

Atividade Programação Orientada a Objetos

1) Construa uma classe `Jogador` contendo atributos para armazenar o nome, a quantidade de gols, a quantidade de chutes e um inteiro representando o tipo do chute (0 para representar chutes no meio do gol, 1 para chutes no canto esquerdo e 2 para chutes no canto direito). Implemente nesta classe um método construtor que receba como parâmetro o nome do jogador e um int representando o tipo de chute. Todo objeto instanciado da classe `Jogador` deve estar com a quantidade de gols e chutes zeradas. Crie um método de acesso (get) para o atributo do tipo de chute. Crie dois métodos `chuta()` e `fazGol()`, respectivamente para incrementar os chutes realizados e gols marcados pelo jogador. Crie um método `imprime()` que imprima os valores dos atributos. **(1.25 ponto)**

2) Construa uma classe `Goleiro` contendo atributos para armazenar o nome, a quantidade de defesas, a quantidade de tentativas de defesas e um inteiro representando o tipo da defesa (0 para representar fica parado no meio do gol, 1 para defesas no canto esquerdo e 2 para defesas no canto direito). Além disso, um atributo deve ser definido para armazenar se o goleiro se adiantou ou não, no momento da defesa. Implemente nesta classe um método construtor que receba como parâmetro o nome do goleiro. Todo objeto instanciado da classe `Goleiro` deve estar com a quantidade de defesas e tentativas zeradas, tipo da defesa para o meio do gol, e não deve estar adiantado. Crie métodos de acesso (get) para o atributo do tipo de defesa e para o atributo que indica se o goleiro está adiantado. Crie dois métodos `pula()` e `defendeu()`, respectivamente para incrementar as tentativas realizadas e as defesas com sucesso. Crie um método `prepara()` que modifica o tipo da defesa (0 para 1, 1 para 2, 2 para 0) sendo que sempre o goleiro se adianta quando o tipo de defesa for 0 (meio do gol), caso contrário ele não se adiantará. Crie um método `imprime()` que imprima os valores dos atributos. **(1.25 ponto)**

3) Construa uma classe `Juiz` contendo um atributo para armazenar o nome e um atributo para armazenar se o juiz é ladrão ou não. Implemente nesta classe um método construtor que receba como parâmetro o nome do juiz e um valor lógico representando se ele é ladrão ou não. Crie um método `boolean validaCobranca(Goleiro goleiro)` que receba um objeto `Goleiro` e verifique se ele se adiantou ou não, retornando true ou false. Crie um método `imprime()` que imprima os valores dos atributos. **(1.25 ponto)**

4) Construa uma classe `DisputaPenaltis` contendo atributos para armazenar a quantidade de cobranças realizadas e a quantidade de gols anulados. Todo objeto instanciado da classe `DisputaPenaltis` deve estar com a quantidade de cobranças e gols anulados zeradas. Crie um método `void realizaCobranca(Jogador batedor, Goleiro goleiro, Juiz juiz)` que realize a tarefa da cobrança do pênalti (goleiro se prepara, goleiro pula, batedor chuta, juiz valida a cobrança, atualiza-se os gols marcados ou defesas realizadas). Crie um método `imprime()` que imprima os valores dos atributos. **(1.25 ponto)**

```
public class Prog
{
    public static void main(String[] args)
    {
        Jogador ronaldo = new Jogador("Ronaldo",0);
        Goleiro marcos = new Goleiro("Marcos");
        Juiz joao = new Juiz("Joao",false);
        DisputaPenaltis disputa = new DisputaPenaltis();

        disputa.realizaCobranca(ronaldo, marcos, joao);
        ronaldo.imprime();
        marcos.imprime();
        joao.imprime();
        disputa.imprime();
    }
}
```

Exemplo de uma classe principal para execução (simulação das classes)