

1. Classe **PersonagemDeJogo**

Crie um personagem para um jogo com:

- Atributos: nome, tipo (guerreiro, mago, arqueiro), vida, força.
- Métodos:
 - `atacar()`: imprime uma frase com o tipo de ataque.
 - `receberDano(int dano)`: diminui a vida.
 - `exibirStatus()`: mostra o nome, tipo e vida atual.

Desafio extra: Crie dois personagens e simule uma pequena luta no `main`.

2. Classe **PetVirtual**

Simule um bichinho virtual:

- Atributos: nome, fome, felicidade (0–100).
- Métodos:
 - `alimentar()`: diminui a fome.
 - `brincar()`: aumenta felicidade.
 - `status()`: imprime o estado atual do pet.

Desafio extra: Se a fome passar de 80, o pet fica triste (felicidade diminui).

3. Classe **PlaylistMusical**

Simule uma playlist:

- Atributos: nome da playlist, música atual.
- Métodos:

- `adicionarMusica(String musica)`
- `tocarProxima()`
- `mostrarMusicaAtual()`

Desafio extra: Crie uma lista de 5 músicas e toque todas em sequência.

4. Classe **DiarioSecreto** (com senha)

Crie um diário que só pode ser acessado com senha:

- Atributos: dono, senha, texto.
- Métodos:
 - `escrever(String senha, String texto)`
 - `ler(String senha)`

Desafio extra: Se a senha estiver errada, mostre "Acesso negado".

5. Classe **Lâmpada**

Simule uma lâmpada inteligente:

- Atributos: ligada (boolean), cor (branco, amarelo...).
- Métodos:
 - `ligar()`, `desligar()`
 - `trocarCor(String novaCor)`
 - `mostrarEstado()`

Desafio extra: Se a lâmpada estiver desligada, não permita trocar a cor.

6. Classe **Robô** com energia

Um robô que realiza tarefas, mas consome energia:

- Atributos: nome, energia (0–100).
- Métodos:
 - `executarTarefa(String tarefa)`: consome 10 de energia.
 - `recarregar()`: enche a energia.
 - `status()`: imprime o nível de energia.

Desafio extra: Não permitir executar tarefa com energia < 10.