

Laboratório 2

Inventário do Jogador, Classes e Raça

MC322 - Programação Orientada a Objetos

Professor: Marcos Raimundo

PEDs: Augusto Cesar / Heigon Soldera

PAD: Matheus Seiji Luna Noda

1. Descrição Geral

No primeiro laboratório construímos algumas classes para nosso jogo e configuramos o ambiente para trabalhar com Java (pelo menos era o objetivo rs). Agora vamos avançar na construção do nosso jogo de cartas adicionando mais algumas classes que compõem a estrutura básica do jogo, junto com algumas funcionalidades e de quebra usar alguns recursos das últimas aulas como herança, composição e visibilidade. Abaixo temos uma sugestão de como deverá ser a estrutura das classes, mas como sempre, sugestões são bem vindas.

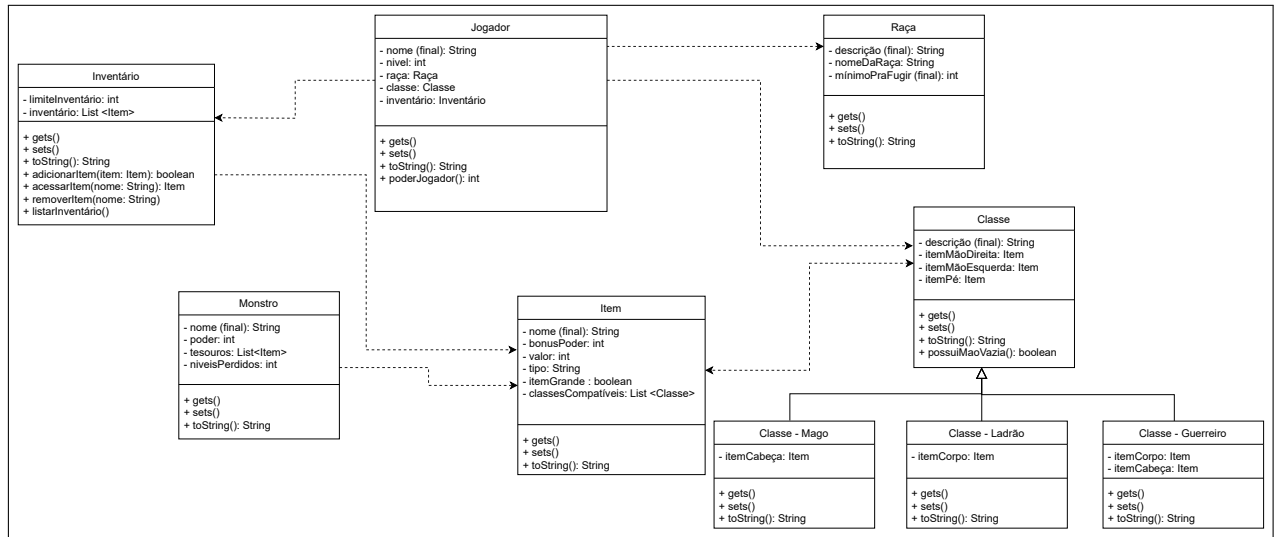


Figura 1: Diagrama de Classe

2. Objetivos

Os principais objetivos deste laboratório consistem em:

- Aplicação do conceito de Herança Simples;
- Utilização do conceito de interface mínima. Tudo que não é necessário para outra classe deve estar *private*, o que necessário para herdeiro deve estar *protected* e o que for necessário para fora da classe deve estar *public*;
- Utilização de Coleção de Dados (ArrayList e LinkedList);
- Utilização de constantes (finals);
- Consolidação dos conteúdos abordados no Laboratório 01 (visibilidade de variáveis e métodos; instanciação de objetos; impressão de objetos utilizando *toString()*)

3. Atividades

As seguintes atividades são esperadas nesse laboratório:

- Elaboração das novas classes e adaptações as classes anteriores.
- Classe Main para a execução do projeto;
- Aplicação do conceito de Herança Simples;
- Utilização de Coleção de Dados (*ArrayList* e *LinkedList*);
- Refinamento dos conteúdos abordados no Laboratório 01 (visibilidade de variáveis e métodos; instanciação de objetos; impressão de objetos utilizando *toString()*)

4. Descrição das Classes

Classe - Jogador

A classe Jogador representará cada participante do jogo. Além de nome e nível, a classe será composta da raça a qual o jogador pertence e a classe escolhida.

Método - poderJogador

O método *poderJogador* calcula o poder de ataque do jogador com base no valor da soma dos itens equipados mais o nível do jogador.

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, composto por uma lista de itens.

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. Uma personagem pode carregar infinitos itens pequenos, mas apenas um “*item grande*”. Além disso, os objetos possuem classes específicas que podem utilizá-los representado em uma lista de classe.

Classe - Raça

Identifica a Raça do Jogador.

Classe - Classe

Essa Classe será responsável por gerenciar os itens equipados do Jogador, tendo ela 3 herdeiros que possuirão um conjunto de itens diferentes permitidos a equipar. Ela é responsável pela verificação no caso de mais de um elemento pesado ser equipado. Impedindo que isso aconteça e apontando um erro.

Classe - Inventário

No inventário deverão ser armazenados todos os itens que o jogador carrega. Ele é responsável pela adição e remoção.

Método - adicionarItem

Método para adicionar item ao inventário. Retorna verdadeiro em caso de sucesso e falso em caso de erro, por exemplo, inventário cheio.

Método - acessarItem

Método para retornar um objeto Item do inventário a partir do atributo nome de Item.

Método - removerItem

Método para remover um objeto Item do inventário a partir do atributo nome de Item.

Método - listarInventário

Método para imprimir todos os itens no inventário.

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**, implemente situações onde:
 - Simule o comportamento do inventário (adicionando e removendo itens)
 - Equipe itens do inventário ao Jogador
 - Impeça que o jogador equipe mais de um item grande ou que sua classe não suporta;
 - Imprima o poder do Jogador baseado em seu nível e itens
- 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:** 02/04 - 19h