

Atividade T4 - ECM251

"RPGs e CLI"

Prof. Murilo Zanini de Carvalho

Novembro|2020

Descrição:

Desenvolver personagens de RPG não é uma tarefa simples, mesmo sendo extremamente prazerosa de se fazer. Alguns exemplos de sistemas que podem criar fichas de RPG são:

- <https://ficha.epicorpg.com.br/>
- <https://newtonrocha.wordpress.com/2012/06/11/tutorial-como-criar-uma-ficha-de-rpg-completavel-que-adiciona-imagens-usando-o-adobe-life-cycle-es2-tio-nitro-tutoriais-doidimais/>
- <https://www.rederpg.com.br/2007/06/02/como-criar-personagens/>

Cada característica escolhida, cada detalhe, envolveu a escolha de uma entre diversas possibilidades. Consumindo uma quantidade quase ilimitada de recursos como criatividade, pesquisa e empenho, estes personagens não podem ser deixados de lado ou esquecidos (ou pior ainda, jogados fora por que o papel que eles estavam escritos estavam muito sujos).

Você decidiu que esses dias ficariam melhores com suas habilidades de código!

Implementação:

Para testar seu sistema, você deve:

- Perguntar o que o usuário quer fazer (criar um novo personagem, alterar um personagem existente, consultar os personagens existentes e deletar algum personagem).
- Os dados dos personagens que devem ser armazenados em cada registro são:
 - Nome do Personagem;
 - Raça;
 - Profissão;
 - Mana;
 - Ataque;
 - Ataque Mágico;
 - Defesa;
 - Defesa Mágica;
 - Velocidade;
 - Destreza;
 - Experiência;
 - Nível atual.

Por hora, a implementação de uma lista de equipamentos e itens é opcional, mas será considerada como pontuação extra na avaliação.

A sua classe main.java deve apenas iniciar o seu sistema e TODA interação com o usuário deve acontecer apenas na sua classe principal do sistema.

Utilize os padrões de projeto que já foram vistos em aula (DAO, Methods Extraction) para a construção da solução. A interface para utilização do sistema deve ser por linha de comando. Para a construção do arquivo de DB do banco de dados SQLite, utilizar: <https://sqliteonline.com/>.

Para utilização da biblioteca de parser de JSON, recomenda-se o uso de: <https://github.com/stleary/JSON-java>

ATENÇÃO: Todos os personagens, lugares, incidentes, organizações e religiões retratados são fictícios.

_Regras:

- A atividade pode ser realizada em duplas (apenas para deixar claro, uma dupla tem no máximo 2 pessoas).
- A cada nova classe ou interface que for criada, comitar ela no gerenciador de código que estiver sendo utilizado.
- Os nomes e os R.A.s da dupla devem estar na classe main.java do sistema.
- Utilizar a documentação do tipo Javadoc para todos os elementos que forem descritos no sistema.
- Notificar o professor que a atividade foi finalizada, enviando para ele uma mensagem na ferramenta de comunicação utilizada na disciplina (Slack). Apenas um dos membros da dupla deve fazer isso. Aguardar o professor confirmar que já clonou o repositório.
- As aulas dos dias 23/11/2020 e 30/11/2020 fica reservadas para o atendimento e retirada de dúvidas durante a elaboração do problema.
- Data de entrega: 01/12/2020

_Boa Atividade!

Critérios de Avaliação Utilizados:

Critério	Peso
Pilar de abstração da Orientação a Objetos.	1
Pilar do encapsulamento da Orientação a Objetos.	1
Pilar da herança da Orientação a Objetos.	1
Pilar do polimorfismo da Orientação a Objetos.	1
Utilização correta das enumerações (enums).	2
Utilização correta das interfaces.	2
Modelos criados para representar os dados do sistema.	6
Implementação do padrão DAO conforme apresentado em sala de aula.	6
Utilização correta das estruturas de dados.	4
Cadastro dos personagens.	4
Consulta no banco local.	4
Atualização de dados cadastrados.	4
Recuperação de dados cadastrados.	4
Boas práticas de programação O.O..	2
Implementação dos modelos dos itens (EXTRA)	2
Implementação dos dados dos itens com os personagens (EXTRA)	4
Implementação dos modelos dos equipamentos (EXTRA)	2
Implementação dos dados dos equipamentos nos personagens (EXTRA)	4

Método de Entrega: Link para o repositório no Github com o projeto.