MC302
Primeiro semestre de 2022

Laboratório 2

---

1 Objetivo

Este laboratório envolve a prática dos conceitos de criação e manipulação de objetos, visibilidade, sobrecarga e métodos.

2 Atividade

Nesta atividade o foco será a manipulação de objetos das classes **CartaLacaio** e **CartaMagia**. A primeira tarefa será configurar o ambiente com a criação de um novo projeto.

Os seguintes passos **devem** ser tomados para a criação do projeto:

1. Abra o Eclipse.
2. Crie um novo projeto (File -> New -> Java Project).
3. Digite o nome do projeto **Lab2RAxxxxxx** onde **xxxxxx** deve ser substituído pelo seu RA.
4. Não inclua a opção “Create modulo-info.java file"(desmarque se estiver marcado).
5. Clique em Finish.

Cole as duas classes do laboratório 1 (**CartaLacaio.java e CartaMagia.java**) dentro do projeto que foi criado (na pasta **src**).

3 Classe CartaLacaio

A classe CartaLacaio, criada no laboratório 1, deste laboratório é baseada em um jogo de cartas de computador chamado Hearthstone[1] ©. Nesse jogo existem cartas do tipo *Lacaio* que possuem atributos como ataque e vida distintos para cada carta.

Você deverá construir dois novos construtores para a classe CartaLacaio:

O primeiro construtor irá instanciar apenas os atributos ID, nome e custoMana. Esse construtor será denominado **construtor reduzido**.

O segundo construtor irá copiar todos os atributos de um objeto da classe CartaLacaio passado por parâmetro. Esse construtor será denominado **construtor cópia**.

```
1 public CartaLacaio(int ID, String nome, int mana) {}
2 public CartaLacaio (CartaLacaio origem) {}
```

novos_construtores

[1] http://us.battle.net/hearthstone/pt

```java
public class CartaLacaio {

    private int ID;
    private String nome;
    private int ataque;
    private int vidaAtual;
    private int vidaMaxima;
    private int custoMana;

    public CartaLacaio(int ID, String nome, int ataque, int vida, int mana) {
        this.ID = ID;
        this.nome = nome;
        this.ataque = ataque;
        this.vidaAtual = vida;
        this.vidaMaxima = vida;
        this.custoMana = mana;
    }

    //Novos construtores aqui

    //Outros metodos aqui

}
```

CartaLacaio.java

4 Classe Main

Crie uma nova classe através do Eclipse chamada Main e escolha a opção para gerar automaticamente o método main. No método main instancie objetos do tipo CartaLacaio e CartaMagia conforme a seguir.

```java
// -- dentro da classe Main
    public static void main(String[] args) {
        // instanciando objetos
        CartaLacaio lac1 = new CartaLacaio(1, "Frodo Bolseiro", 2, 1, 1);
        CartaLacaio lac2 = new CartaLacaio(2, "Aragorn", 5, 7, 6);
        CartaLacaio lac3 = new CartaLacaio(3, "Legolas", 8, 4, 6);
        CartaMagia mag1 =  new CartaMagia(4, "You shall not pass", 4, true, 7);
        CartaMagia mag2 =  new CartaMagia(5, "Telecinese", 3, false, 2);

    }
```

instancias

5 Tarefas

Dentro da pasta onde está este lab (pasta LAB2RAxxxxxx), crie um arquivo texto simples chamado respostas.txt e inclua as respostas para as perguntas abaixo:

1. Instancie um objeto CartaLacaio referenciado pela variável **lac4** através do **construtor reduzido**. Imprima na tela o estado do objeto lac4. O que acontece com os atributos que não foram inicializados pelo construtor? Por que isso acontece?

2. Altere o ataque do objeto lac1. Faça com que ele receba o ataque do objeto lac3. Imprima o estado de lac1.

3. Comente o método **toString()** implementado na classe CartaMagia. O que ocorre quando é executada a impressão do estado de um objeto CartaMagia?

4. Instancie um novo objeto lac5 utilizando o **construtor cópia** passando o conteúdo de lac2 como argumento. Imprima o estado dos objetos lac2 e lac5. Há alguma diferença entre esses dois objetos?

5. Altere a visibilidade do atributo **nome** na classe CartaMagia para *public*. Execute os comandos a seguir na classe main. na classe Main

```
na classe Main
System.out.println(mag1.nome);
System.out.println(mag1.getNome());na classe Main
```

Comandos na classe Main

   Tente fazer o mesmo com a visibilidade *private* aplicada ao atributo nome. O que acontece e por que? Qual o objetivo de utilizarmos atributos privados e os métodos get() em set() para cada atributo?

6. Crie na classe CartaLacaio dois métodos sobrecarregados chamados **buffar**. O termo "buffar" é um neologismo do termo em inglês "buff", utilizado para denominar um aprimoramento. O comportamento expresso pelos métodos **buffar** será de aprimorar os atributos de uma carta.

   - O primeiro método (void buffar(int a)) deve receber um inteiro que representa o quanto cada atributo da carta irá se fortalecer. A função deve aumentar os atributos ataque e vida no valor passado por parâmetro.
   - O segundo método (void buffar(int a, int v)) deve receber dois inteiros: o primeiro irá aumentar o ataque e segundo irá aumentar a vida da carta.
   - Os dois métodos sobrecarregados devem alterar o nome da carta da seguinte forma: nomeAnterior + " Buffed". Essa etapa deve ser realizada a partir de um método "void alteraNomeFortalecido()"que você deverá criar.
   - **Obs**: Verifique se os argumentos passados nos métodos buffar realmente irão "buffar"a carta. Caso contrário, não faça nenhuma alteração. Teste os dois métodos sobrecarregados "buffando"o ataque e vida de pelo menos duas cartas, imprimindo os atributos "buffados".

6 Submissão

Submeta no google classroom a pasta do projeto comprimida como um arquivo zip, de tal forma que o arquivo se chame Lab2RAxxxxxx.zip (xxxxxx deve ser substituído pelo seu RA). A pasta deve ter sido preparada conforme especificado na Seção 2. Não se esqueça de incluir na pasta o arquivo respostas.txt.