

MC322/MC336
Segundo semestre de 2015

Laboratório 4

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

PED: Rafael Arakaki (rafaelkendyarakaki@gmail.com)

1 Objetivo

O objetivo deste laboratório será a familiarização com a interface *Comparator* do Java e o método para ordenação *sort()* disponibilizado pelas *Collections* do Java.

2 Atividade

Nesta atividade o aluno deverá criar um método **compare()** para comparar objetos **Carta** quanto a diversos critérios.

Para facilitar, crie um novo projeto no Eclipse chamado “Lab4”. Siga as instruções abaixo para importar os arquivos .class e a documentação para seu projeto.

Procedimentos para montar o projeto:

1. Crie um novo projeto no Eclipse.
2. Baixe o pacote **Lab4.zip** disponibilizado no Ensino Aberto.
3. Crie uma nova classe neste projeto chamada "JogadorCompXXXXX"(substituindo “XXXXX” pelo seu RA), no pacote default.
4. Copie o conteúdo do arquivo disponibilizado src/JogadorCompXXXXX.java para ser o conteúdo da classe "JogadorCompXXXXX" recém criada.
5. Faça o mesmo processo para a classe Main.java, copiando o conteúdo disponível em src/Main.java
6. Altere as referências para "JogadorCompXXXXX"(substituindo “XXXXX” pelo seu RA) na classe Main e na classe JogadorCompXXXXX.
7. Neste momento irão aparecer diversos erros e warnings no Eclipse.
8. 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.
9. 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 **compare()** e o atributo **modoComparacao** da classe JogadorCompXXXXX.

O objetivo é implementar corretamente a interface **Comparator** na classe JogadorCompXXXXX para que esta esteja apta a ordenar um ArrayList de Cartas através de diversos critérios. Os critérios são definidos por enumeração no atributo **modoComparacao** e são os seguintes:

1. *ComparaCartas.COMPARA_VIDA*: Ordena as cartas por vida em ordem **decrecente**. (Cartas de magias têm zero de vida).

2. *ComparaCartas.COMPARA_MANA*: Ordena as cartas por custo de mana em ordem **crescente**.
3. *ComparaCartas.COMPARA_ATAQUE*: Ordena as cartas por ataque em ordem **decrecente**. (Cartas de magias têm zero de ataque).
4. *ComparaCartas.COMPARA_TIPO*: Ordena as cartas pelo seus atributos **TipoCarta**. A ordem é de magias antes e lacaio depois.
5. *ComparaCartas.COMPARA_PERSONALIZADO*: Uma ordem hierárquica com diversos desempates entre as cartas.

A interface Comparator requer que seja implementado o método *compare()*, que toma como entrada dois objetos que serão comparados e tem como saída um inteiro que identifica qual dos objetos tem precedência sobre o outro na ordenação (isto é, qual vem antes de qual). O valor retornado deve ser um inteiro negativo, zero ou positivo para identificar que o primeiro objeto deve vir antes, empate ou depois do segundo objetivo respectivamente.

Segue abaixo um exemplo de como seria uma ordenação de Integers de modo crescente:

```
1 public int compare(Integer a, Integer b) {
2     return a.intValue() - b.intValue();
3 }
```

A atividade consiste em implementar os modos de ordenação através do método **compare()**. Como exemplo, no arquivo disponibilizado se encontra a ordenação decrescente por ataque (*ComparaCartas.COMPARA_ATAQUE*). Portanto a tarefa é implementar os outros quatro modos de ordenação.

Segue a especificação do modo de comparação personalizado:

1. Primeiro são ordenados por **TipoCarta**, com todas as magias antes dos lacaio.
2. Em seguida, dentre as cartas de magia, têm precedência aquelas que são de área e então depois as que são de alvo (ordenação por **TipoMagia**).
3. Por fim, dentre as cartas de magia que tiverem o mesmo **TipoMagia** deverão ser desempatadas pelo custo de mana crescentemente.
4. Para as cartas lacaio, devem ser ordenadas primeiramente pelo custo de mana crescentemente.
5. Por fim, dentre as cartas lacaio que tiverem o mesmo **custo de mana** deverão ser desempatadas pelo ataque decrescentemente.

Para facilitar a depuração do código, o jogo foi modificado para que cada jogador inicie com 10 cartas na mão.

Após implementar (e testar) todas as ordenações acima, submeta seu arquivo JogadorCompXXXXX.java com o modo de comparação personalizado (isto é, o atributo modoComparacao deve ter valor ComparaCartas.COMPARA_PERSONALIZADO).

3 Exemplos

Abaixo se encontra um exemplo das cartas 10 cartas do Baralho ordenadas pelo ataque (*ComparaCartas.COMPARA_ATAQUE*):

```
1 [Dragao (Atq = 7, Vida = (7/7)). Mana = 7
2 , Mago Aprendiz (Atq = 5, Vida = (1/1)). Mana = 3
3 , Gigante de Gelo (Atq = 5, Vida = (4/4)). Mana = 4
4 , Gigante de Pedra (Atq = 4, Vida = (5/5)). Mana = 4
5 , Gnomo (Atq = 2, Vida = (1/1)). Mana = 1
6 , Guerreiro Espadachim (Atq = 2, Vida = (3/3)). Mana = 2
7 , Recruta (Atq = 1, Vida = (2/2)). Mana = 1
8 , Rajada Congelante (3 dano em Alvo). Mana = 2
9 , Onda de Choque (1 dano em Area). Mana = 2
10 , Raio (7 dano em Alvo). Mana = 5
11 ]
```

Abaixo se encontra o mesmo exemplo agora ordenado pela comparação personalizada (*ComparaCartas.COMPARA_PERSONALIZADO*):

```
1 [Onda de Choque (1 dano em Area). Mana = 2
2 , Rajada Congelante (3 dano em Alvo). Mana = 2
3 , Raio (7 dano em Alvo). Mana = 5, Gnomo (Atq = 2
4 , Vida = (1/1)). Mana = 1, Recruta (Atq = 1, Vida = (2/2)). Mana = 1
5 , Guerreiro Espadachim (Atq = 2, Vida = (3/3)). Mana = 2
6 , Mago Aprendiz (Atq = 5, Vida = (1/1)). Mana = 3
7 , Gigante de Gelo (Atq = 5, Vida = (4/4)). Mana = 4
8 , Gigante de Pedra (Atq = 4, Vida = (5/5)). Mana = 4
9 , Dragao (Atq = 7, Vida = (7/7)). Mana = 7
10 ]
```

4 Observações

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

- Por fim, submeta sua classe JogadorCompXXXXX com o modo de comparação personalizado (isto é, o atributo modoComparacao deve ter valor ComparaCartas.COMPARA_PERSONALIZADO).
- Não submeta nenhum arquivo a não ser JogadorCompXXXXX.java (onde XXXXX é o RA do aluno).
- Seu programa deve compilar sem erros ou *warnings*.
- Seu método compare() deve operar conforme as especificações. Atenção à questão da ordem ser crescente ou decrescente conforme foi pedido no enunciado para cada item.
- Não altere nenhuma seção da classe JogadorCompXXXXX com exceção do método **compare()** e do atributo ComparaCartas.

5 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 **Lab4** e dentro desta subpasta submeta o arquivo fonte com a sua implementação. Ao submeter, marque a opção “Compartilhado com Formadores” na opção de compartilhamento.