



Laboratório 1

Início da Implementação do Jogo Munchkin

MC322 - Programação Orientada a Objetos

Professor: Marcos Raimundo
PEDs: Augusto Cesar / Heigon Soldera
PAD: Matheus Seiji Luna Noda

1. Descrição Geral

Neste semestre, o foco no curso de Programação Orientada a Objetos (MC322) será a implementação de um sistema inspirado no jogo Munchkin. No primeiro laboratório, serão desenvolvidas três classes principais: Jogador, Monstro e Item. Essas classes formarão a base do sistema, que será expandido ao longo do curso com novas funcionalidades à medida que os conceitos forem sendo explorados.

Para representar a estrutura desse sistema, utilizaremos diagramas de classes, um tipo de diagrama UML (Unified Modeling Language). Esses diagramas são essenciais para visualizar a arquitetura de sistemas orientados a objetos, mostrando as classes, interfaces, atributos, métodos e suas relações dentro do sistema. Assim, proporcionam uma visão geral do sistema e como suas diferentes partes interagem entre si.

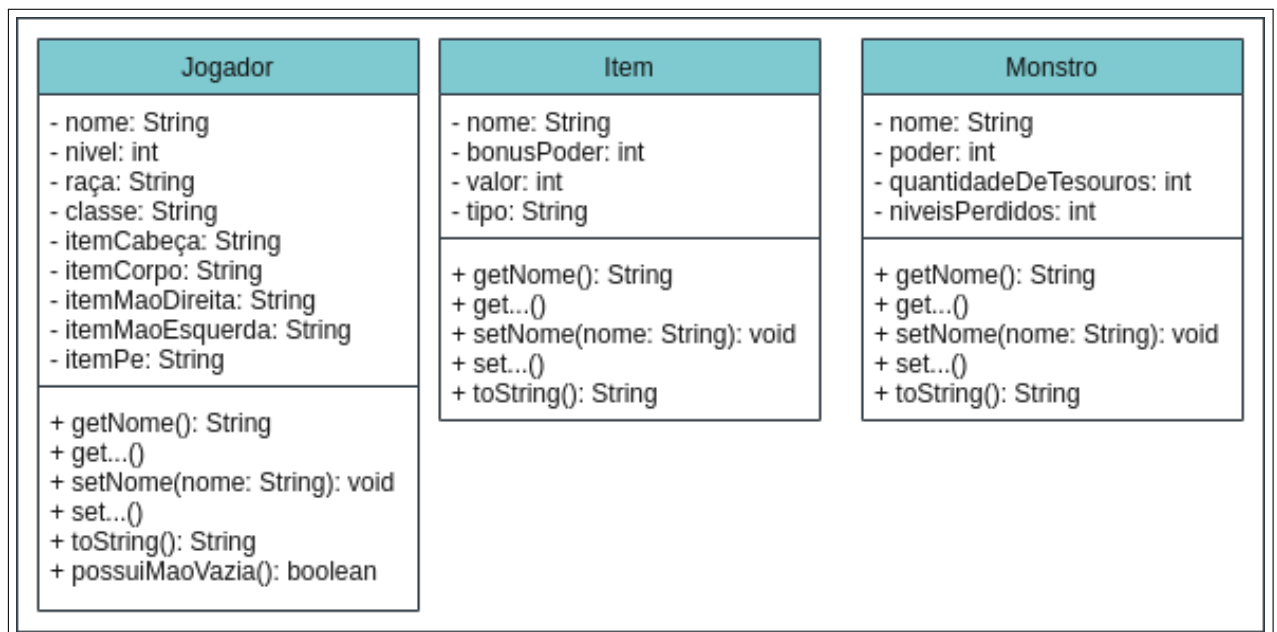


Figura 1: Diagrama de Classe

2. Objetivos

Os principais objetivos deste laboratório consistem em:

- Reconhecimento e familiaridade com o ambiente de desenvolvimento (IDE) e a linguagem a ser utilizada nesta disciplina (Java);
- Desenvolvimento das classes e seus respectivos atributos e métodos;
- Desenvolvimento da visibilidade dos atributos e métodos.

3. Atividades

As seguintes atividades são esperadas nesse laboratório:

- Criação de um projeto na IDE de sua escolha: Dê preferência para o VSCode como IDE, contudo, outras IDEs (Eclipse, Sublime, Atom, etc) podem ser utilizadas;
- Elaboração das classes Jogador, Monstro e Item: seu projeto deve compreender estas três classes além de uma classe Main para a execução do projeto;
- Definição dos métodos, atributos, bem como suas respectivas visibilidades: você deve definir qual visibilidade (*public*, *private*, *protected*) os atributos e métodos possuem;
- Métodos de acesso (*getters e setters*): defina os métodos de acesso aos atributos das classes.
- Métodos de impressão (*toString()*): para visualizar a classe instanciada e seus atributos.
- Método *possuiMaoVazia()*: validar se o jogador possui alguma mão sem item equipado.

4. Descrição das Classes

Classe - Jogador

A classe Jogador representará cada participante do jogo. Ela terá atributos como nome, nível, raça, classe, itens equipados, entre outros.

Classe - Monstro

A classe Monstro será responsável por modelar os inimigos encontrados pelos jogadores durante a exploração das masmorras. Cada monstro terá atributos como nome, nível de poder e tesouro que concede ao ser derrotado.

Classe - Item

A classe Item representará os equipamentos e objetos que os jogadores podem encontrar e utilizar durante o jogo. Cada item terá atributos como nome, tipo (arma, armadura, elmo, etc.), bônus de poder e valor.

5. Avaliação

Além da correta execução do laboratório, os seguintes critérios serão utilizados para a composição da nota do laboratório:

- Entrega realizada dentro do prazo estipulado;
- Qualidade do código desenvolvido (tabulação, comentários);
- Instanciação dos objetos na classe Main;
- Desenvolvimento correto dos métodos de acesso.

6. Entrega

- **A entrega do Laboratório é realizada exclusivamente via Github.**
- Utilize os horários de laboratório e atendimentos para tirar eventuais dúvidas de submissão e também relacionadas ao desenvolvimento do laboratório.
- **Prazo de Entrega:** 19/03 - 19h