Teoria:

* O que é o **JDK**?

kit de desenvolvimento java, é um conjunto de ferramentas de sofware necessario para desenvolver aplicações java.

* Explique o que é o **Eclipse**.

ambiente de desenvolvimento integrado (ide) serve para programar em java para criar apis e ate mesmo sofware.

* Defina **classes** e **objetos**.

uma classe é como um molde para criar algo, é como uma caixa de ferramentas que contém instruções sobre como construir algo especifico.

um objeto é como um avião de papel real que voce fez usando o molde.

cada avião de papel que voce faz seguindo esse molde é um objeto.

cada avião pode ser de uma cor diferente, voar de maneiras diferentes, e ser decorado de fomar unica, mas todos são feitos usando o mesmo model.

* O que são **atributos** e **métodos**?

**Atributos** são como características que um objeto possui. Eles representam os dados que descrevem o estado de um objeto. Por exemplo, se pensarmos em um objeto "Carro", seus atributos poderiam incluir coisas como cor, modelo, ano e número de portas.

**Métodos** são ações que um objeto pode realizar. Eles definem o comportamento do objeto e o que ele pode fazer. Por exemplo, métodos de um objeto "Carro" poderiam ser acelerar, frear, virar à esquerda, virar à direita, ligar e desligar o motor.

* Qual a responsabilidade das classes de **entidade** em um sistema orientado a objetos?

as classes de entidade em um sistema orientado a objetos são responsáveis por representar objetos do mundo real ou conceitual. elas encapsulam dados(atributos) e comportamentos (métodos) relacionados a esses dados, definem relacionamentos entre objetos, abstraem conceitos complexo, gerenciam o estado do objetos e podem lidar com a persitência de dados. em resumo, essas classes organizam e estruturam os dados essencias do sistema, facilitando sua manipulação e integração com outras partes do sofware

* O que são métodos **set** e **get**? Qual a função de cada uma delas?

Set é o metodo utilizado para atribuir ou definir valores a atributos privados de uma classe

Get são metodos usados para obeter ou recuperar o valor atual do atributos privados de uma classe.

famos entrada e saida de dados.

* Explique as diferenças de cada **modificador de visibilidade**:

**private**

Acesso: Restrito à própria classe.

Objetivo: Proteger dados, implementação interna.

Benefício: Melhor encapsulamento, segurança dos dados.

**public**

Acesso: Disponível para todas as classes.

Objetivo: Facilitar acesso externo a métodos e atributos.

Benefício: Integração e interação fácil entre partes do sistema.

private: Acesso restrito à classe, protege dados.

public: Acesso global, facilita interação externa.

* Explique para que servem as seguintes palavras reservadas da linguagem:

**class**: Define a estrutura para criar objetos.

**package:** Organiza classes em grupos para gerenciar e evitar conflitos de nomes.

**void:** Indica que um método não retorna nenhum valor.

**import:** Permite usar classes e pacotes de outros arquivos Java no seu programa.

**public:** Permite acesso a métodos, atributos ou classes de qualquer parte do código.

**private:** Restringe o acesso a métodos ou atributos apenas à própria classe onde estão definidos.

**new:** Cria uma nova instância (objeto) de uma classe.

**try:** Define um bloco de código onde pode ocorrer uma exceção.

**catch:** Captura exceções que ocorrem dentro de um bloco try para tratamento específico.

* Explique o que é **encapsulamento**.

Encapsulamento em programação orientada a objetos é esconder os detalhes internos de implementação de um objeto, permitindo acesso controlado aos seus atributos e métodos.

* Explique os atalhos:

CTRL + SHIFT + O

Ação: Organiza os imports automaticamente no arquivo Java atual.

CTRL + SHIFT + F

Ação: Formata o código-fonte selecionado ou o arquivo inteiro conforme as convenções de formatação definidas.

CTRL + SHIFT + S

Ação: Salva todos os arquivos abertos no momento.

syso + CTRL + espaço

Ação: Insere automaticamente System.out.println(); no código.

main + CTRL + espaço

Ação: Gera automaticamente o método main dentro de uma classe Java.

* Para que serve a classe **Scanner**?

A classe Scanner em Java serve para capturar e processar entrada de dados do usuário ou de outros fluxos, como arquivos, facilitando a leitura de diferentes tipos de dados para serem utilizados em um programa.

* Explique como funciona o tipo de dado **UUID** em Java.

O UUID em Java é um tipo de dado que representa um identificador único globalmente. Ele é gerado pela classe java.util.UUID, permitindo criar UUIDs aleatórios ou a partir de strings. Esses identificadores são usados para garantir que cada entidade em um sistema tenha um identificador único e são cruciais em ambientes distribuídos onde a duplicação de identificadores deve ser evitada a todo custo.

Prática:

* Crie um projeto Java chamado **javaExercicio01**, dentro do projeto crie os pacotes abaixo:



* No pacote “entities”, crie uma classe chamada Funcionario, com os atributos abaixo:



* Na classe acima, faça o encapsulamento dos atributos gerando os métodos setters e getters.
* No pacote “main”, crie uma classe chamada Main.java e declare um método main. Neste método faça com que o usuário possa preencher os dados do funcionário utilizando a biblioteca Scanner e ao final o programa deverá imprimir os dados do funcionário.

Exemplo:

**\*\*\* CADASTRO DE FUNCIONÁRIO \*\*\***

**INFORME O NOME DO FUNCIONÁRIO:**

**Sergio Mendes**

**INFORME O CPF DO FUNCIONÁRIO:**

**123.456.789-00**

**INFORME A MATRÍCULA DO FUNCIONÁRIO:**

**F001-2023**

**INFORME O SALÁRIO DO FUNCIONÁRIO:**

**3000**

**DADOS DO FUNCIONÁRIO:**

**ID...........: defa88b2-7874-42a3-896b-887e6c3a44ad**

**NOME.........: Sergio Mendes**

**CPF..........: 123.456.789-00**

**MATRICULA....: F001-2023**

**SALARIO......: 3000**

Teoria (código limpo e padrões):

Qual dos seguintes princípios é fundamental para escrever código limpo em Java?

a) Uso excessivo de comentários.

b) Manter classes pequenas e métodos concisos.

c) Adotar o padrão de codificação baseado em espaços em branco.

d) Utilizar nomes de variáveis e métodos longos para facilitar a compreensão.

Em relação aos comentários em Java, qual das seguintes afirmações é verdadeira?

a) Comentários extensivos são essenciais para um código limpo.

b) Comentários devem explicar cada linha de código em detalhes.

c) O código deve ser autoexplicativo, minimizando a necessidade de comentários.

d) Comentários devem ser escritos apenas no idioma inglês.

Qual é a prática recomendada para nomear variáveis e métodos em Java?

a) Usar abreviações sempre que possível para reduzir o tamanho do código.

b) Utilizar nomes descritivos que indiquem claramente a finalidade da variável ou método.

c) Nomear variáveis e métodos com uma única letra para economizar espaço.

d) Adotar nomes genéricos como "var1", "var2", etc., para facilitar a leitura.

Qual dos seguintes conceitos contribui para tornar o código mais limpo e legível em Java?

a) Aninhar vários loops e condicionais em um único método.

b) Ignorar o uso de constantes e valores literais.

c) Evitar a duplicação de código através da criação de métodos e classes reutilizáveis.

d) Escrever classes com muitas responsabilidades para aumentar a coesão.

Qual é a melhor prática para lidar com exceções em Java, promovendo código limpo?

a) Capturar todas as exceções em um bloco try-catch global.

b) Ignorar exceções e deixar o sistema lidar com elas automaticamente.

c) Tratar exceções específicas em níveis apropriados e fornecer mensagens de erro significativas.

d) Evitar o uso de blocos try-catch para não comprometer a performance do código.

O que é acoplamento em Java e por que é importante considerá-lo ao escrever código limpo?

a) Acoplamento refere-se à medida em que uma classe depende de outra. É importante reduzir o acoplamento para tornar o código mais flexível e fácil de manter.

b) Acoplamento é a quantidade de comentários presentes no código. Quanto mais comentários, mais acoplamento.

c) Acoplamento é o processo de criar relacionamentos entre classes, o que é fundamental para garantir a coesão do código.

d) Acoplamento é a técnica de mesclar várias classes em uma única, facilitando a comunicação entre elas.

Qual é a importância de escrever testes unitários para garantir código limpo em Java?

a) Testes unitários são desnecessários quando se escreve código limpo.

b) Testes unitários ajudam a verificar se o código está funcionando corretamente e fornecem documentação viva.

c) Testes unitários aumentam a complexidade do código e devem ser evitados.

d) Testes unitários são úteis apenas para desenvolvedores experientes.

Quais são algumas das características de uma classe bem projetada em Java?

a) Classes com muitas responsabilidades e métodos longos.

b) Classes que dependem fortemente de outras classes e não possuem coesão.

c) Classes que aderem ao princípio de responsabilidade única, têm baixo acoplamento e alta coesão.

d) Classes que não utilizam encapsulamento.

O que é "Refatoração" em Java e como ela contribui para o código limpo?

a) Refatoração é o processo de introduzir mais bugs no código para melhorar sua qualidade.

b) Refatoração é o ato de reescrever completamente o código de um projeto.

c) Refatoração é o processo de fazer pequenas melhorias incrementais no código, mantendo seu comportamento externo inalterado.

d) Refatoração é uma técnica usada apenas por programadores novatos.