

MC322/MC336
Segundo semestre de 2015

Laboratório 5

Professor: Fábio Luiz Usberti (fusberti@ic.unicamp.br)

PED: Rafael Arakaki (rafaelkendyarakaki@gmail.com)

1 Objetivo

O objetivo deste laboratório será a familiarização dos requisitos de um Motor para o jogo LaMa (Lacaio & Magias). Além disso, será cobrado o estudo sobre classes abstratas em Java.

2 Atividade

Nesta atividade o aluno deverá preencher o arquivo MotorXXXXXXX.java em suas seções faltantes, principalmente as que dizem respeito à impressão de informações das Jogadas realizadas em cada turno pelos Jogadores.

Procedimentos para montar o projeto:

1. Crie um novo projeto no Eclipse.
2. Baixe o pacote **Lab5.zip** disponibilizado no Ensino Aberto.
3. Crie uma nova classe neste projeto chamada "MotorXXXXXXX"(substituindo "XXXXXX" pelo seu RA), no pacote default.
4. Copie o conteúdo do arquivo disponibilizado src/MotorXXXXXXX.java para ser o conteúdo da classe "MotorXXXXXXX" recém criada.
5. Neste momento irão aparecer diversos erros e warnings no Eclipse.
6. Clique com botão direito no seu projeto, selecione as propriedades, vá em "Java Build Path -> Add External Class Folder", e adicione a pasta "bin" disponibilizada neste pacote.
7. Dê OK e feche as janelas. Neste momento já deve ser possível compilar o projeto sem erros.

Neste laboratório será requerido que o aluno altere **apenas** o método **processarJogada()** da classe MotorXXXXXXX.

Neste método deverá ser impresso com o método **System.out.println()** algumas informações referentes às jogadas realizadas pelos Jogadores. Dependendo do tipo de Jogada, diferentes informações deverão estar presentes. **No mínimo** imprima as seguintes informações para cada Jogada:

1. **ATAQUE:** Imprima qual o ID do lacaio atacando e qual o ID do lacaio que sofre o ataque. Imprima o poder de ataque do lacaio atacante.
2. **LACAI:** Imprima o ID e o nome do lacaio que foi baixado. Imprima também o custo de mana deste lacaio.

3. *MAGIA*: Imprima o nome da magia e o dano mágico. Se for de alvo, imprima também o alvo. Senão apenas descreva que é uma magia de área.
4. *PODER*: Imprima o alvo deste poder heróico, inclusive quando o alvo for o herói do outro jogador.

Como exemplo temos no código a impressão para os lacaio, que está parcialmente realizada conforme as especificações acima. Para esta atividade cabe completar a implementação das impressões e responder às questões propostas.

3 Questões

Foram criadas três questões embutidas no código fonte referentes ao uso de classes e métodos abstratos. Responda-as no início do código, em uma seção de comentários preparada para isso. Responda de maneira sucinta e procure utilizar os termos técnicos aprendidos em sala de aula, quando isto couber.

1. **Q1**: Explique para que serve o método `super()`.
2. **Q2**: Explique qual a diferença em se declarar o escopo das variáveis como *protected* ao invés de *public* ou *private*.
3. **Q3**: Explique qual a diferença de métodos abstratos para métodos não-abstratos.
4. **Q4**: Explique qual a utilidade das classes abstratas.
5. **Q5**: Como você implementaria em seu Motor, em termos de código, para verificar se um lacaio já atacou naquele turno ? Usaria alguma classe da biblioteca do Java ?

4 Extra

Existe uma atividade extra neste laboratório, que não será cobrada, porém é recomendado que se faça para evitar dificuldades nos laboratórios futuros. Pense a respeito da complexidade de requisitos em um Motor, pois este precisará controlar todo o jogo.

Cada Jogada de um jogador precisará ser validada quanto à sua correção (isto é, não infringe nenhuma regra do jogo). Como por exemplo:

- Um lacaio não pode atacar duas vezes no mesmo turno.
- Um lacaio não pode atacar no turno em que for baixado.
- Uma carta magia/lacaio só pode ser utilizada se esta estiver na mão do Jogador.
- Um lacaio que foi morto obviamente não pode atacar.
- ...

Todas estas questões entre outras deverão ser realizadas nos próximos laboratórios. Se necessário, revise o estudo das *Collections* de Java e de demais conceitos já estudados, pois a partir do próximo laboratório teremos esta carga “substancialmente maior” de trabalho a ser realizado.

5 Observações

Atente-se às seguintes observações antes de submeter sua atividade:

- Não submeta nenhum arquivo a não ser MotorXXXXXXX.java (onde XXXXX é o RA do aluno).
- Seu programa deve compilar sem erros ou *warnings*.
- Atenção às especificações mínimas nas impressões requisitadas. Trabalhos que não cumprirem todos os requisitos receberão **no máximo** nota parcial.
- Aproveite este laboratório para observar a complexidade de um Motor e a quantidade de requisitos que serão tratados nos próximos laboratórios (verificação de validade de qualquer jogada feita pelos jogadores).

6 Submissão

Para submeter esta atividade utilize a página da disciplina no Ensino Aberto. Utilize o recurso de portfólio para submeter a atividade. Para isso, crie uma subpasta dentro de seu portfólio com o nome **Lab5** e dentro desta subpasta submeta o arquivo fonte com a sua implementação. Ao submeter, marque a opção “Compartilhado com Formadores” na opção de compartilhamento.