RPG Character Management System Documentation

Introduction

This program is designed to manage and simulate characters for a role-playing game (RPG). It includes functionalities for both player characters (PCs) and non-player characters (NPCs), handling initiatives, perception checks, saving throws, and more.

Classes and Interfaces

Dados Interface

- **Methods**:
  - `d20()`: Simulates a 20-sided dice roll.
  - `d12()`: Simulates a 12-sided dice roll.
  - `d10()`: Simulates a 10-sided dice roll.
  - `d8()`: Simulates an 8-sided dice roll.
  - `d6()`: Simulates a 6-sided dice roll.
  - `d4()`: Simulates a 4-sided dice roll.
  - `d2()`: Simulates a 2-sided dice roll.
  - `d100()`: Simulates a 100-sided dice roll.

Tiradas Interface

- **Methods**:
  - `tirar_iniciativa(int modificador_iniciativa)`: Rolls initiative for characters.
  - `tirar_salvacion_reflejo(int modificador_salvacion_reflejo)`: Rolls reflex saving throw.
  - `tirar_salvacion_voluntad(int modificador_salvacion_voluntad)`: Rolls will saving throw.
  - `tirar_salvacion_fortaleza(int modificador_salvacion_fortaleza)`: Rolls fortitude saving throw.
  - `tirar_percepcion(int modificador_percepcion)`: Rolls perception check.
  - `tirar_tasacion(int modificador_tasacion)`: Rolls appraisal check.

Personaje Class (Abstract)

- **Attributes**:

  - `nombre_personaje`: Character's name.
  - `iniciativa_actual`: Current initiative value.

`PersonajeJugador` Class (Extends `Personaje`)

- **Attributes**:
  - `modificador_iniciativa`: Initiative modifier.
  - `percepcion`: Perception skill value.
  - `tasacion`: Appraisal skill value.
- **Methods**:
  - Implements `Tiradas` interface methods for rolling dice and checks.

`PersonajeNoJugador` Class (Extends `Personaje`)

- **Attributes**:
  - `numero_npc`: NPC identifier.
  - `salvacion_reflejos`, `salvacion_voluntad`, `salvacion_fortaleza`: Saving throw values.
- **Methods**:
  - Implements `Tiradas` interface methods for rolling dice and checks.

`FuncionesUtiles` Class

- **Methods**:
  - `lanzar_iniciativa(ArrayList<Personaje> personajes)`: Rolls initiative for all characters.
  - `ordenar_iniciativa(ArrayList<Personaje> personajes)`: Sorts characters by initiative.
  - `mostrar_iniciativa(ArrayList<Personaje> personajes)`: Displays characters' initiatives.
  - `tirar_percepcion(ArrayList<Personaje> personajes)`: Rolls perception check for all characters.
  - `tirar_tasacion(ArrayList<Personaje> personajes)`: Rolls appraisal check for all characters.
  - Various methods for handling saving throws (`tirada_reflejos`, `tirada_fortaleza`, `tirada_voluntad`).

`CrearObjetosPj` Class

- **Methods**:
  - `listaDePj()`: Creates and returns a list of player characters.

CrearNPC Class

- **Methods**:
  - `crearNPC()`: Prompts user to create a new non-player character with various attributes.

Usage

- **Initialization**:
  - Characters are initialized with specific attributes such as name, initiative, skills, and saving throws.
- **Functionality**:
  - Initiative, perception checks, and saving throws are handled using dice rolls and modifiers.
  - NPCs can be created interactively with user-defined attributes.
  - Characters can be managed, sorted, and displayed based on their initiative values.

Conclusion

This program provides a robust framework for managing and simulating characters in an RPG setting, offering both interactive NPC creation and automated character management functionalities.