### **Apresentação do Projeto: RPG de Turnos no Terminal (Java)**

#### **Introdução**

Este projeto consiste em um jogo de RPG de turnos executado no terminal. Ele foi desenvolvido com a linguagem **Java** e tem como objetivo demonstrar a aplicação de princípios fundamentais da **Programação Orientada a Objetos (POO)**:

* **Herança**: Reutilização de atributos e métodos entre classes.
* **Encapsulamento**: Proteção de atributos usando métodos get e set.
* **Classe Abstrata**: Definição de uma estrutura comum para as classes filhas.

O jogador controla um personagem que participa de combates por turno contra inimigos. A cada vitória, ele ganha experiência e evolui.

### **Explicação das Classes**

* **Personagem (classe abstrata)**  
   Define atributos como nome, hp, ataque, defesa.  
   Possui métodos: atacar() (abstrato), receberDano(), estaVivo().
* **Jogador**  
   Herda de Personagem.  
   Implementa atacar() e inclui sistema de **nível e experiência**.  
   Ao subir de nível, os atributos aumentam.
* **Inimigo**  
   Também herda de Personagem.  
   Sua força é ajustada conforme o nível. Pode conter recompensas no futuro.
* **Combate**  
   Classe com método estático iniciar() que controla o turno do jogador e do inimigo até que um seja derrotado.
* **Main**  
   Inicializa o jogo, cria o jogador, uma sequência de inimigos e executa os combates em ordem.

### **Mecânica do Jogo**

* O jogador digita seu nome ao iniciar.
* Uma série de combates acontece automaticamente.
* A cada vitória, o jogador ganha experiência.
* Ao atingir a experiência necessária, **sobe de nível**.
* Se os pontos de vida chegarem a 0, o jogador perde.
* Se derrotar todos os inimigos, vence o jogo.

### **Conclusão**

Este projeto é uma maneira prática e divertida de aplicar os conceitos de **POO em Java**.  
 Além disso, exercita a lógica de programação e estruturação de projetos em pacotes e classes.

**Possíveis melhorias futuras:**

* Sistema de itens e equipamentos
* Magias e ataques especiais
* Múltiplas escolhas e caminhos na história
* Interface gráfica simples (GUI)