Exercícios – Classes abstractas e Interfaces

---

1. Defina qual o seu conceito de classe abstracta.

2. Defina qual o seu conceito de interface.

3. Defina uma interface com o nome Jogo. Crie os métodos iniciar(), jogar() e finalizar().

4. Crie uma classe abstracta com o nome JogoComBola e faça-a implementar a interface Jogo. Crie um método abstracto denominado setNomesEquipes(String equipe1, String equipe2).

5. Crie uma classe com o nome Futebol e faça-a herdar da classe JogoComBola. Repare que, por se tratar de uma classe concreta, você será obrigado pelo compilador a implementar os métodos da classe abstracta e também os métodos definidos na interface. Coloque implementações simples nestes métodos como mensagens, por exemplo.

6. Para treinar, crie uma classe com o nome Vólei nos mesmos moldes da classe Futebol.

7. Crie uma classe abstracta com o nome JogoDeCartas e faça-a herdar da interface Jogo. Crie um método abstracto qtdCartasDistribuidas() com o retorno do tipo inteiro. Crie outro método abstracto com o nome numeroParticipantes() também com retorno inteiro.

8. (classe abstracta, herança, sobrescrita, polimorfismo)

Crie uma hierarquia de classes para representar os diferentes tipos de funcionários de um escritório que tem os seguintes cargos: gerente, assistente, vendedor. Escreva uma classe base abstracta chamada Funcionario que declara um método abstracto:

Assinatura double calculaSalario()

Esta classe também deve definir os seguintes atributos: nome (tipo String), matricula (tipo String) e salario_base (tipo double). Use encapsulamento e forneça um construtor que recebe os valores correspondentes a serem armazenados nos respectivos atributos. Esta classe abstracta deverá ser estendida pelas outras classes representativas dos tipos de funcionários, portanto devem ser escritas as classes Gerente, Assistente e Vendedor. Em cada classe deve-se sobrescrever o método calculaSalario de forma que cálculo do salário é feito assim: O gerente recebe duas vezes o salário_base, o assistente recebe o salário_base

e o vendedor recebe o salário_base mais uma comissão definida no construtor de sua classe. Crie uma classe Teste com um método main que cria um objecto de cada tipo e os armazena em uma lista e depois calcula a folha salarial dos três funcionários e imprime o valor total. Indique quais conceitos de POO você usou e como foi usado.

9. (classe abstracta, sobrescrita, polimorfismo)

Explique os resultados da execução do método main abaixo e quais são os conceitos fundamentais de orientação a objectos que são aplicados.

```
abstract class Actor {

  public abstract void acto();

}

class ActorFeliz extends Actor {

  public void acto() {

    System.out.println("Actor feliz");

  }

}

class ActorTriste extends Actor {

  public void acto() {

    System.out.println("Actor triste");

  }

}

class Palco {

  private Actor actor = new ActorFeliz();

  public void altera() { actor = new ActorTriste(); }

  public void actuar() { actor.acto(); }

}

public class Programa {

  public static void main(String[] args) {
```

```
    Palco palco = new Palco();
    palco.actuar();
    palco.altera();
    palco.actuar();
  }
}
```

10. (classe abstracta, sobrescrita, polimorfismo)

Escreva uma classe abstracta chamada CartaoWeb. Essa classe representa todos os tipos de cartões web e conterá apenas um atributo: destinatario (tipo String). Nessa classe você deverá também declarar o método public abstract void showMessage(). Crie classes filhas da classe CartaoWeb: DiaDosNamorados, Natal, Aniversario. Cada uma dessas classes deve conter um método construtor que receba o nome do destinatário do cartão. Cada classe também deve implementar o método showMessage(), mostrando uma mensagem ao usuário com seu nome e que seja específica para a data de comemoração do cartão. Escreva um programa e no método main crie um array de CartaoWeb. Insira instâncias dos 3 tipos de cartões neste array. Após, use um ciclo for para exibir as mensagens deste cartão chamando o método showMessage(). Em que linha(s) acontece polimorfismo nesse código?