Programação Orientada a Objetos - COM230 - Turma 012

Página inicial
Avisos
Cronograma
Atividades
Fóruns
Calendário Lives
Collaborate
Notas

Menu das Semanas
Semana 1
Semana 2
Semana 3
Semana 4
Semana 5
Semana 6
Semana 7
Semana 8
Orientações para realização da prova
Orientações para realização do exame

Documentos e informações gerais

Gabaritos
Referências da disciplina
Repositório de REA's

Revisar envio do teste: Semana 7 - Atividade Avaliativa

Usuário	LIZIS BIANCA DA SILVA SANTOS
Curso	Programação Orientada a Objetos - COM230 - Turma 012
Teste	Semana 7 - Atividade Avaliativa
Iniciado	16/11/23 19:22
Enviado	16/11/23 19:35
Data de vencimento	17/11/23 05:00
Status	Completada
Resultado da tentativa	10 em 10 pontos
Tempo decorrido	12 minutos
Instruções	Olá, estudante!

1. Para responder a esta atividade, selecione a(s) alternativa(s) que você considerar correta(s);
2. Após selecionar a resposta correta em todas as questões, vá até o fim da página e pressione "Enviar teste".
3. A cada tentativa, você receberá um conjunto diferente de questões.

Pronto! Sua atividade já está registrada no AVA.

Resultados exibidos	Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente

Pergunta 1 1,66 em 1,66 pontos

Quando é necessário construir sistemas que executem diferentes instruções ao mesmo tempo, a linguagem Java dispõe do conceito de *threads*. As *threads* podem ser utilizadas de duas formas básicas, sendo uma delas por herança.

Considerando as características do uso das *threads*, assinale a alternativa correta.

Resposta Selecionada: c. A forma correta de criar uma *thread* por herança é usando "extends" da classe *thread*.

Respostas:
a. A forma correta de criar uma *thread* por herança é usando "implements" da interface *thread*.
b. A forma correta de criar uma *thread* por herança é usando "implements" da interface *Run*.
c. A forma correta de criar uma *thread* por herança é usando "extends" da classe *thread*.
d. A forma correta de criar uma *thread* por herança é usando "extends" da interface *thread*.
e. A forma correta de criar uma *thread* por herança é usando "implements" da interface *Runnable*.

Comentário da resposta: **JUSTIFICATIVA**
A forma correta de criar uma *thread* por meio de herança é usando o comando "extends" para a classe *Thread*. Por exemplo: class Atividade1 extends *Thread*. A implementação da interface *Runnable* é uma outra forma de criação de uma *thread* sem o uso de herança.

Pergunta 2 1,66 em 1,66 pontos

Um Socket é utilizado para a comunicação entre processos distribuídos. De forma clássica, temos o modelo cliente servidor que utiliza Sockets. Por exemplo, o Protocolo de Controle de Transmissão (TCP) é um protocolo amplamente utilizado para transmissão de dados em uma rede que suporta terminais cliente/servidor.

Sobre o que foi apresentado, observe as asserções a seguir e as relações propostas entre elas.

I. A principal diferença entre os dois principais protocolos de transporte é que o UDP é sem conexão, significando não haver sessão entre o cliente e o servidor, enquanto o TCP é orientado à conexão.

PORQUE

II. O protocolo TCP indica que uma conexão exclusiva deve, primeiro, ser estabelecida entre o cliente e o servidor para que a comunicação ocorra, caracterizando uma transmissão relativamente segura quanto à existência do *host*.

Analisando as asserções anteriores, assinale a alternativa correta.

Resposta Selecionada: a. As asserções I e II são proposições verdadeiras, e a II é uma justificativa para I.

Respostas:
a. As asserções I e II são proposições verdadeiras, e a II é uma justificativa para I.
b. A asserção II é uma proposição verdadeira, e a I é uma proposição falsa.
c. A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
d. As asserções I e II são proposições falsas.
e. As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa para I.

Comentário da resposta: JUSTIFICATIVA
A asserção I é verdadeira, pois o UDP está mais direcionado para fluxos em *stream*, prezando mais pela velocidade e pela entrega do que pelo controle e pelo tratamento de erros. É um protocolo sem conexão que permite que pacotes de dados sejam transmitidos entre aplicativos.

A asserção II é verdadeira, e justifica a I, pois o TCP age sobre o estabelecimento de conexão, para que, após esse procedimento, a comunicação seja efetivada de forma segura entre as partes.

A principal diferença entre os dois é que o UDP é sem conexão, o que significa que não há sessão entre o cliente e o servidor, enquanto o TCP é orientado à conexão, o que significa que uma conexão exclusiva deve, primeiro, ser estabelecida entre o cliente e o servidor para que a comunicação ocorra.

Pergunta 3 1,66 em 1,66 pontos

O CharStreams, em Java, possui um processo bem parecido com o ByteStream. Há classes de *streams* para lidar com caracteres, que derivam das classes Reader e Writer. Os Streams de Caracteres também podem ser usados como filtros para *streams* de *bytes*.

Sobre o fluxo de *bytes* e caracteres, identifique se são (V) verdadeiras ou (F) falsas as afirmativas a seguir.

I. ( ) Em Java, o armazenamento de caracteres é realizado por meio do padrão Unicode, em que os fluxos são convenientes para processarmos arquivos textuais.

II. ( ) A classe Java ByteInputStream obtém *bytes* de entrada de um arquivo. É usado para ler dados orientados a *byte* e estende a classe abstrata InputStream.

III. ( ) Recomenda-se a finalização do fluxo quando este não estiver mais em uso, a fim de garantir que, caso ocorra algum erro, os *streams* não sejam afetados.

IV. ( ) Invocar o método flush() garante que o último dos dados que você pensou que já havia escrito realmente saia para o arquivo.

Assinale a alternativa que apresenta a sequência correta.

Resposta Selecionada: a. V, F, V, V.

Respostas:
a. V, F, V, V.
b. F, F, V, V.
c. V, V, F, V.
d. F, V, V, F.
e. V, F, F, V.

Comentário da resposta: JUSTIFICATIVA
A afirmativa I é verdadeira, pois as classes CharacterStream são usadas, principalmente, para ler caracteres da origem e gravá-los no destino. O fluxo de caracteres Java é definido por duas classes abstratas: Reader e Writer. A classe Reader é usada para operações de entrada baseadas em fluxo de caracteres, e a classe Writer é usada para operações de saída baseadas em fluxo de caracteres.

A afirmativa II é falsa, uma vez que a descrição relacionada é pertencente à classe chamada FileInputStream. A classe FileInputStream cria um InputStream, que pode ser usado para ler *bytes* de um arquivo. É utilizado para ler dados orientados a *bytes* (fluxos de *bytes* brutos) como dados de imagem, áudio, vídeo etc.

A afirmação III é verdadeira, pois, se nos esquecermos de fechar o fluxo, o canal subjacente permanecerá aberto e, então, acabaremos com um vazamento ou um desperdício de recursos.

A afirmação IV é verdadeira, pois recomenda-se o uso do flush() quando precisar ter certeza de que todos os seus dados do *buffer* foram gravados.

Pergunta 4 1,66 em 1,66 pontos

As classes ByteStream são usadas para ler *bytes* do fluxo de entrada e gravar *bytes* no fluxo de saída. Em outras palavras, podemos dizer que as classes ByteStream leem/escrevem os dados de 8 *bits*. Podemos armazenar vídeo, áudio, personagens etc. usando classes ByteStream. Essas classes fazem parte do pacote java.io.

Sobre o texto anterior, avalie as afirmativas a seguir.

I. As classes ByteStream são divididas em dois tipos de classes, ou seja, InputStream e OutputStream. Essas classes são abstratas e as superclasses de todas as classes de fluxo de entrada/saída.

II. O método int read() retorna um inteiro, uma representação integral do *próximo byte* disponível da entrada. O inteiro 0 é retornado assim que o final da entrada é encontrado, caracterizado por "\n".

III. ByteArrayInputStream é uma classe de fluxo de *bytes* usada para ler os *bytes* da matriz de *byte por byte*. Ele estende a classe abstrata InputStream, que faz parte do pacote de classes java.io.

Está correto o que se afirma em:

Resposta Selecionada: a. I e III, apenas.

Respostas:
a. I e III, apenas.
b. I, II e III.
c. I e II, apenas.
d. I, apenas.
e. II e III, apenas.

Comentário da resposta: JUSTIFICATIVA
A afirmativa I está correta, uma vez que as classes ByteStream são usadas para ler *bytes* do fluxo de entrada e gravar *bytes* no fluxo de saída. Podemos armazenar vídeo, áudio, personagens *etc.* usando classes ByteStream. Essas classes fazem parte do pacote java.io.

A afirmativa II está incorreta, primeiro, porque o inteiro retornado com o final de entrada será -1; segundo, porque o caractere "\n" indica quebra de linha em algumas linguagens de programação.

A afirmativa III está correta, pois essa classe contém um *buffer* interno no qual existem *bytes* a serem lidos em um fluxo.

Pergunta 5 1,68 em 1,68 pontos

A utilização de APIs é uma crescente indiscutível, tanto no ponto de vista da empresa, que pode ter seus dados sem maiores problemas a qualquer hora e em qualquer lugar, quanto no ponto de vista do programador, que tem a sensação de não precisar estabelecer um trabalho adicional no processo de estabelecimento de conexão com banco de dados.

```
public static void openFile()
{
 try{
   input = new ObjectInputStream(
    Files.newInputStream(Paths.get("clients.ser")));
 }
 catch (IOException ioException){
   System.err.println("Erro ao abrir arquivo.");
   System.exit(1);
 }
}
```

Fonte: Elaboração do autor, 2023.

Sobre o código anterior, escrito em Java versão 8, analise as afirmativas a seguir.

I. A mensagem IoException pode ser lançada caso nenhum erro seja capturado durante a sua execução.

II. O bloco "try" está incorreto para instruções que precisam de uma execução dedicada e sem erros ao usuário.

III. O servidor constrói um objeto ServerSocket para especificar o número da porta na qual sua conversa ocorrerá.

Está correto o que se afirma em:

Resposta Selecionada: a. I e III, apenas.

Respostas:
a. I e III, apenas.
b. I, II e III.
c. II e III, apenas.
d. I, apenas.
e. I e II, apenas.

Comentário da resposta: JUSTIFICATIVA
A afirmativa I está correta, pois, se uma exceção não for capturada, uma mensagem relacionada à exceção de entrada e saída pode ser exibida, uma vez que estamos lidando com tentativa de abertura de arquivo, caracterizando fluxo de dados.

A afirmativa II está incorreta, pois é essencial a utilização do bloco "try" para que alguma exceção possível possa ser capturada, dado o risco de erro na manipulação de arquivos. As instruções "try-catch" são usadas em java para lidar com erros indesejados durante a execução de um programa.

A afirmativa III está correta, pois o objeto ServerSocket é construído com as especificações de porta e serviço para a comunicação ocorrer efetivamente. Esse número é usado pelo protocolo TCP/IP para identificar o aplicativo que recebe os dados. Um *endpoint*, geralmente, inclui um número de porta e um endereço IP.

Pergunta 6 1,68 em 1,68 pontos

A linguagem Java tem recursos para a programação de sistemas distribuídos em redes de computadores, e o uso de Sockets é um desses importantes recursos. Nesse sentido, analise as asserções a seguir e a relação proposta entre elas.

I. O pacote java.net contém duas classes, Socket e ServerSocket, que implementam, respectivamente, o cliente e o servidor, em uma ligação confiável, com o protocolo TCP.

II. Esse pacote também contém uma classe DatagramaPacket para a comunicação entre cliente e servidor, utilizando uma conexão não confiável, com o protocolo UDP.

Analisando essas asserções, é correto afirmar que:

Resposta Selecionada: d. as duas asserções são verdadeiras, mas a segunda não justifica a primeira.

Respostas:
a. as duas asserções são falsas.
b. a primeira asserção é falsa e a segunda é verdadeira.
c. as duas asserções são verdadeiras e a segunda justifica a primeira.
d. as duas asserções são verdadeiras, mas a segunda não justifica a primeira.
e. a primeira asserção é verdadeira e a segunda é falsa.

Comentário da resposta: **JUSTIFICATIVA**
A proposição I está correta, pois as classes Socket e ServerSocket implementam, respectivamente, o cliente e o servidor, em uma ligação confiável, com o protocolo TCP. A proposição II está correta, mas não justifica a primeira, pois as asserções apenas se complementam. A primeira apresenta dois tipos de Sockets, e a segunda apresenta uma classe DatagramaPacket para a comunicação entre cliente e servidor, utilizando uma conexão não confiável, com o protocolo UDP.