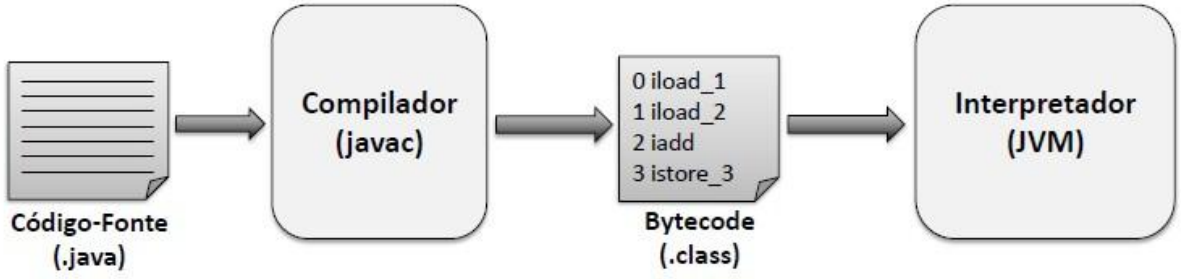
NEWTON PAIVA wyden

Lista de Exercícios da AV1

Professor: Otaviano Martins Monteiro
Aluno(a): Daniel Fernandes de Morais
Curso: Análise e Desenvolvimento de Sistemas (ADS)
Matrícula: 202451075781
Valor: 1 ponto (10% da nota final do semestre, ou seja, 10 pontos no formato antigo).

---

1) As etapas abaixo referem-se à linguagem de programação Java. Explique cada uma dessas etapas:



Resposta: O código fonte (.java) é um programa que foi escrito pelo desenvolvedor humano.

O Compilador (Javac) realiza verificações de erros de sintaxe e traduz o programa para o Bytecode

Bytecode que é uma linguagem intermediária composta por instruções independentes de plataforma, ele não é executado diretamente pelo sistema operacional, mas sim lido pelo interpretador

Interpretador(JVM) é quem lê e executa as intruções do bytecode, ela pode interpretar passo a passo ou converter em código de máquina o Bytecode, por meio de JIT (just in time) garantindo que o mesmo progrmaa rode em qualquer sistema operacional que tem a JVM instalada

2) Explique o que são tipos de dados primitivos e tipos de dados por referência.

Resposta? Os dados primitivos são:

Para os valores inteiros são: Byte, Short, Int e Long
Para os valores Reais: Float e Double
para caracteres: Char
Para Lógico: Boolean

Já os dados de referência (Qualquer tipo de dado que não seja primitivo):

Classes, objetos, arrays e Strings

3) Explique o que é programação orientada a objetos.

Resposta: É um paradigma de programação que organiza o código em torno de objetos, que são representações de coisas do mundo real ou conceitos abstratos dentro do sistema.

Se baseia na ideia principal que cada objeto possui tanto atributos (características do objeto) como também métodos (são ações que o objeto pode executar, basicamente funções)

Os objetos são criados através dessas classes, que funcionam como modelos ou moldes

Em suma é uma forma de programar que deixar o código mais organizado e fácil de manter, levando em conta que imita a forma que os humanos pensam e lidam com objetos no mundo real

4) Diferencie programação orientada a objetos da programação estruturada.

Resposta: A programação estruturada organiza o código em funções e segue uma sequência de passos, os dados e as funções ficam separados e o foco é como realizar as coisas (passo a passo)

Já a POO organiza o código em objetos que juntam dados e ações no mesmo lugar, com o foco de modelar coisas deixando mais organizado, tendo suas características e comportamentos

5) Associe: A - Modelo
B – Classe
C – Objeto
D – Método
E – Atributo

(**E**) Representa as características dos objetos
(**D**) Determina as ações e o comportamento dos objetos de uma determinada classe.
(**B**) É uma espécie de molde para a criação de objetos.
(**C**) Consiste na materialização da classe.
(**A**) É uma representação simplificada de algo do mundo real.

6) Na classe “Carro”, a cor “verde” e a ação de “acelerar” são respectivamente:

(A) Objeto e atributo;
(B) **Atributo e método; -RESPOSTA CORRETA LETRA B**
(C) Atributo e objeto;
(D) Objeto e método;

7) Uma classe “Pessoa” em um sistema acadêmico terá sempre os mesmos atributos e métodos de uma classe “Pessoa” em um sistema de pacientes de uma clínica médica? Explique:

RESPOSTA :
NÃO, as duas classes têm propósitos diferentes, uma tratasse de um sistema de escola onde terão métodos próprios e atributos únicos, e outra se trata de uma classe para sistema médico, que terão seus próprios métodos e atributos também, por mais que possam ter alguns dados parecidos, como nome, idade, etc. ambas possuem atributos e métodos únicos para suas funções.

8) Explique os modificadores de acesso private, protected, default e public.

Resposta:
private -> modificador de acesso onde os dados somente podem ser acessado dentro da própria classe, não podem ser acessados em outras classes, para serem acessados tem que ser associados a um método publico e este método publico ser acessado externamente, é o padrão mais restrito.

Protected -> Modificador de acesso onde os dados podem ser acessados dentro da mesma classe e em classes que estão presentes no mesmo pacote e de subclasses

Default -> o padrão (Pode ser identificado com default ou pode ser que simplesmente deixado em branco que vai ser considerado default) este modificador pode ser acessado apenas dentro do mesmo pacote, não sendo visível para classes em outros pacotes.

Public -> Este é o modificador mais liberal, este pode ser acessado independentemente do pacote ou classe, permitindo ser acessado em qualquer lugar, sem restrições

9) Se não for definido no código, o nível de acesso para um atributo, o mesmo será considerado automaticamente com qual nível de acesso?

RESPOSTA:
Será considerado DEFAULT

10) Escreva os nomes dos modificadores de acesso na ordem do mais restrito ao de menor restrição.

RESPOSTA: private >> protected >> default >> public

11)Como um atributo privado pode ser acessado através de outra classe?

RESPOSTA:
O atributo privado pode ser acessado em outra classe criando um método publico na mesma classe, este que terá acesso ao atributo privado, já que esta na mesma classe, desta forma ao usar este método publico em outra classe, o atributo privado poderá ser ajustado ou chamado

12)Explique a palavra reservada this.

RESPOSTA:
É utilizada para referenciar o próprio objeto da classe, permitindo diferenciar atributos e métodos do objeto de variáveis locais ou parâmetros com o mesmo nome.

13)O que é um construtor na programação orientada a objetos?

RESPOSTA:
São métodos com o mesmo nome da classe que é chamado automaticamente quando o objeto é criado, ele tem a função de inicializar os atributos do objeto e preparar eles para serem usados, eles não retornam valor e definem parâmetros para definir os valores iniciais dos atributos

14)Qual é a importância do construtor?

RESPOSTA:
O construtor é importante porque inicializa os atributos de um objeto ao cria-lo, garantindo que ele esteja em um estado válido e pronto para ser usado, e permite definir valores iniciais através de parâmetros.

15)Explique o que é encapsulamento.

RESPOSTA:
O encapsulamento é uma prática da orientação a objetos que protege os dados de uma classe, tornando os atributos private e permitindo que sejam acessados ou alterados apenas por métodos get e set, garantindo controle e segurança sobre o sistema.

16)Explique a sintaxe necessária para criar um objeto no Java.

RESPOSTA:
Para criar um objeto em Java, é necessário seguir três etapas principais: declarar a variável que fará referência ao objeto, instanciá-lo utilizando a

palavra-chave new e, se necessário, inicializá-lo por meio de um construtor, passando os valores correspondentes.

<classe> <nome do objeto> = new <classe> (ParametroDoConstruturoSeTiver)

Pessoa pessoa1 = new Pessoa(“Daniel”, 21)

17) Explique os tipos existentes de associação na programação orientada a objetos.

RESPOSTA

Associação simples (ou fraca):
É o relacionamento mais básico entre classes, onde um objeto pode conhecer outro e utilizá-lo, mas ambos existem de forma independente.
Exemplo: Um Aluno conhece um Curso, mas se o Aluno deixar de existir, o Curso continua existindo.

Agregação:
Representa uma relação “tem-um” mais forte que a associação simples, mas ainda com certa independência. Um objeto agregado faz parte de outro, mas pode existir separadamente. É chamada também de associação de todo-parte fraca.
Exemplo: Um Departamento possui Professores. Se o Departamento for excluído, os Professores continuam existindo.

Composição:
É a associação mais forte, chamada de associação de todo-parte forte. O objeto parte depende totalmente do objeto todo. Se o objeto todo for destruído, os objetos partes também deixam de existir.
Exemplo: Um Carro é composto por Motor. Se o Carro for destruído, o Motor não existe mais independentemente.

18) Crie uma classe Circulo e uma classe Retangulo, que representem um círculo e um retângulo, respectivamente. A classe Circulo deve ter um atributo raio e métodos para calcular a área e o perímetro do círculo. Utilize a fórmula da área (π * raio^2) e a fórmula do perímetro (2 * π * raio). A classe Retangulo deverá ter os atributos base e altura e métodos para possibilitar o cálculo da área e do perímetro, lembre-se que a fórmula da área do retângulo é base x altura e o perímetro do retângulo consiste na soma de todos os lados. No método main (em outra classe), crie um objeto de Circulo e um objeto de Retangulo. Chame os métodos para calcular a área e o perímetro dos objetos criados e imprima os resultados na tela.

ARQUIVO MANDADO VIA ZIP NA ATIVIDADE

19) Crie uma classe ContaBancaria, que deverá ter os seguintes atributos privados: numero (número da conta), titular (nome do titular da conta) e saldo (saldo atual). Além disso, a classe deve ter métodos para depositar dinheiro, sacar dinheiro e exibir o saldo atual. Implemente também um método que formate o nome do titular da conta para ser impresso no cartão, atendendo as seguintes regras: Se o nome completo

for por exemplo “Nayara Rocha Souza Ramos”, o método que você deverá implementar deverá gerar a String “NAYARA R. S. RAMOS”. Ou seja, o primeiro nome será apresentado com todas as letras maiúsculas, os próximos nomes ou sobrenomes (exceto o último) serão abreviados, apresentando apenas a primeira letra em maiúsculo seguido do caractere “.” e o último sobrenome será impresso por completo e também com as letras maiúsculas. No entanto, para sobrenomes que tenham as preposições “de”, “da”, “das”, “do”,“dos” e “e”, essa preposição será omitida. Tome cuidado com a forma de implementar essa validação, pois existem alguns sobrenomes que não são preposições e possuem 3 caracteres ou menos.
Segue um exemplo: o nome “Matheus dos Anjos Zoe”, tem a preposição “dos” que deverá ser ignorada na formatação do nome, mas tem o sobrenome “Zoe” que contém 3 caracteres e deverá ser exibido. Deste modo, esse nome formatado será “MATHEUS A. ZOE”.

ARQUIVO MANDADO VIA ZIP NA ATIVIDADE

20) Crie uma classe chamada Pessoa, que tenha os seguintes atributos privados: nome, telefone e e-mail. Crie um construtor que solicite esses 3 parâmetros. Crie 2 objetos da classe Pessoa, usando o construtor que você criou anteriormente. Crie métodos que possibilitem alterar os valores dos atributos criados anteriormente, para novos valores. Crie métodos para imprimir os atributos de cada objeto.

ARQUIVO MANDADO VIA ZIP NA ATIVIDADE

21) Criar uma classe “Aluno”, que tenha os seguintes atributos: int registroAcademico, String CPF, String nome, vetor do tipo double para armazenar 4 notas finais (uma para cada disciplina) e um vetor do tipo inteiro para armazenar a quantidade de faltas para cada uma das 4 disciplinas. A classe “Aluno” terá os seguintes métodos: calcularMedia(), que retorna a média aritmética do aluno, além do método calcularSituacao(), que retorna se o aluno foi aprovado ou reprovado, com base em dois critérios: todas as notas devem ser acima de 60 e a quantidade de faltas em cada disciplina deve ser inferior a 5. Nesta atividade, a classe “Aluno” deverá ter um construtor, além dos métodos get() e set() que forem necessários. No método main (em outra classe), crie dois objetos de Aluno e faça testes.

ARQUIVO MANDADO VIA ZIP NA ATIVIDADE