

ATIVIDADE

Assunto:

Interfaces.

Orientações:

A atividade deve ser executada individualmente e entregue através do ambiente *Google Classroom*.

Regras de criação dos programas:

Crie um novo projeto Java denominado **AtividadeInterfaces**. As classes devem possuir os nomes informados no texto. Ao final, o projeto deve ser exportado para um arquivo em formato ZIP.

Nome completo:

Robert Silva Queiroz

1. Quais as diferenças entre classes abstratas e interfaces? Explique.

Definição e Propósito:

Classe Abstrata: É uma classe que pode conter métodos abstratos (sem corpo) e concretos (com implementação). Serve como um esqueleto ou modelo para outras classes, oferecendo um ponto de partida comum para classes relacionadas.

Interface: Define um contrato que outras classes devem seguir, especificando apenas a assinatura dos métodos, sem implementação (exceto com default methods, discutidos abaixo). Interfaces são usadas para garantir que diferentes classes implementem determinados métodos, permitindo flexibilidade na implementação.

2. Interfaces podem ter métodos concretos? Explique.

Sim, a partir do **Java 8**, interfaces podem ter métodos concretos usando default e static methods.

Default Methods: Permitem que uma interface forneça uma implementação padrão para um método. Isso é útil para adicionar novos métodos a interfaces existentes sem quebrar as implementações das classes que já implementam essa interface.

Static Methods: São métodos concretos que podem ser chamados diretamente na interface, sem precisar de uma instância.

3. Demonstre como o uso de *default methods* pode evitar a repetição de código.

Imagine que você tem várias classes que representam diferentes tipos de animais, e todas essas classes precisam implementar uma interface **Animal** com um método **dormir()** que deve imprimir uma mensagem padrão. Sem default methods, cada classe teria que implementar o método **dormir()** separadamente, mesmo que a implementação fosse idêntica.

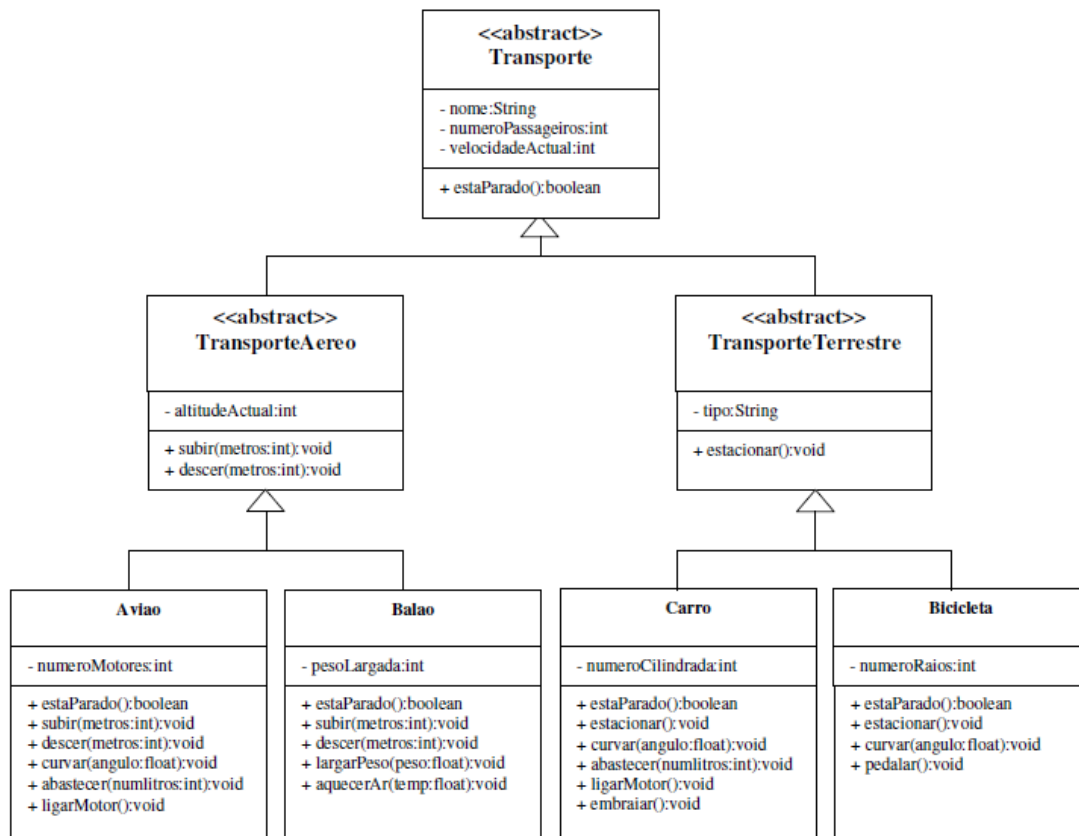
4. Em uma situação em que classes abstratas e interfaces são opções viáveis, qual deve ser utilizada prioritariamente?

Interface: Se a classe precisa implementar múltiplos comportamentos de diferentes fontes (ou seja, se há uma necessidade de múltiplas heranças de comportamento), opte por interfaces.

Classe Abstrata: Se há um comportamento padrão que pode ser compartilhado entre várias subclasses, e essas subclasses formam uma hierarquia relacionada.

Em geral, **interfaces** são mais flexíveis e são preferidas quando você deseja evitar limitações de herança única e promover a composição de comportamentos.

5. Considere o diagrama UML a seguir e faça o que se pede:



O que se pede:

- Crie uma interface de nome Motorizado em que são declarados os métodos void ligarMotor() e void abastecer(int numLitros).
- Implemente a interface Motorizado nas classes Aviao e Carro.
- Escreva um programa de teste capaz de verificar a implementação anterior.
- Crie uma interface de nome Conduzivel onde é declarado o método void curvar(float angulo).
- Implemente a interface Conduzivel nas classes Aviao, Carro e Bicicleta.
- Complete o programa de teste criado anteriormente por forma a testar estas últimas implementações.

Boa sorte!

Prof. Igor.