PRÁTICA LABORATORIAL 12

Objetivos:

- Herança
- Classes Abstratas

EXERCÍCIOS

Parte 1

1. Crie uma classe abstrata chamada "FiguraGeometrica" com o seguinte atributo: Cor (String). A classe deve ter um construtor que recebe a cor como parâmetro.
    a. Crie duas subclasses concretas de FiguraGeometrica: "Retangulo" e "Circulo". Essas classes devem ter um construtor que chama o construtor da superclasse (FiguraGeometrica) e também ter um método específico para cada uma delas:
    b. Para Retangulo: "calcularArea()" que calcula a área do retângulo (base x altura) e retorna o valor como um double.
    c. Para Circulo: "calcularArea()" que calcula a área do círculo (π x raio^2) e retorna o valor como um double.
    d. Crie uma classe chamada "FigurasGeometricasDemo" com um método main que cria um objeto de cada uma das classes (FiguraGeometrica, Retangulo e Circulo) e chama o método "calcularArea()" de cada um deles. (Exemplo de Solução na Próxima Página).

```java
public class ExercicioFigurasGeometricas {
    public static void main(String[] args) {
        FiguraGeometrica figura = new Retangulo("vermelho", 4.0, 5.0);
        System.out.println("Área do retângulo: " + figura.calcularArea());

        figura = new Circulo("verde", 3.0);
        System.out.println("Área do círculo: " + figura.calcularArea());
    }
}
```

2. Considere um sistema que gere aviões. O sistema deve permitir a criação de diferentes tipos de aviões, cada um com suas características específicas. Para isso, implemente as classes a seguir
    a. A classe abstrata Aviao deve conter os seguintes atributos privados: modelo (String): indica o modelo do avião. fabricante (String): indica o fabricante do avião. capacidadePassageiros (int): indica a capacidade de passageiros do avião. velocidadeMaxima (double): indica a velocidade máxima do avião. A classe deve ter um construtor que inicializa esses atributos e métodos getters e setters para cada um deles.
    b. A classe AviaoComercial deve ser uma subclasse de Aviao e deve conter os seguintes atributos privados: numeroTripulantes (int): indica o número de tripulantes necessários para operar o avião. numeroComissarios (int): indica o número de comissários de bordo necessários para atender os passageiros. A classe deve ter um construtor que inicializa esses atributos e métodos getters e setters para cada um deles.
    c. A classe AviaoCarga deve ser uma subclasse de Aviao e deve conter os seguintes atributos privados: capacidadeCarga (double): indica a capacidade de carga do avião. A classe deve ter um construtor que inicializa esses atributos e métodos getters e setters para cada um deles.
    d. A classe AviaoCombate deve ser uma subclasse de Aviao e deve conter os seguintes atributos privados: numeroMissoes (int): indica o número de missões que o avião já realizou. numeroDisparos (int): indica o número de disparos que o avião já efetuou. A classe deve ter um construtor que inicializa esses atributos e métodos getters e setters para cada um deles.
    e. Na classe principal do sistema, crie pelo menos um objeto de cada tipo de avião e exiba as informações de cada um, como modelo, fabricante, capacidade de passageiros, velocidade máxima e atributos específicos de cada tipo de avião.

3. Implemente um jogo sobre Personagens que enfrentam inimigos perigosos, o objetivo será o jogador definir qual a sua personagem e, seguidamente, jogar contra os inimigos, para isso:
    a. Crie uma classe abstrata “Entidade” que tem como atributos nome (String), vida (int), força (int). Seguidamente, crie duas subclasses Personagem e NPC (non-playable character).
    b. A classe Personagem terá o atributo nível (int), categoria (ENUM contendo, por exemplo, cavaleiro, mago, arqueiro, etc…) e arma (String).
    c. Deve ser implementado o método atacar que recebe um inimigo como parâmetro. Este método “atacar” compara as estatísticas da personagem vs NPC. Tenha em consideração o seguinte:
        i. O jogo será por turnos, pelo que no primeiro turno de ataque começa sempre o personagem. Será deduzida a força da personagem à vida do inimigo. Ao fim do ataque do jogador, o inimigo retribui, sendo deduzida à vida do jogador a força do inimigo.
        ii. O método deve executar até que uma das entidades fique com a vida igual ou menor que zero. Se o jogador ficar vivo, imprime uma mensagem na consola com o seguinte aspeto: “Parabéns (“categoria da personagem”) (“nome da personagem”), ganhou um combate contra um (“inimigo”) com o uso de (“arma”).” e passa de nível e incrementa a vida e a força em 10%, restaurando a vida ao valor total mais incremento. Se o jogador morrer, o programa deve encerrar com a mensagem “Perdeu”.
    d. Numa classe Jogo, defina um método main, instancie uma personagem e três inimigos, de seguida, faça a personagem enfrentar todos os inimigos.

Bom trabalho! ☺