

Sistemas Distribuídos

Painel / Meus cursos / Departamento de Ciência da Computação / Tecnologia em Análise e Desenvolvimento de Sistemas / OTES11-TADS / Trabalhos / Prova 2 2023-1

Iniciado em	Wednesday, 21 Jun 2023, 19:06
Estado	Finalizada
Concluída em	Wednesday, 21 Jun 2023, 20:31
Tempo empregado	1 hora 24 minutos
Avaliar	6,80 de um máximo de 10,00(68%)

Questão 1

Correto

Atingiu 1,00 de 1,00

🚩 Marcar questão

Considerando transparência de localização para comunicação interprocessos envolvendo RPC e/ou RMI, assinale V (verdadeiro) ou F (falso)

- ☐ V

☒ F
- Atingiu 1,00 de 1,00
- A resposta correta é: F
- a) Normalmente necessita de sintaxe especial para funcionar
- ☒ V

☐ F
- Atingiu 1,00 de 1,00
- A resposta correta é: V
- b) Implementada pelo middleware de comunicação interprocessos
- ☒ V

☐ F
- Atingiu 1,00 de 1,00
- A resposta correta é: V
- c) Depende da disponibilização de uma interface de serviço ou remota
- ☒ V

☐ F
- Atingiu 1,00 de 1,00
- A resposta correta é: V
- d) Facilita a invocação remota
- ☒ V

☐ F
- Atingiu 1,00 de 1,00
- A resposta correta é: V
- e) É a inovação dos paradigmas de chamada/invocação remota

Questão 2

Parcialmente correto

Atingiu 0,80 de 1,00

🚩 Marcar questão

Em se tratando de RMI e dos middlewares comentados em aula, assinale as afirmações a seguir com V para verdadeiro e F para falso.

- V

F

V

V

F
- a) A semântica de invocação remota para RMI que apresenta mecanismos de tolerância a falhas é a no máximo uma vez.
- b) As medidas de tolerância a falhas, em se tratando de semânticas RMI, são implementadas no objeto remoto.
- c) Para que a semântica RMI ao menos uma vez (at least once) funcione deve ser possível a replicação de mensagem.
- d) A semântica RMI de mais fácil implementação é a semântica talvez.
- e) Quando não previamente definida, a definição de qual semântica de invocação RMI utilizar depende da especificação da interface remota.
- a) Apenas a semântica Maybe (talvez) não apresenta mecanismos de tolerância a falhas.
- b) Na verdade as semânticas e portanto os mecanismos de tolerância a falhas são implementados no middleware RMI, servindo tanto ao cliente quanto ao objeto remoto (servidor)
- c) A semântica de invocação ao menos uma vez disponibiliza retransmissão (e consequentemente replicação) de mensagens quando necessário.
- d) É de fato a mais leve e de fácil implementação.
- e) Quando for possível escolher qual semântica de invocação RMI utilizar, deve-se observar a natureza/funcionalidade dos métodos envolvidos na interface remota, de modo que havendo apenas métodos idempotentes a semântica com melhor desempenho pode ser a talvez ou ao menos uma vez, embora aquela que de fato garanta a execução do método, para esse caso, seja a ao menos uma vez.

Questão 3

Incorreto

Atingiu 0,00 de 1,00

🚩 Marcar questão

Considerando que cada afirmação a seguir vale 1 ponto, qual é o valor do somatório das afirmações verdadeiras?

- a) Uma interface de serviço define os métodos de um objeto remoto.
- b) A IDL é usada para implementar operações de objetos remotos.
- c) É possível utilizar exclusivamente o paradigma de programação estruturada para desenvolver objetos remotos.
- d) Stubs e skeletons fazem parte do objeto remoto.
- e) Objetos remotos sempre são servidores, pois sempre provêm serviço.

Resposta: 2

A resposta correta é: 0

Questão 4

Parcialmente correto

Atingiu 0,80 de 1,00

🚩 Marcar questão

Considerando o modelo de objetos local e distribuídos, comunicação interprocessos envolvendo RPC e/ou RMI, assinale V (verdadeiro) ou F (falso).

- V

F

V

V

V
- a) RPC e RMI são extensões do modelo de invocação de método local.
- b) As chamadas e invocações remotas utilizam o mesmo protocolo de camada de aplicação.
- c) O middleware desempenha papel fundamental na comunicação interprocessos.
- d) Objetos distribuídos podem ser úteis na tolerância a falhas.
- e) O espaço de endereçamento do objeto invocado indica se o mesmo é local ou remoto.

Questão 5
Parcialmente
correto
Atingiu 0,40 de
1,00
🚩 Marcar
questão

Tendo em vista a comunicação inter-processos baseada no paradigma Publish/Subscribe, associe o conceito (resposta a pergunta o que é?) à definição que melhor se adequa:

Objeto notificação	Subscriber	❌
Subscribe	Assinar um evento	✅
Objeto de Interesse	Objeto gerador de eventos	✅
Assinante	Manifesta interesse em eventos	❌
Observer	Serviço de Eventos	❌

Sua resposta está parcialmente correta.

Você selecionou corretamente 2.

A resposta correta é:

Objeto notificação → Representação de evento,

Subscribe → Assinar um evento,

Objeto de Interesse → Objeto gerador de eventos,

Assinante → Subscriber,

Observer → Publisher and Subscriber.

Questão 6
Completo
Atingiu 1,50 de
1,50
🚩 Marcar
questão

Qual é a principal diferença entre o modelo de objetos tradicional e o modelo de objetos distribuídos? Explique.

O modelo de objetos tradicional, não consegue fazer chamadas a outros objetos de diferentes processos locais e de objetos remotos.

O modelo de objetos distribuídos, é uma extensão do modelo de objetos tradicional. Tem por objetivo, uma implementação de comunicação interprocessos por meio de chamada de métodos remotos, implementados em uma interface remota que irá disponibilizar os métodos que podem ser invocados remotamente.

A principal diferença ente os dois modelos é a área da memória manipulada e consequentemente, o local de manipulação dos objetos. Mas também temos a diferença de transparência, segurança, latência, entre outros. Enquanto no modelo de objetos distribuídos temos objetos podendo estar em diferentes computadores ou em diferentes processos locais, interagindo entre eles, invocando os métodos (devidamente autorizados remotamente por uma interface) destes objetos e recebendo apenas o resultado das operações. No modelo de objeto tradicional, somos restringidos a apenas um processo (podendo ter várias threads) manipulando uma única área de memória local.

Comentário:

ok. Só um toque para implementados de uma interface remota e não implementados em uma interface remota. Ou quiseste dizer implementados em uma referência remota?

Questão 7
Incorreto
Atingiu 0,00 de
1,00
🚩 Marcar
questão

De acordo com o exemplo de implementação de comunicação interprocessos RMI em linguagem java, disponibilizado durante as aulas, o serviço de nomes RMI pode ser invocado de formas diferentes por duas classes. Quais são essas classes? Escreva o nome de uma classe seguida de vírgula espaço em branco e o nome da outra classe.

Resposta: AplicacaoCliente.java, AplicaçãoServidora.java ❌

A resposta correta é: Naming, Registry

Questão 8
Completo
Atingiu 2,50 de
2,50
🚩 Marcar
questão

Um determinado servidor implementado como um objeto remoto Java RMI possui, entre outros, um método que lê o valor de um determinado atributo privado, e outro método que altera o valor deste mesmo atributo. Explique qual a semântica de invocação RMI é a mais adequada para cada um destes métodos, considerando que este servidor deve atender **vários clientes concorrentemente**.

Neste caso, estamos lidando com chamada do método getters(apenas leitura) e setter(alteração da variável).

Para o getter (lê o valor). Estamos apenas lendo da variável, ou seja, a operação é idempotente, não irá mudar o resultado caso a operação seja chamada novamente. Portanto, nesse caso a invocação mais adequada é a invocação ao menos uma vez. Com isso, podemos economizar memória do histórico caso fosse feita a invocação semântica no máximo uma vez, pois não vamos precisar guardar as respostas e sim, vamos executar o método novamente, mesmo se a mensagem é duplicada. Melhoramos assim, o desempenho do acesso concorrente tanto em CPU como Memória.

Para o setter (altera o valor). Estamos escrevendo na variável e, nesse caso, precisamos ter total tolerância a falhas. Portanto, precisamos da semântica de invocação no máximo uma vez, para que, mesmo que o requisitante mande uma nova mensagem duplicada para a mesma operação, apenas o resultado já calculado (estando na memória do histórico) seja enviado sem alterar o valor da variável mais de uma vez, gerando inconsistência.

Comentário:

Terminar revisão

→ Prova

Seguir para...

Como redirecionar email institucional para email mais utilizado →

Você acessou como **VICTOR EDUARDO REQUIA** (Sair)

[OTES11-TADS](#)

[Resumo de retenção de dados](#)

[Obter o aplicativo para dispositivos móveis](#)

