



Laboratório 5

Design Pattern

MC322 - Programação Orientada a Objetos

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



1. Descrição Geral

Neste laboratório iremos explorar os Design Patterns: Singleton e MVC.

A classe `BauDeTesouros` agora será um Singleton e as classes `Jogador` e `Masmorra` terão que ser alteradas para o padrão MVC. Dessa forma serão incluídas a `JogadorView` e `MasmorraView` Para organizar as informações para o usuário e a `JogadorController` e `MasmorraController` para realizar a integração entre as Views e as classes Model `Jogador` e `Masmorra` que continuará implementando as regras de negócio.

Singleton: Garante que apenas uma instância da classe `Jogo` exista no programa, otimizando a utilização de recursos e memória.

MVC: Separa a lógica de negócio (Model) da interface do usuário (View) e do controle da interação (Controller), proporcionando maior modularidade, flexibilidade e facilidade de manutenção.

2. Objetivos

Os principais objetivos deste laboratório consistem em:

- Consolidação dos conteúdos vistos nos labs anteriores;
- Design Patterns;

3. Atividades

A principal atividade neste laboratório será adaptar as classes de `BauDeTesouros`, `Jogador` e `Masmorra` utilizando Design Pattern. Para classe `BauDeTesouros`, usaremos o padrão Singleton, que é útil quando temos objetos ou informações que são acessadas em diferentes momentos do nosso programa e não é interessante instanciar-la mais de uma vez.

3.1. Implementando o Padrão Singleton na Classe `BauDeTesouros`

- **Atributo Privado:** Crie um atributo privado estático `instancia` do tipo `BauDeTesouros` para armazenar a instância única da classe.
- **Construtor Privado:** Crie um construtor privado para evitar a instanciação direta da classe fora da classe em si.
- **Método de Acesso:** Crie um método estático (ex. `getInstance()`) que verifica se a instância `instancia` já foi criada. Se não, cria uma nova instância e a retorna. Se já existir, retorna a instância existente.

3.2. Implementando o Padrão MVC nas Classes `Jogador` e `Masmorra`

3.2.1. Criando as Classes View e Controller:

- **JogadorView:** Crie uma classe `JogadorView` para exibir as informações do jogador na interface do usuário. Essa classe deve ter métodos para atualizar a exibição com base nos dados do jogador (nome, vida, itens, etc.).
- **MasmorraView:** Crie uma classe `MasmorraView` para exibir as informações da masmorra na interface do usuário. Essa classe deve ter métodos para atualizar a exibição com base nos dados da masmorra (salas, monstros, itens, etc.).
- **JogadorController:** Crie uma classe `JogadorController` para controlar a interação entre a `JogadorView` e a classe `Jogador` (Model).
- **MasmorraController:** Crie uma classe `MasmorraController` para controlar a interação entre a `MasmorraView` e a classe `Masmorra` (Model).

3.2.2. Descrevendo as Responsabilidades das Classes:

- **Model (Jogador e Masmorra):**
 - Armazena e manipula os dados do jogador e da masmorra.
 - Essa classe não deve ter conhecimento da interface do usuário ou da interação com o usuário.
- **View (JogadorView e MasmorraView):**
 - Exibe os dados do jogador e da masmorra na interface do usuário.
 - Essa classe não deve ter lógica de negócio ou regras do jogo.
- **Controller (JogadorController e MasmorraController):**
 - Controla a interação entre a View e o Model.
 - Atualiza a View com base nos dados do Model.
 - Recebe comandos do usuário e atualiza o Model de acordo.

4. Execução

O jogo manterá a lógica anterior. Atente-se somente a seguir o fluxo esperado, um loop de jogo que faça sentido e possa ser entendido sem vasculhar o código.

5. Avaliação

- Entrega realizada dentro do prazo estipulado;
- Qualidade do código desenvolvido (tabulação, comentários);
- Implementação das adaptações mencionadas.

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:** 04/06 - 19h