



Teste de Conhecimento

avalie sua aprendizagem

ARQUITETURA DE SISTEMAS
CCTE168_AB_20221110137_V1

Lupa Calc.

Aluno: DOUGLAS MATOS DA SILVA
Disc.: ARQ. SISTEMAS

Mat.: 202101110137
2022.1 EAO IGJ / EX

Prezado (a) Aluno(a),

Você terá agora seu **TESTE DE CONHECIMENTO**. Lembre-se que este exercício é opcional, mas não valerá ponto para sua avaliação. O mesmo será composto de questões de múltipla escolha.

Após responder cada questão, você terá acesso ao gabarito comentado e/ou à explicação da mesma. Aproveite para se familiarizar com este modelo de questões que será usado na sua AV e AVS.

1. Em relação ao Framework CCM (CORBA Component Model) podemos afirmar que:

- ☐ O modelo de instalação define o ambiente de execução para as instâncias do componente.
- ☒ O modelo de programação é Composto pela CIDL (Component Implementation Definition Language) e pelo CIF (Component Implementation Framework).
- ☐ O nível estendido prevê uma forma simplificada de distribuir um objeto CORBA como componente.
- ☐ O nível básico prevê um conjunto maior de ações, como as portas de comunicação que representam os elementos de conexão entre os componentes.
- ☐ O modelo abstrato especifica como os componentes e suas implementações devem ser empacotados.

Explicação: O nível básico prevê uma forma simplificada de distribuir um objeto CORBA como componente. O nível estendido prevê um conjunto maior de ações, como as portas de comunicação que representam os elementos de conexão entre os componentes. O modelo de empacotamento especifica como os componentes e suas implementações devem ser empacotados. O modelo de execução define o ambiente de execução para as instâncias do componente.

2. O gerenciamento do ciclo de vida dos componentes de servidor é feito através de políticas que controlam o momento de ativação/desativação dos componentes. Quem é responsável pelo container ativar o componente quando for feita a primeira chamada a alguma de suas operações, e desativa quando explicitamente requisitado pela aplicação, deslocando a memória utilizada pelo componente?

- ☐ Session
- ☒ Component
- ☐ Transaction
- ☐ Method
- ☐ Service

Explicação: Method - Ativação/desativação a cada chamada de método, limitando o uso de memória ao tempo de duração da operação, mas acrescentando o custo de ativação e desativação do componente. Transaction - Ativação/desativação a cada transação. Memória permanece alocada durante a transação. Session e Service não fazem parte do gerenciamento do ciclo de vida dos componentes de servidor.

3. O gerenciamento do ciclo de vida dos componentes de servidor é feito através de políticas que controlam o momento de ativação/desativação dos componentes. Associe a sentença abaixo ao respectivo conceito.

" O container ativa o componente, quando for feita a primeira chamada a alguma de suas operações, e desativa, quando explicitamente requisitado pela aplicação, deslocando a memória utilizada pelo componente."

- ☒ Component
- ☐ Method
- ☐ Skeletons
- ☐ Container
- ☐ Transaction

Explicação:

Method: Ativação/desativação a cada chamada de método, limitando o uso de memória ao tempo de duração da operação, mas acrescentando o custo de ativação e desativação do componente.

Transaction: Ativação/desativação a cada transação. Memória permanece alocada durante a transação. Container:

O container ativa o componente, quando for feita a primeira chamada a alguma de suas operações, e desativa, quando explicitamente requisitado pela aplicação, deslocando a memória utilizada pelo componente.

4. O CCM é um framework de componentes do lado do servidor, cuja finalidade é facilitar o desenvolvimento e a instalação de aplicações distribuídas que utilizam a arquitetura de sistemas por componentes. Dentre os tipos de modelos podemos destacar:

- ☐ Modelo Abstrato, Modelo de Programação, Modelo de Padrões, Modelo de Testes e Modelo de Execução.
- ☒ Modelo Abstrato, Modelo de Programação, Modelo de Empacotamento, Modelo de Instalação e Modelo de Execução.
- ☐ Modelo Abstrato, Modelo de Análise, Modelo de Projeto, Modelo de Instalação e Modelo de Execução.
- ☐ Modelo de Negócio, Modelos de Projeto, Modelo de Testes e Modelo de Implementação.
- ☐ Modelo de Negócio, Modelo de Projeto, Modelo de Padrões e Modelo de Implementação.

Explicação:

Os cinco tipos de modelos são: Modelo Abstrato (Define os atributos, portas de comunicação e home dos componentes), Modelo de Programação (Composto pela CIDL (Component Implementation Definition Language) e pelo CIF (Component Implementation Framework)), Modelo de Empacotamento (Especifica como os componentes e suas implementações devem ser empacotados), Modelo de Instalação (Define um mecanismo padrão para a instalação de aplicações) e Modelo de Execução (Define o ambiente de execução para as instâncias do componente).

5. Os frameworks são os mais indicados para fornecer uma base mais sólida para a próxima geração de aplicativos baseados em componentes distribuídos, em escala empresarial, avalie se as afirmativas a seguir são verdadeiras (V) ou falsas (F):

- () O Microsoft COM+ possui o Windows como dependência de plataforma, mas não possui nenhuma dependência de Linguagem.
- () O Enterprise JavaBeans (EJB) possui tanto dependência de plataforma quanto dependência de Linguagem.
- () O Enterprise JavaBeans (EJB) possui a Linguagem Java como dependência de linguagem.

As afirmativas são respectivamente:

- ☐ F, F e V.
- ☐ F, V e V.
- ☐ V, V e V.
- ☐ V, F e F.
- ☒ V, F e V.

Explicação: O Enterprise JavaBeans (EJB) não possui tanto dependência de plataforma, mas possui dependência de Linguagem.

6. Em relação ao provimento e construção de componentes, analise as afirmativas a seguir:

I - O arquiteto de sistemas, baseado nos requisitos do novo sistema, vai executar o design da nova aplicação, identificando todos os componentes necessários e aplicando reuso aos componentes que já existirem. Somente serão construídos os componentes que não existirem.

II - Quanto mais madura a organização (empresa) no conceito de arquitetura de sistemas maior o conjunto de componentes reutilizáveis ela vai ter, e menor o conjunto de componentes a serem desenvolvidos para resolverem os problemas.

III - Um componente reutilizado é um componente que já foi testado, é um componente que não tem problemas de desenvolvimentos a serem sanados. Sendo assim, quanto mais a reutilizabilidade de código menor o custo, menor o tempo e maior a qualidade.

Assinale:

- ☐ se somente a afirmativa I estiver correta.
- ☒ se todas as afirmativas estiverem corretas.
- ☐ se somente a afirmativa II e III estiverem corretas.
- ☐ se somente a afirmativa I e II estiverem corretas.
- ☐ se somente as afirmativas I e II estiverem corretas.

Explicação: Todas as afirmativas são verdadeiras.

Col@bore

Antes de finalizar, clique aqui para dar a sua opinião sobre as questões deste exercício.

Sugira! Sinalize! Construa!

