Teste Técnico - Respostas

Lucas do Nascimento Silva

JAVA

1. Ocorre um erro de quando tenta acessar uma posição inexistente no vetor.

**if** (!printed) {

System.*out*.println("Talvez");

}

1. String.
2. Ocorre uma sobrescrita de métodos.
3. Entra em Loop.

A:

**public** **void** executaX(){

**new** Thread() {

    @Override

**public** **void** run() {

**while** (**true**) {

    System.out.println("Continue a nadar");

    }

  }.start();

  }

}

B:

**public** **class** Test **extends** Thread

{

**public** **static** **void** main(String[] args)

    {

        Thread t = Thread.*currentThread*();

        System.***out***.println("Current thread: " + t.getName());

        t.setName("Geeks");

        System.***out***.println("After name change: " + t.getName());

        System.***out***.println("Main thread priority: "+ t.getPriority());

        t.setPriority(***MAX\_PRIORITY***);

        t.start();

    }

}

public class VerficadorNumeroPrimo {

public static void main(String[] args) {

    Random r=new Random();

    int a=r.nextInt();

    if (a%2==0) {

        System.out.println("numero "+a+" nao é primo");

    }

    else {

        int b=a;

        int c=b+20;

        while (b<c){

            if (a%b==0 && a%a==0){

                System.out.println("numero "+a+" é primo");

                break;

          }

          else {

              b=b+1;

          }

      }

    if (b>=c){

        System.out.println("numero "+a+" nao é primo");

    }

SQL

1. SELECT prod.dt\_vencimento as dataVencimento, prod.Nm\_produto as NomeProduto, op.vl\_Principal as ValorPrincipal FROM Produto prod, Operação op WHERE BETWEEN : dt\_abertura AND : dt\_vencimento AND dt\_abertura >= ’01/01/2019’ AND dt\_vencimento <= ’31/01/2019’ GROUP BY op.vl\_principal, prod.dt\_vencimento, prod.Nm\_produto ORDER BY op.dt\_abertura DESC
2. SELECT \* FROM Produto as prod

Left JOIN operacao on prod.id\_produto = id\_produto where Ic\_ativo = false

1. SELECT \* FROM Produto as prod

Inner JOIN operacao on prod.id\_produto = id\_produto GROUP BY sum(id\_produto)

1. Os atributos das entidades, creio que recebam a tipagem correta, poderíamos incluir um campo de idOperacao na tabela de produto e assim teríamos a relação de quais produtos pertecem a determinada operação de forma mais fácil. A cardinalidade do idProduto para seja 1 para N. N produtos fazem parte de uma operação. E as chaves primárias ficariam por parte de idProduto e idOperacao, na tabela Operacao idProduto seria chave estrangeira.

EJB

1. É uma camada da aplicação que recebe requisições e respostas. Onde nessa camada encapsula a lógica de negócio da aplicação.
2. O que realiza a execução do EJB é um container alocado no servidor de aplicação.
3. CMP você tem mais controle quanto ao banco de dados, o BMP gerencia todas as consultas insert, select e update. Tudo sendo manipulado pelo EJB. Enquanto o BMP, você tem que realizar as conexões e a busca dos dados manualmente.
4. Por referencia.
5. É um facilitador de consumo de mensagens JMS, o que facilita a integração entre sistemas.
6. Enquanto o StateFul mantem um relacionamento de um para um entre o lado cliente com a instancia do bean e os mantem na memoria ate o cliente utiliza-lo ou acontecer um timeout, o Stateless suporta N clientes e tem uma gestão mais eficiente no uso de memória e não há controle de sessão.
7. O EJB, cria uma série de instancias acionadas por mensagens. Se o bean acionado por mensagens utilizar injeção de dependência, o ejb injetara essas referencias antes de instanciar. O container chamará o método com a anotação @PostConstruct. Um bean que não possui sessão sem estado não é passivo, sendo assim tem dois estados. Inexistente e pronto para receber mensagem. Ao final do ciclo de vida o container chama o método com a anotação @PreDestroy, que será coletada pelo garbage collector.
8. O ciclo de vida se inicia obtendo uma referencia deum bean de sessão com preservação do estado. O container executa qualquer injeção de dependência em seguida chama o método anotado com @PostConstruct. Feito isso, o bean está pronto para atender os métodos de negócios.
9. O ciclo de vida se inicia quando o container executa qualquer injeção de dependência e invoca o método anotado @PostConstruct, caso exista. O bean agora está pronto para ter seus métodos invocados.
10. O ciclo de vida do Entity bean se inicia quando o servidor inicia pela primeira vez e varias instancias de bean são criadas e colocados em pool. A instancia não está vinculada a dados específicos, não há um vinculo com tabelas do banco. Assim que vinculado com a anotação @Entity, o bean pertence a um objeto de negócios e uma tabela.

Hibernate / JPA

1. JPA é uma API que descreve uma interface para frameworks de persitencia de dados. O JPA define um mapeamento objeto-relacional para objetos java. POJO, que são as beans das entidades. Hibernate é um framework para mapeamento objeto-relacional.
2. Quando um relacionamento recebe a anotação @OneToMany ou @ManyToMany,ele e recebe o status de Lazy, é que qualquer tipo de busca na entidade a entidade fraca não será carregado junto, faz-se necessário uma ação direta do desenvolvedor para obter os dados. Quando um relacionamento recebe a anotação OneToMany ou @ManyToMany,ele é carregado de forma EAGER, ele trará todos os dados das entidades relacionados junto e o JPA executará tudo numa só query.

A-Select cliente from Cliente cli, Endereco endereco where cli.idEndereco = :endereco.idEndereco;

B- Select cliente from Cliente cli, Telefone tel where cli.idTelefone = :tel.idTelefeno and tel.descricao = :‘9’;

C- Select endereco From Endereco end, Cliente cli where end.idEndereco = : cli.idCliente and sum(cli.idEndereco)