1. **Variáveis static e Variáveis de Instância**

Variáveis static: São variáveis de classe, ou seja, compartilhadas por todas as instâncias dessa classe. Elas podem ser acessadas sem a necessidade de criar uma instância da classe. Modificações em uma variável static afetam todas as instâncias porque a variável pertence à classe, não a uma instância específica.

Diferenças principais:

Variáveis static são inicializadas quando a classe é carregada pela primeira vez, enquanto variáveis de instância são inicializadas quando uma nova instância é criada.

Variáveis static são acessadas usando o nome da classe, variáveis de instância requerem uma referência de objeto.

Existe apenas uma cópia de uma variável static por classe, enquanto cada instância tem sua própria cópia de uma variável de instância.

1. **Conceitos de POO**

Abstração: Refere-se ao conceito de esconder a complexidade de implementação e mostrar apenas as funcionalidades aos usuários. Permite focar no que um objeto faz, em vez de como ele faz.

Encapsulamento: A prática de manter campos (variáveis) em uma classe privada e fornecer acesso a eles via métodos públicos (getters e setters). Isso protege a integridade dos dados.

Instância: Uma ocorrência específica de uma classe. Enquanto uma classe é a definição ou molde, uma instância é um objeto concreto baseado nessa classe.

Tipos de dados primitivos em Java: int, double, boolean, char.

1. **Sobre a Classe Blue**

a. Variáveis de classe: static int color; e public static String NAME = "Blue"; (porque são marcadas com static).

b. Construtores: public Blue(int hue) { ... } (é o único construtor explicitamente declarado).

c. Implementação de setHue:

java

private int hue;

public void setHue(int hue) {

this.hue = hue;

}

d. Implementação do construtor (Considerando a implementação de setHue sugerida):

java

public Blue(int hue) {

this.hue = hue;

}

1. **Código Bird e Nightingale**

a. Saída do método main:

mathematica

Before: 0 After: 1Flap Flap:0

Justificativa: referenceCount é incrementado no construtor de Nightingale, que é chamado antes do segundo print. a não é inicializado, então seu valor padrão é 0.

b. Quantas instâncias da variável Count e a:

Uma única instância de referenceCount será criada e compartilhada entre todas as instâncias de Bird e Nightingale porque é uma variável static.

Cada objeto Bird e Nightingale terá sua própria cópia da variável de instância a, resultando em 15 instâncias dessa variável (uma para cada objeto criado).

1. **Implementação das Classes Pessoa, Trabalhador e Estudante**

java

class Pessoa {

public void reportar() {}

}

class Trabalhador extends Pessoa {

private String nome;

private double salario;

public Trabalhador(String nome, double salario) {

this.nome = nome;

this.salario = salario;

}

@Override

public void reportar() {

System.out.println("Nome:" + nome + " Salario: " + salario);

}

}

class Estudante extends Pessoa {

private String nome;

private int idade;

public Estudante(String nome, int idade) {

this.nome = nome;

this.idade = idade;

}

@Override

public void reportar() {

System.out.println("Nome:" + nome + " Idade: " + idade);

}

}

Estas classes implementam a funcionalidade básica conforme solicitado, permitindo a criação de objetos Trabalhador e Estudante que podem ser armazenados em uma lista de objetos Pessoa e ter seus métodos reportar () chamados para exibir suas informações.