

Curso de Bacharelado de Sistemas de Informação

Disciplina: **Programação Orientada a**

Professor: **Giovany Frossard Teixeira**

Objetos I

Observações: 1ª Prova

Data:

Aluno:

Nota:

1ª Questão (40 pontos) Implemente as classes **Equipe** (12 pontos), **Soccer2D** (4 pontos), **SEK** (4 pontos), **Asimov2D** (5 pontos), **ITADroids** (5 pontos), **AsimovSEK** (5 pontos) e **México** (5 pontos) para que a classe **LARC_2016** possa funcionar corretamente.

Restrições:

- Equipes tem quantidade máxima de membros, quantidade atual de membros, desempenho padrão e quantidade de partidas disputadas;
- Por padrão toda Equipe tem desempenho padrão 1 e quantidade de partidas disputadas 0;
- Não é permitido instanciar objetos das classes Equipe, Soccer2D ou SEK;
- Equipes sabem jogar e no comportamento padrão (classe Equipe) apenas incrementam o número de partidas disputadas por ambos times (nesse contexto retorna "");
- Soccer2D e SEK são tipos de Equipes;
- Asimov2D e ITADroids são equipes de Soccer2D;
- AsimovSEK e México são equipes de SEK;
- Toda vez que Asimov2D joga seu desempenho é incrementado em 1 unidade;
- Quando AsimovSEK joga com 3 robôs tem seu desempenho incrementado em 3 unidades;
- ITADroids tem desempenho padrão 3;
- México tem desempenho padrão 5;
- Toda vez que joga México tem seu desempenho decrementado em 1 unidade;
- Toda vez que uma equipe joga ela precisa definir o vencedor e isso é feito pela chamada do método definirVencedor no método jogar.

```
package prova20161;
```

```
public class LARC_2016 {
```

```
    public static void main(String[] args) {
```

```
        /* Toda Equipe tem quantidade de membros atual e a quantidade máxima
```

```
        esses são os 2 últimos parâmetros dos construtores abaixo
```

```
        Equipes Soccer2D também tem o caminho da logo (primeiro parâmetro)
```

```
        Equipes SEK também tem a quantidade de robôs (primeiro parâmetro)
```

```
        Ex: equipe1 tem caminho c:/ e equipe2 tem 3 robos
```

```
        */
```

```
        Asimov2D equipe1 = new Asimov2D("c:/", 7, 10);
```

```
        AsimovSEK equipe2 = new AsimovSEK(3, 6, 6);
```

```

ITADroids equipe3 = new ITADroids("c:/ita", 6, 8);
Mexico equipe4 = new Mexico(1, 4, 5);

System.out.println(equipe1.jogar(equipe3));
System.out.println(equipe1.jogar(equipe3));
System.out.println(equipe1.jogar(equipe3));
System.out.println(equipe1.jogar(equipe3));

System.out.println(equipe2.jogar(equipe4));
System.out.println(equipe2.jogar(equipe4));
System.out.println(equipe2.jogar(equipe4));
System.out.println(equipe2.jogar(equipe4));
}

```

Na classe **Equipe** DEVE ser definido o seguinte método:

```

protected static String definirVencedor(Equipe equipe1, Equipe equipe2){
    double valorEquipe1 = Math.random();
    double valorEquipe2 = Math.random();
    if((valorEquipe1 * equipe1.getDesempenho()) > valorEquipe2 * equipe2.getDesempenho())
        return equipe1.getClass().getSimpleName() + " VENCEU " + equipe2.getClass().getSimpleName();
    else
        return equipe1.getClass().getSimpleName() + " PERDEU para " + equipe2.getClass().getSimpleName();
}

```

Exemplo de Saída (por conta do comportamento randômico esses valores podem mudar):

```

Asimov2D VENCEU ITADroids
Asimov2D VENCEU ITADroids
Asimov2D VENCEU ITADroids
Asimov2D VENCEU ITADroids
AsimovSEK PERDEU para Mexico
AsimovSEK VENCEU Mexico
AsimovSEK VENCEU Mexico
AsimovSEK VENCEU Mexico

```

Obs1. Ao final da prova compactar o projeto java contendo todas as classes no arquivo <nome_aluno>.zip e enviar para o ambiente Moodle da disciplina no espaço disponibilizado pelo professor.

Obs2. Mudanças na classe LARC_2016 podem acarretar perda de pontos, de 0 a 40 pontos dependendo da mudança promovida. Ou seja, não é permitido alterar a especificação da prova.

Boa Prova !!!