**UNIVILLE - Universidade da Região de Joinville**   
  
**Disciplina**: Programação Orientada a Objetos  
  
Professor: Leanderson André  
  
Curso: Sistemas de Informação  
  
Aluno: Eliton Misael Zimmermann

**Exercícios Programação Orientada a Objetos**

**1 – O que é um objeto?**

R: É uma coisa, entidade que podemos imaginar e que tenha uma identidade.

**2 – Identifique 3 características e 3 comportamentos de um carro:**

R: Características: possui portas, pneus, motor  
Comportamento: parado, em movimento,

**3 - No contexto de Orientação a Objetos, as características e comportamentos são chamados respectivamente de?**

R:

**4 - Qual é o objetivo da programação orientada a objetos?**

R: Alguns dos objetivos é reduzir a complexidade no desenvolvimento de softwares e aumentar a produtividade.

**5 – O que é abstração? Cite um exemplo**

R: É a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais.

Exemplo: Uma conta de banco possui as características de saldo, limite, número. Sendo as ações possíveis: sacar, depositar e ver estrato. Aplicando a abstração, ficaria da seguinte maneira.

public class Conta {

int numero;

double saldo;

double limite;

void depositar(double valor){

this.saldo += valor;

}

void sacar(double valor){

this.saldo -= valor;

}

void imprimeExtrato(){

System.out.println("Saldo: "+this.saldo);

}

}

**6 - O que é classe?**

R: Uma classe é um gabarito para a definição de objetos. Através da definição de uma classe, descreve-se que propriedades ou atributos --o objeto terá. A definição de uma classe descreve também qual o comportamento de objetos da classe, ou seja, que funcionalidades podem ser aplicadas a objetos da classe.

**7 – Qual o padrão utilizado para nomear classes? Cite um exemplo**

R: Por convenção, toda classe deve começar com uma letra maiúscula e não pode conter letras com caracteres acentuados. Portanto, não é possível declarar uma classe com qualquer caractere especial

Exemplo: Pessoa, ContaBanco, RelatorioMes, Compra,...

**8 – Qual opção apresenta corretamente o nome da classe carro elétrico?**

R: CarroEletrico

**9 - Qual é o padrão utilizado para nomear os atributos? Cite um exemplo.**

R: Os atributos (variáveis) podem começar com qualquer letra e os caracteres $ ou \_, porém não podem começar com números.

Caso o nome de um atributo (variável) seja composto por mais de uma palavra, a primeira letra de cada palavra deve ser em maiúscula.

Exemplo: vendasMes, relatorioMensalLoja, compra...

**10 - Qual opção apresenta corretamente o nome do atributo cor de fundo?**

R: corDeFundo

**11 - Qual é o padrão utilizado para nomear os métodos? Cite um exemplo.**

Métodos (funções) podem começar com qualquer letra minúscula, não podem possuir acentos e não podem começar com números.

Caso o nome de um método seja composto por mais de uma palavra, a primeira letra da segunda palavra e das palavras posteriores deve ser em maiúscula.

Exemplo: imprimir, calcularMedia, valorUnitario...

**12 - Qual opção apresenta corretamente o nome do método está vazio?**

R: estaVazio

**13 - No contexto de orientação a objetos, o que é um objeto? Em que momento existe um objeto? Quando ele deixa de existir?**

**14 - Qual é o objetivo do operador new?**

R: New é utilizado para criar uma nova instância de uma determinada classe

**15 - O que é o construtor? Qual é o seu objetivo? Qual deve ser o seu nome? Cite um exemplo.**

R: O construtor é definido como um método cujo nome deve ser o mesmo nome da classe e sem indicação do tipo de retorno -- nem mesmo void. O construtor é unicamente invocado no momento da criação do objeto através do operador new. Exemplo abaixo.

public class Carro{

/\* CONSTRUTOR DA CLASSE Carro \*/

public Carro(){

//Faça o que desejar na construção do objeto

}

}

public class Aplicacao {

public static void main(String[] args) {

//Chamamos o construtor sem nenhum parâmetro

Carro fiat = new Carro();

}

}

**16 - Caso o construtor da classe Aluno não for declarado, qual será o seu construtor?**

R: Será o construtor padrão, pois ao criar uma classe, por padrão sempre teremos um construtor padrão.

**17 - Crie um exemplo de instanciação da classe Aluno. Utilize o construtor padrão.**

class Pessoa {

String tipo = "";

public Pessoa(String tipo){

this.tipo = tipo; //Aqui, o this.tipo representa a variavel que esta fora do escopo, ou seja, o que recebemos como argumento

//é passado para a variavel de instancia.

}

public String getTipo(){

return tipo;

}

}

class Executa {

public static void main(String args[]){

Pessoa p = new Pessoa("Pessoa Juridica");

System.out.println("O tipo de pessoa é: "+p.getTipo());

}

}

**18 - O que é encapsulamento?**

R: É a técnica utilizada para esconder uma ideia, ou seja, não expôr detalhes internos para o usuário, tornando partes do sistema mais independentes possível. Em programação orientada a objetos significa separar o programa em partes, o mais isoladas possível. A ideia é tornar o software mais flexível, fácil de modificar e de criar novas implementações.

**19 - Qual é o objetivo do modificador de acesso public?**

R: Seu objetivo é acessar a qualquer código externo da classe, ela pode ser acessada de qualquer lugar e entidade.

**20 – Qual o objetivo de acesso private?**

R: Seu objetivo é proibir qualquer acesso externo à própria classe, inclusive das filhas.

**21 – Como é aplicado o encapsulamento de uma classe? Considere a classe Aluno com o atributo matrícula.**

O encapsulamento organiza os dados que estejam relacionados, agrupando-os (encapsulando-os) em objetos (classes), reduzindo as colisões de nomes de variáveis e, da mesma forma, reunindo métodos relacionados às suas propriedades (ou variáveis de classe).

Um mecanismo da linguagem de programação para restringir o acesso a alguns componentes dos objetos, escondendo os dados de uma classe e tornando-os disponíveis somente através de métodos.

**22 – Qual o objetivo dos métodos getters? Crie um exemplo.**

R: O uso do get é usado para obter informações. Esse tipo de método sempre retorna um valor.

**23 – Qual o objeitivo dos métodos setters? Crie um exemplo**

R: O uso do set para definir valores. Esse tipo de método geralmente não retorna valores.

**24 – O diagrama de classe UML é composto por 3 partes. O que vai em cada parte?**

R: Nome da classe, atributos e métodos.

**25 – Qual é o padrão utilizado para representar em atributo no diagrama de classe UML? Crie um exemplo.**

|  |
| --- |
| **Clientes** |
| -codPessoa -numcliente |
| +setCadastrar() +getConsultar() |

**26 – Qual é o padrão utilizado para representar um método no diagrama de classe UML? Crie um exemplo?**

Uma classe é representada por um retângulo com três divisões, são elas: O nome da classe, seus atributos e por fim os métodos.

**27 – Como o construtor de uma classe pode ser diferenciado no diagrama de classe UML? Cite um exemplo.**

**28 - Quais são os símbolos utilizados no diagrama de classe UML para representar**

**os modificadores de acessos ao atributos e métodos? Crie um exemplo.**

+ public  
- private  
# protected

|  |
| --- |
| **Clientes** |
| -codPessoa #numcliente |
| +setCadastrar() +getConsultar() |

**29 - Considere a classe Cliente, com os atributos nome, e-mail e telefone com os**

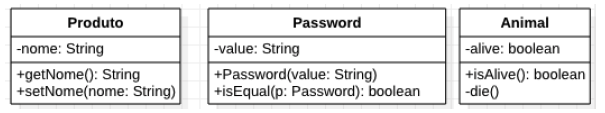
**respectivos métodos getters e setters. Desenvolva o diagrama de classe UML.**

**30 - Considere a classe Cliente apresentado no Problema 29, e desenvolva o código**

**Java correspondente.**

**31 - Desenvolva o código java das classes apresentadas no diagrama de classes**

**UML abaixo:**



**32 -**

**REFERÊNCIAS**

**Construtor**

<https://www.devmedia.com.br/construtores-em-java/28618>