) <h:commandButton value="Anterior" action="#{formBean.mover(-1)}"/>
d) <h:commandButton value="Anterior" action="#{FormBean.mover(-1)}"/>

Cópia Internet - http://concursos.srh.uerj.br/

39|

Java Persistence Query Language (JPQL) é uma linguagem de consulta que faz parte da especificação JPA. Considere uma aplicação em Java que usa JPA, na qual está definida uma classe de entidade denominada br.app.acme.Cliente.

Além disso, essa aplicação contém o trecho de código abaixo, que cria um objeto do tipo javax.persistence.Query, cuja referência é *qry*.

```
javax.persistence.Query qry = entityManager.createQuery(
            "SELECT OBJECT(c) FROM br.app.acme.Cliente c " +
            " WHERE c.uf = :uf");
qry.setParameter("uf", "Rio de Janeiro");
```

A expressão adequada para execução da consulta em JPQL representada pela referência *qry* é:

a) java.util.Collection clientes = qry.getResultList()
b) java.util.Collection clientes = qry.executeQuery()
c) br.app.acme.Cliente[] clientes = qry.getResultList()
d) java.util.Collection clientes = qry.getSingleResult()

40|

Considere a interface I e as classes X e Y apresentadas a seguir, todas implementadas em Java e localizadas em um mesmo pacote.

```
interface I {
    void m(double a, double b);
}
```

```
class X implements I {
    public void m(double a, double b) {
        System.out.println((int) (a - b));
    }
}
```

```
class Y extends X implements I {
    I ref = new X();

    public void m(double a, double b) {
        System.out.println(a + b);
    }
}
```

Dentre as opções a seguir, aquela que apresenta uma expressão sintaticamente correta é

a) (new I()).ref.m(0.20, 0.15)
b) (new X()).ref.m(0.20, 0.15)
c) (new Y()).ref.m(0.20, 0.15)
d) ((I)(new Y())).ref.m(0.20, 0.15)

Cópia Interna - concursos.srh.uerj.br/