

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 1 - Atividade Avaliativa

Usuário

LIZIS BIANCA DA SILVA SANTOS

Curso

Programação Orientada a Objetos - COM230 - Turma 012

Teste

Semana 1 - Atividade Avaliativa

Iniciado

11/10/23 22:21

Enviado

11/10/23 22:26

Data de vencimento

13/10/23 05:00

Status

Completada

Resultado da tentativa

10 em 10 pontos

Tempo decorrido

4 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

2,5 em 2,5 pontos

Os paradigmas de programação visam classificar as linguagens de programação de acordo com suas funcionalidades. Nesse sentido, um paradigma fornece e determina a visão que o programador tem sobre a estruturação e a execução do programa.

Qual o paradigma em que o programador modela o problema, por meio de classes que representam objetos e situações do mundo real utilizando um encapsulamento de funcionalidades? Assinale a alternativa correta.

Resposta Selecionada: 

a. Programação orientada a objetos.

Respostas: 

a. Programação orientada a objetos.

b. Programação declarativa.

Programação procedural.

c.

d. Programação funcional.

e. Programação de baixo nível.

Comentário da resposta:

JUSTIFICATIVA

A programação declarativa utiliza palavras-chave, também conhecidas como *tags* ou assertivas, e é muito utilizada como linguagens de marcação e para a busca de dados em bancos de dados, como a linguagem SQL. A programação funcional é organizada em funções e utiliza a lógica tradicional de programação, sem usar assertivas. A programação orientada a objetos trabalha com classes que funcionam como moldes para os objetos existentes no mundo real. Na programação procedural o foco está na execução de procedimentos.

Pergunta 2

2,5 em 2,5 pontos

Scanner é uma classe pertencente ao pacote java.util, relacionada com a obtenção de dados de entrada de tipos variados, como *int*, *double* e *strings*. Ainda que não seja extremamente eficiente para cenários com restrição temporal, é um modo fácil para se ler dados de entrada em um programa Java.

Para criar um objeto da classe Scanner, normalmente passamos o objeto predefinido System.in, que representa o fluxo de entrada padrão. Podemos passar um objeto da classe File, se a intenção for ler a entrada de um arquivo.

Considerando o apresentado, analise as afirmativas a seguir.

I. Para ler valores numéricos de um determinado tipo de dados XYZ, a função a ser usada é nextXYZ(). Por exemplo, para ler um valor do tipo *short*, podemos usar nextShort().

II. Para ler *strings*, usamos nextLine(). Essa função permite que, para cada linha de entrada, seja feita a leitura, sem possibilidade de salvar o que foi lido.

III. Para ler um único caractere, usamos next().charAt(0). A função next() retorna o próximo *token*/palavra na entrada como uma *string*, e charAt(0), o primeiro caractere dessa *string*.

IV. A classe Scanner lê uma linha inteira e divide em *tokens*. *Tokens* são pequenos elementos que têm algum significado para o compilador Java.

Está correto o que se afirma em:

Resposta Selecionada: 

e. I, III e IV, apenas

Respostas: 

a. I, II e III, apenas

b. II, III e IV, apenas

c. I, II, III e IV

d. I e II, apenas

e. I, III e IV, apenas

Comentário da resposta:

JUSTIFICATIVA

A afirmativa I está correta, pois o *next* prepara a leitura para uma próxima entrada, indicando, posteriormente, o tipo de dado que será inserido para o correto armazenamento e para a correta interpretação no programa. A afirmativa II está incorreta. De fato, usamos o nextLine() para ler uma *string*. Contudo, por meio dessa leitura, é permitido utilizar funções e variáveis para armazenar e utilizar aquilo que foi lido. A afirmativa III está correta, pois a função next() sempre vai indicar ideia de “seguinte”, “próximo” e, no caso, a próxima *string* ou caractere de entrada. Utilizamos o charAt(pos) para indicar o caractere em determinada posição “pos”. Nesse caso, charAt(0) é a primeira posição de sua sentença recebida. A afirmativa IV também está correta. Para entender como um todo, o compilador se comporta como nós entendemos um texto e seu significado: dividindo os termos, interpretando cada palavra para um significado no Java, definindo o que é símbolo, palavra reservada, variável etc.

Pergunta 3

2,5 em 2,5 pontos

A programação multiplataforma significa criar um *software* que será implementado e executado em várias plataformas. Esse aplicativo pode rodar em Android e iOS, por exemplo. Para que o *software* seja considerado multiplataforma, ele deve ser executado em mais de uma arquitetura de computador ou sistema operacional.

Em relação à linguagem Java e suas características, assinale a alternativa correta a seguir.

Resposta Selecionada: 

d. Os binários são executados em uma máquina virtual Java (JVM), reconhecida pela maior parte dos dispositivos existentes

Respostas: 

a. O Java não é multiplataforma, ou seja, é preciso instalar o Java “certo” para cada plataforma e programar pensando nos recursos de *hardware* disponíveis

b. O Java é a linguagem mais popular que existe; por esse motivo, é capaz de executar em qualquer plataforma, seja *desktop* ou *mobile*.

c. O Java é uma linguagem de programação de propósito geral, com base em classes, de alto nível e procedural, portanto, multiplataforma.

d. Os binários são executados em uma máquina virtual Java (JVM), reconhecida pela maior parte dos dispositivos existentes

e. O Java é executado em um *kit* de desenvolvimento Java (JDK), reconhecido pela maior parte dos dispositivos existentes

Comentário da resposta:

JUSTIFICATIVA

O *bytecode* em Java é a razão pela qual o Java é independente de plataforma; assim que um programa Java é compilado, o *bytecode* é gerado. Para ser mais preciso, um *bytecode* Java é o código de máquina na forma de um arquivo .class. Um *bytecode* em Java é o conjunto de instruções para Java Virtual Machine (JVM) e age de forma semelhante a um compilador. Daí o fato de essa linguagem ser portátil.

Pergunta 4

2,5 em 2,5 pontos

O uso dos métodos públicos *get* e *set* é importante para a programação orientada a objetos. Em linguagem Java, esses métodos são criados para recuperar e atribuir, respectivamente, os valores aos atributos de uma classe.

Nesse contexto, analise as asserções a seguir e a relação proposta entre elas.

I. A importância do uso dos *getters* e *setters* está relacionada à proteção dos atributos de uma classe frente a possíveis acessos indevidos que possam comprometer a integridade dos dados.

II. Os métodos *getters* garantem a proteção dos atributos de acesso não autorizado e *setters* permitem a criação de instruções validadoras dos dados, garantindo, assim, a integridade dos valores.

Analizando essas asserções, é correto afirmar que:

Resposta Selecionada: 

b. as duas asserções são verdadeiras e a segunda justifica a primeira.

Respostas: 

a. as duas asserções são falsas.

b. as duas asserções são verdadeiras e a segunda justifica a primeira.

c. as duas asserções são verdadeiras, mas a segunda não justifica a primeira.

d. a primeira asserção é verdadeira e a segunda é falsa.

e. a primeira asserção é falsa e a segunda é verdadeira.

Comentário da resposta:

JUSTIFICATIVA

A primeira proposição está correta, pois, de fato, em relação aos atributos de uma classe, quando privados, o acesso ou a atribuição de valores ocorrem somente quando são utilizados os métodos públicos de acesso *getters* e *setters*. A segunda proposição está correta e justifica a primeira, pois, quando esses métodos são bem definidos, com a inclusão de código validador internamente, existe uma proteção da integridade dos dados da classe. Além de garantir a proteção dos atributos de acesso não autorizado, utilizando os métodos getters como afirma a primeira proposição, existe a possibilidade de validação desses dados dentro dos métodos *setters*, para que haja uma garantia ainda maior da integridade dos dados (como justifica a importância do uso dos *getters* e *setters* na segunda proposição).

Quinta-feira, 15 de Agosto de 2024 21h54min54s BRT

← OK