

Curso POO Teoria [#07a](#) - Relacionamento entre Classes



```
+ getCategory()  
+ getVitórias()  
+ getDerrotas()  
+ getEmpates()  
+ setNome()  
+ setNacionalidade()  
+ setIdade()  
+ setAltura()  
+ setPeso()  
+ setCategoria()  
+ setVitórias()  
+ setDerrotas()  
+ setEmpates()
```

Agregação:

A agregação é uma forma de associação caracterizada por um relacionamento do tipo parte-todo, em que uma das pontas da

associação é a parte é o todo.

Por exemplo as classes Turma, Estudante e Instrutor, o Estudante é ligado a Turma, o Instrutor também é ligado a Turma.

Ou seja Turma é um todo, enquanto os Estudantes e o Instrutor são partes do todo.

Classe Lutador

//Atributos

privado nome: Caractere

privado nacionalidade: Caractere

privado idade : Inteiro

privado altura: Real

privado peso: Real

privado categoria: Caractere

privado vitorias: Inteiro

privado derrotas: Inteiro

privado empates: Inteiro

```
publico Metodo Constructor(no: Caractere
                           na: Caractere
                           id: Inteiro
                           al: Real
                           pe: Real
                           vi: Inteiro
                           de: Inteiro
                           em: Inteiro)
```

```
publico Metodo apresentar()
```

```
FimMetodo
```

```
publico Metodo status()
```

```
FimMetodo
```

```
publico Metodo ganharLuta()
```

```
FimMetodo
```

```
publico Metodo perderLuta()
```

```
FimMetodo
```

```
publico Metodo empatarLuta()
```

```
FimMetodo
```

FimClasse

Publico Metodo getNome()

Retorne nome

FimMetodo

Publico Metodo setNome(no:Caractere)

nome = no

FimMetodo

Publico Metodo getNacionalidade()

Retorne nacionalidade

FimMetodo

Publico Metodo setNacionalidade(na: Caractere)

Nacionalidade=na

FimMetodo

Publico Metodo getIdade()

Retorne idade

FimMetodo

Publico Metodo setIdade(id: inteiro)

Idade = id

FimMetodo

Publico Metodo getAltura()

Retorne altura

FimMetodo

Publico Metodo setAltura(al: Real)

Altura = al

FimMetodo

Publico Metodo getPeso()

Retorne peso

FimMetodo

Publico Metodo setPeso(pe: Real)

Peso = pe

setCategoria()

FimMetodo

Publico Metodo getCategoria()

Retorne peso

FimMetodo

Publico Metodo setCategoria()

Se(peso<52.2)então

Categoria = “ Inválido ”

Senão se (peso<=70.3) então

Categoria = “Leve”

Senao se (peso<= 83.9) então

Categoria = “Médio”

Senao se(peso<=120.2)então

Categoria = “ Pesado ”

Senao Categoria = “Inválida”

Fimse

FimMetodo

FimClasse

FimMetodo

Publico Metodo getVitorias()

Retorne vitorias

FimMetodo

Publico Metodo setVitorias(vi: Inteiro)

Vitorias = vi

FimMetodo

Publico Metodo getDerrotas()

Retorne derrotas

FimMetodo

Publico Metodo setDerrotas(vi: Inteiro)

Derrotas = de

FimMetodo

Publico Metodo getEmpates()

Retorne empates

FimMetodo

Publico Metodo setEmpates(em : Inteiro)

Empates = em

FimMetodo

Publico método apresentar()

Escreva ("Lutador: ", getNome())

Escreva ("Origem: ", getNacionalidade())

Escreva (getIdade(), "anos")

Escreva (getAltura(), " m de altura")

```

Escreva (" Pesando: ", getPeso(), "Kg")
Escreva ("Ganhou: ", getVitorias())
Escreva ( "Perdeu:", getDerrotas())
Escreva ("Empatou:", getEmpates())
FimMetodo

Publico Metodo status()

Escreva (getNome())

Escreva("é um peso ", getCategory())

Escreva (getVitorias(), "vitórias")
Escreva (getDerrotas(), "derrotas")
Escreva (getEmpates(), "empates")
FimMetodo()

// Programa Principal

L: vetor [0 ... 5]

L[0]: novo Lutador ("PrettyBoy",
                    "França" ,
                    31, 1.75 , 68.9 , 11 , 3 , 1)

L[1]: novo Lutador ("Putscript",
                    "Brasil" ,
                    29, 1.68 , 57.8 , 14 , 2, 3)

L[2]: novo Lutador ("Snapshadow",
                    "EUA" ,
                    35, 1.65 , 80.9 , 12 , 3,1)

L[3]: novo Lutador ("DeadCole",
                    "Austrália" ,
                    23, 1.93 , 81.6, 13 , 0,2)

```

L[4]: novo Lutador ()

L[0] = novo Lutador (.....)

L[1] = novo Lutador (.....)

L[2] = novo Lutador (.....)

L[3] = novo Lutador (.....)

L[4] = novo Lutador (.....)

L[5] = novo Lutador (.....)

L[0].apresentar()

L[2].status()

L[3].getCategoria()

L[1].ganharLuta()

L[0].empatarLuta()

FimClasse