

**MC302**  
Primeiro semestre de 2017

**Laboratório 5**

**Professores:** Esther Colombini (esther@ic.unicamp.br) e Fábio Luiz Usberti (fusberty@ic.unicamp.br)

**PEDs: (Turmas ABCD)** Elisangela Santos (ra149781@students.ic.unicamp.br), Lucas Faloni (lucasfaloni@gmail.com), Lucas David (lucasolivdavid@gmail.com), Wellington Moura (wellington.tylon@hotmail.com)

**PEDs (Turmas EF)** Natanael Ramos (naelr8@gmail.com), Rafael Arakaki (rafaelkendyarakaki@gmail.com)

**PAD: (Turmas ABCD)** Igor Torrente (igortorrente@hotmail.com)

**PAD: (Turmas EF)** Bleno Claus (blenoclaus@gmail.com)

---

## 1 Objetivo

O objetivo deste laboratório será a familiarização com as classes e métodos disponibilizados no Trabalho 1 e a criação de um Jogador com procedimentos específicos.

## 2 Atividade

Nesta atividade o aluno deverá criar uma classe Jogador que irá executar jogadas especificadas no enunciado.

Para facilitar, crie um novo projeto no Eclipse chamado “Lab5”. 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 Código de Apoio** no moodle ou na página da disciplina.
3. Copie para o pacote padrão (default) o arquivo Main.java disponibilizado no código de apoio. É normal que após a inclusão do arquivo Main.java apareçam diversos erros e warnings.
4. Clique com botão direito no seu projeto, selecione Propriedades, vá em “Java Build Path -> Add External Class Folder”, e adicione a pasta “bin” disponibilizada no código de apoio.
5. Clique na setinha ao lado esquerdo do item que foi criado (logo abaixo da frase “JARs and class folders on the build path”).
6. Escolha a opção Javadoc location, navegue até a pasta “doc” disponibilizada neste pacote e selecione-a.
7. Dê OK e feche as janelas.
8. Copie a classe JogadorRAxxxxxx do código de apoio para o projeto.

Você deve ter os arquivos Main.java e *JogadorRAxxxxxx.java* no projeto. Note que deve-se renomear a classe do jogador **JogadorRAxxxxxx** substituindo “xxxxxx” pelo seu RA (renomeie o arquivo e o construtor também). A partir destas mudanças já deverá ser possível compilar e executar o projeto, porém a classe **JogadorRA** não realizará nenhum movimento e irá perder o jogo para a classe **JogadorAleatorio**.

Neste laboratório será requerido que o aluno implemente alguns procedimentos específicos. **Atente-se a não fugir das especificações contidas neste enunciado.** Implementações que funcionarem porém não seguirem as especificações receberão nota parcial na correção da atividade.

O seu Jogador deve criar, em ordem, no **List<Jogada>** que é devolvido durante o método **processarTurno()**, as seguintes Jogadas:

1. **Se houver mana disponível** para alguma carta de **Magia**, deverá utilizar esta carta de magia. Se a magia for de alvo, deverá mirar no herói do oponente (qualquer carta magia que a mana permita). Utilize no máximo uma magia.
2. **Se houver mana disponível** para alguma carta de **Lacaio**, deverá baixar à mesa esta carta de **Lacaio** (qualquer carta lacaio que a mana permita). Baixe no máximo um lacaio.
3. Por fim, deverá atacar com todos os lacaios vivos na mesa e que não foram baixados neste turno no herói do oponente.

Note que só se deve realizar os procedimentos de utilizar magia e baixar lacaio apenas uma vez por turno, portanto este Jogador nunca deve baixar duas cartas lacaios ou utilizar duas magias no mesmo turno (note que isto é um requisito desta atividade e não uma regra do jogo). Outro padrão para este Jogador é que este prioriza Magias em detrimento de uso de Lacaios.

Se for realizado alguma **Jogada** ilegal no jogo (segundo as regras) o **Motor** irá disparar uma **Exception** com um erro correspondente. Faz parte desta atividade criar um código que somente faça jogadas válidas. Se você ainda não leu o enunciado do Trabalho 1, leia antes de realizar este laboratório para saber as regras do jogo.

### 3 Exemplos de jogadas

Nesta seção se encontram exemplos de jogadas que você pode se utilizar como referência.

```
1 //card = objeto Carta do lacaio que quero baixar
2 //null = nao ha alvo nesta jogada.
3 Jogada j = new Jogada(TipoJogada.LACAI0, card, null);
4 minhasJogadas.add(j);
```

Listing 1: Exemplo de **Jogada** para baixar um lacaio à mesa.

```
1 //card = objeto Carta da magia (de alvo) que quero usar
2 //null = para mirar no heroi do oponente
3 Jogada j = new Jogada(TipoJogada.MAGIA, card, null);
4 minhasJogadas.add(j);
```

Listing 2: Exemplo de **Jogada** para usar uma magia no Herói do oponente.

```
1 // card = objeto Carta da magia (de area) que quero usar
2 // null = nao ha alvo especifico nesta jogada
3 Jogada j = new Jogada(TipoJogada.MAGIA, card, null);
4 minhasJogadas.add(j);
```

Listing 3: Exemplo de **Jogada** para usar uma magia em área.

```
1 // card = objeto Carta da magia (de buff) que quero usar
2 // lacaio = objeto Carta do lacaio o qual quero "buffar"
3 // null = nao ha alvo especifico nesta jogada
4 Jogada j = new Jogada(TipoJogada.MAGIA, card, lacaio);
```

```
5 minhasJogadas.add(j);
```

Listing 4: Exemplo de **Jogada** para usar uma magia de buff.

```
1 // card = objeto Carta de meu lacaio de ataque.
2 // null = atacando o heroi do oponente.
3 Jogada j = new Jogada(TipoJogada.ATAQUE, card, null);
4 minhasJogadas.add(j);
```

Listing 5: Exemplo de **Jogada** para atacar o herói do oponente com um lacaio.

```
1 int minhaMana = primeiroJogador ? mesa.getManaJog1() : mesa.getManaJog2();
2
3 for (int i = 0; i < mao.size(); i++) {
4     Carta carta = mao.get(i);
5
6     if (carta instanceof Lacaio && card.getMana() <= minhaMana) {
7         Jogada j = new Jogada(TipoJogada.LACAI0, carta, null);
8         minhasJogadas.add(j);
9         minhaMana -= carta.getMana();
10
11         System.out.println("Jogada: Baixei um lacaio: " + carta);
12         mao.remove(i);
13         i--;
14         break;
15     }
16 }
```

Listing 6: Exemplo de como percorrer as cartas da mão em ordem e baixar à mesa o primeiro **Lacaio** que a Mana permitir.

```
1 List<Carta> meusLacaio = primeiroJogador
2                       ? mesa.getLacaioJog1()
3                       : mesa.getLacaioJog2();
4
5 for (int i = 0; i < meusLacaio.size(); i++) {
6     Carta lacaioVivo = meusLacaio.get(i);
7     System.out.println("Um lacaio vivo descido em turnos anteriores: "
8                       + lacaioVivo);
9 }
```

Listing 7: Exemplo de utilização da **Mesa** para verificar quais lacaio estão vivos e prontos para atacar.

## 4 Tarefas

Realize as tarefas a seguir:

1. Execute 10.000 partidas colocando em combate seu jogador (JogadorRAXXXXXX.java) e o JogadorAleatorio.java utilizando o método **multiplasPartidas** da classe Main. Verifique e mostre a taxa de vitória de seu jogador.
2. Crie um novo jogador com base no JogadorRAXXXXXX.java chamado **JogadorRAXXXXXX-New.java**. Altere o novo jogador com o intuito de melhorar a taxa de vitórias. Escreva um relatório contendo as taxas de vitórias dos jogadores e um resumo (max 10 linhas) de qual foi a nova tática utilizada. O relatório deve estar em um arquivo texto que deverá ser submetido juntamente com o código.

## 5 Observações

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

- Não submeta nenhum arquivo a não ser: `relatorio.txt`, `Jogador{RA}.java`, `Jogador{RA}New.java`, `Main.java`.
- Seu programa deve compilar sem erros ou warnings.
- Sua classe `Jogador` deve somente jogadas válidas e conforme as especificações deste laboratório.
- Uma interface gráfica criada para que os alunos se familiarizem com o jogo está disponível. Para utilizá-la, descomente a linha 37 `//umaPartidaGUI(baralho1, baralho2, verbosidade, tempoLimite);` no método estático `Main.main` e comente a chamada para `Main.umaPartida` logo abaixo. Entretanto, esta mesma se encontra em fase de desenvolvimento e ainda pode ser alterada em futuras versões. Por isso, alunos são encorajados a sempre baixar o pacote mais recente da versão do Motor no moodle.

## 6 Submissão

Para submeter a atividade, utilize o Moodle: <https://www.ggte.unicamp.br/ea>.

1. Compacte o relatório e o jogador desenvolvido no formato `.zip` ou similar e nomeie-o **Lab5-000000.zip**, trocando `'000000'` pelo seu número de RA.
2. Submeta o arquivo compactado através do moodle da disciplina.

### Datas de entrega

- Dia **17/04** Turma **ABCD** até às 23:55h
- Dia **14/04** Turma **EF** até às 23:55h