



Programação Orientada a Objetos 2022

Prof. Dr. José Erinaldo da Fonsêca

São Paulo - SP, Abril de 2022.

Relacionamento

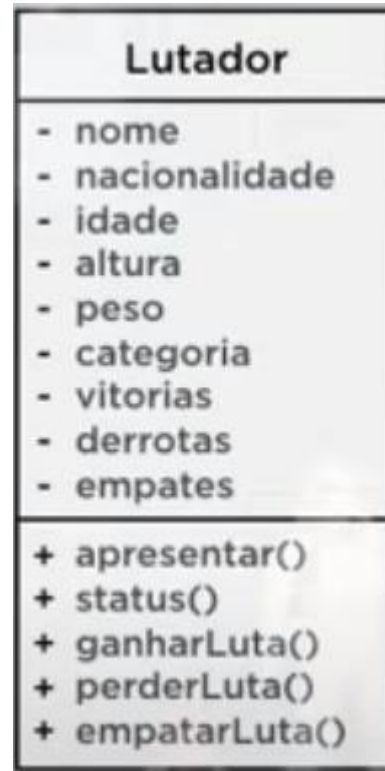
- ▶ Relacionamento entre Classes
- ▶ Agregação

- Suponha um combate entre lutadores.
- Para tal é necessário construir um perfil de cada lutador.



nome
nacionalidade
idade
altura
peso
categoria
vitorias
derrotas
empates

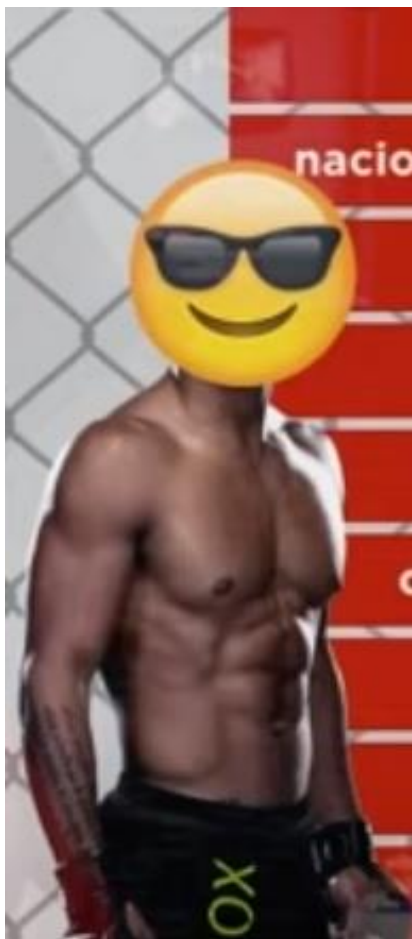
- ▶ Criamos um diagrama de classe.
- ▶ Inserimos os atributos de métodos.
- ▶ Atribuímos a visibilidade.
- ▶ E ainda podemos inserir os métodos gets e sets.



Os lutadores!!

Todos serão cadastrados.

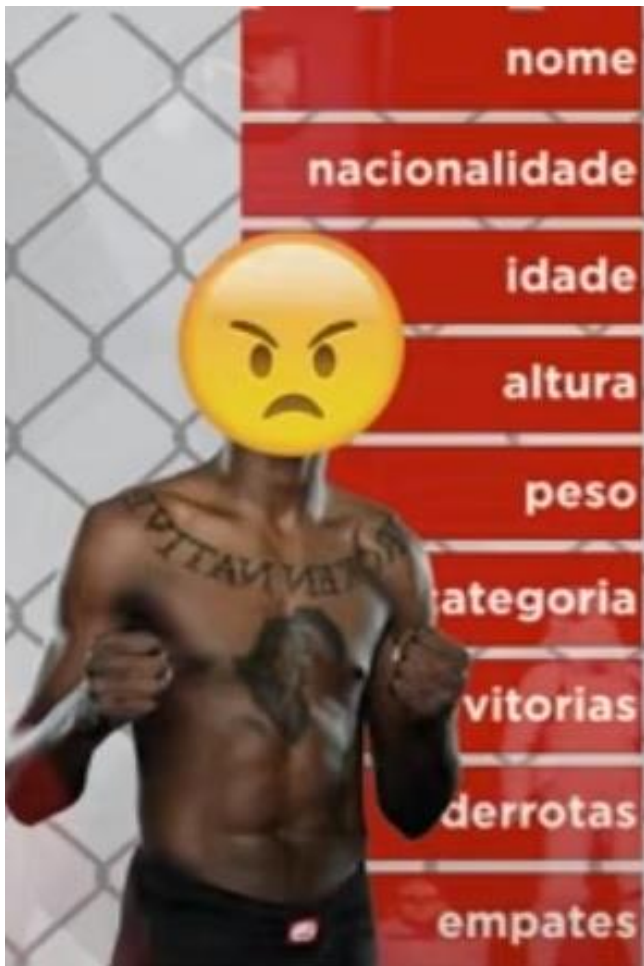
Lutador 1



A muscular man in boxing gear, including black gloves and black shorts with 'OX' in green. His face is replaced by a yellow emoji wearing sunglasses. He is standing in front of a chain-link fence.

nome	Pretty Boy
nacionalidade	França
idade	31
altura	1.75 m
peso	68.9 Kg
categoria	Leve
vitorias	11
derrotas	2
empates	1

Lutador 2

A fighter with a grumpy face emoji for a head, standing in a fighting stance. The emoji is yellow with a frowning mouth and angry eyebrows. The fighter has tattoos on his chest and arms.

nome	Putscript
nacionalidade	Brasil
idade	29
altura	1.68 m
peso	57.8 Kg
categoria	Leve
vitorias	14
derrotas	2
empates	3

Lutador 3



nome	Snapshadow
nacionalidade	EUA
idade	35
altura	1.65 m
peso	80.9 Kg
categoria	Médio
vitorias	12
derrotas	2
empates	1

Lutador 4



nome	Dead Code
nacionalidade	Austrália
idade	28
altura	1.93 m
peso	81.6 Kg
categoria	Médio
vitorias	13
derrotas	0
empates	2

Lutador 5



nome	Ufocobol
nacionalidade	Brasil
idade	37
altura	1.70 m
peso	119.3 Kg
categoria	Pesado
vitorias	5
derrotas	4
empates	3

Lutador 6



A muscular man with a yellow emoji face wearing glasses is standing in front of a table. The table has a red and white striped background and contains the following information:

nome	Nerdaard
nacionalidade	EUA
idade	30
altura	1.81 m
peso	105.7 Kg
categoria	Pesado
vitorias	12
derrotas	2
empates	4

- Como fica o código?
- Todos os atributos e métodos foram inseridos.
- Também é necessário colocar todos os métodos especiais: construtor, gets e sets.

```
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
// Métodos
publico metodo apresentar()
(...)
fim metodo
publico metodo Status()
(...)
fim metodo
publico metodo ganharLuta()
(...)
fim metodo
publico metodo perderLuta()
(...)
fim metodo
publico metodo empatarLuta()
(...)
fim metodo
FimClasse
```

Lutador

- nome
- nacionalidade
- idade
- altura
- peso
- categoria
- vitorias
- derrotas
- empates

- + apresentar()
- + status()
- + ganharLuta()
- + perderLuta()
- + empatarLuta()

Construtor!

```
// Métodos Especiais
publico metodo construtor(no:Caractere,
    nome = no                na:Caractere,
    nacionalidade = na       id:Inteiro,
    idade = id               al:Real,
    altura = al              pe:Real,
    setPeso(pe)              vi:Inteiro,
    vitorias = vi            de:Inteiro,
    derrotas = de            em:Inteiro)
    empates = em
fim metodo
```

Neste caso, a categoria vai ser inserida de forma automática pelo código

Gets e Sets

- Veja que nesse caso, a categoria é estabelecida dentro do setPeso.

```
// Métodos Especiais
publico metodo getNome()
    retorne nome
fim metodo
publico metodo setNome(no:Caractere)
    nome = no
fim metodo
publico metodo getPeso()
    retorne peso
fim metodo
publico metodo setPeso(pe:Caractere)
    peso = pe
    setCategoria()
fim metodo
```


- No setCategoria podemos atribuir a definição de um dos tipos de lutadores.

```
// Métodos Especiais
publico metodo setPeso(pe:Caractere)
    peso = pe
    setCategoria()
fim metodo
publico metodo setCategoria()
    se(peso<52.2)então
        categoria = "Inválido"
    senão se(peso<=70.3)então
        categoria = "Leve"
    senão se(peso<=83.9)então
        categoria = "Médio"
    senão se(peso<=120.2)então
        categoria = "Pesado"
    senão
        categoria = "Inválido"
    fim se
fim metodo
```

- Agora, podemos fazer os métodos

```
// Métodos
publico metodo 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())
fim metodo
publico metodo status()
    Escreva(getNome())
    Escreva("é um peso ", getCategory())
    Escreva(getVitorias(), "vitórias")
    Escreva(getDerrotas(), "derrotas")
    Escreva(getEmpates(), "empates")
fim metodo
```


► Continuando com os métodos

```
// Métodos
publico metodo ganharLuta()
    setVitorias(getVitorias() + 1)
fim metodo
publico metodo perderLuta()
    setDerrotas(getDerrotas() + 1)
fim metodo
publico metodo empatarLuta()
    setEmpates(getEmpates() + 1)
fim metodo

FimClasse
```

Como fica o instanciamento?

Lutador
- nome
- nacionalidade
- idade
- altura
- peso
- categoria
- vitórias
- derrotas
- empates
+ apresentar()
+ status()
+ ganharLuta()
+ perderLuta()
+ empatarLuta()

```
// Programa Principal
L: vetor[0..5]
L[0] = novo Lutador("Pretty Boy",
                    "França",
                    31, 1.75, 68.9,
                    11, 3, 1)

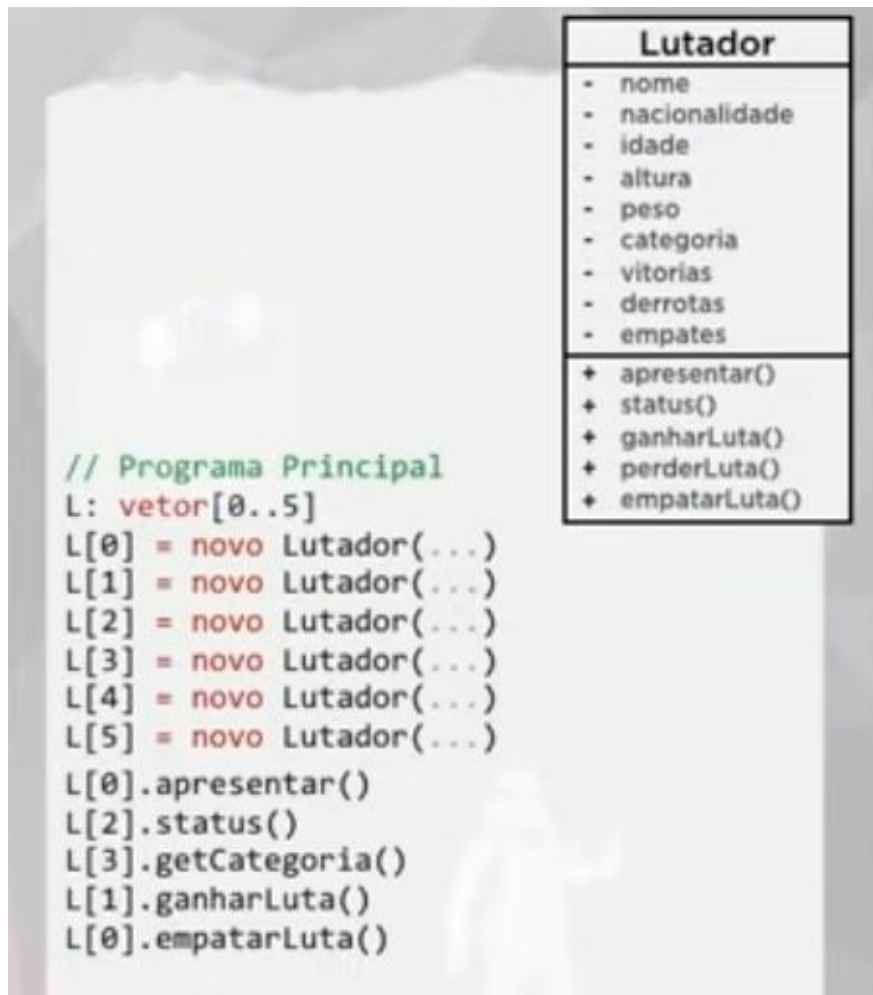
L[1] = novo Lutador("Putsript",
                    "Brasil",
                    29, 1.68, 57.8,
                    14, 2, 3)

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

L[3] = novo Lutador("Dead Code",
                    "Austrália",
                    28, 1.93, 81.6,
                    13, 0, 2)
```

Podemos criar um vetor!

Desenvolvido todo o instanciamento, você pode chamar cada objeto conforme desejado.



The image shows a code editor with a C++ program on the left and a class definition for 'Lutador' on the right. The program creates an array of six 'Lutador' objects and calls various methods on them. The class definition lists attributes and methods.

```
// Programa Principal
L: vetor[0..5]
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()
```

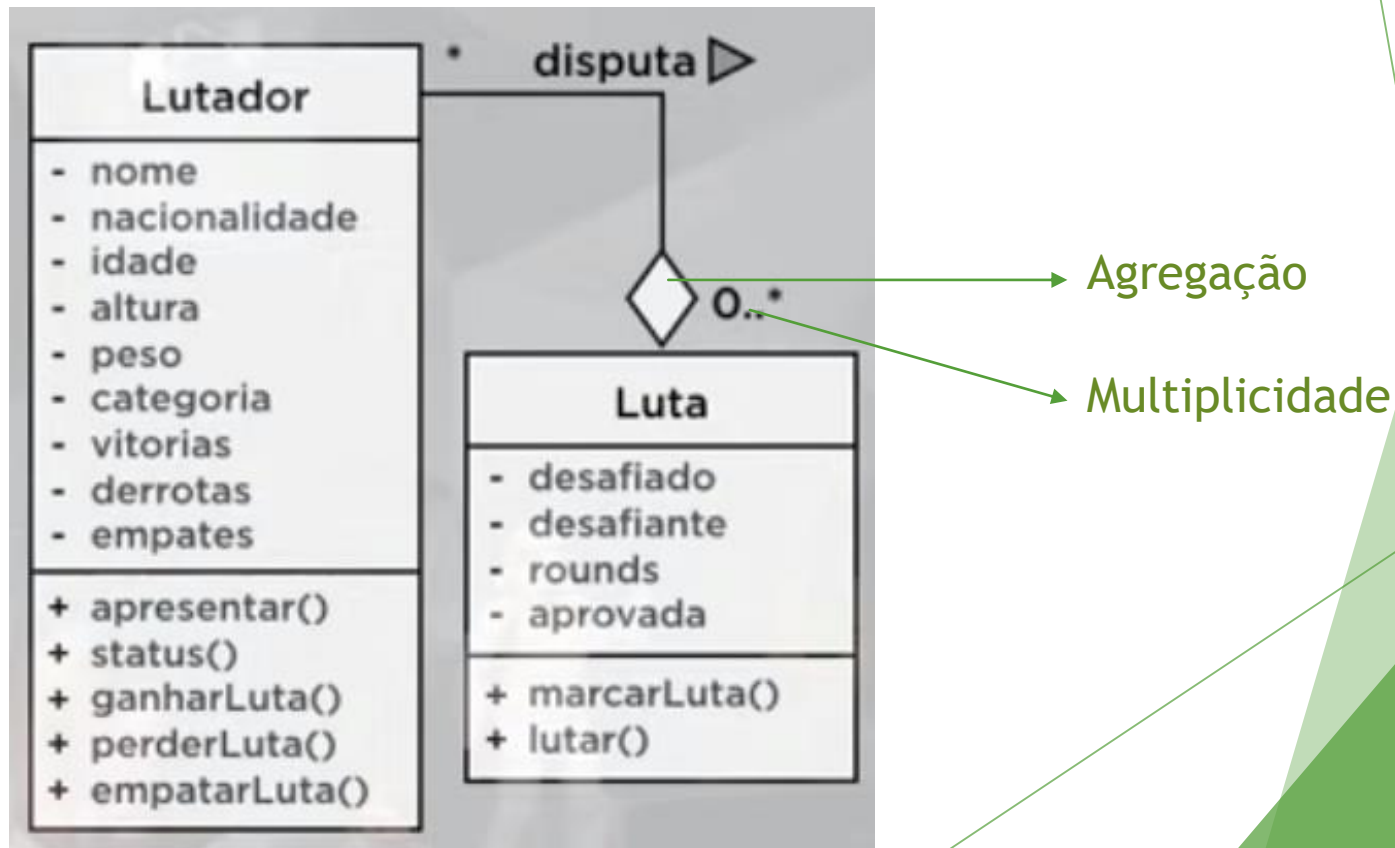
Lutador
- nome
- nacionalidade
- idade
- altura
- peso
- categoria
- vitórias
- derrotas
- empates
+ apresentar()
+ status()
+ ganharLuta()
+ perderLuta()
+ empatarLuta()

E o relacionamento de classes?

Todo lutador pode participar de uma luta.

Portanto, uma nova classe “Luta” pode ser criada.

Abaixo o diagrama de classe mostra essa relação entre classes.



Regras de uma luta!

- ▶ Só pode ser marcado entre lutadores de uma mesma categoria.
- ▶ Desafiado e desafiante devem ser lutadores diferentes.
- ▶ Só pode acontecer se estiver aprovada.
- ▶ Só pode ter como resultado a vitória de um dos lutadores ou empate.

Criando o método “marcarLuta”

Veja que você pode usar os gets de outra classe.

```
classe Luta

// Métodos
publico metodo marcarLuta(l1, l2: Lutador)
    se (l1.getCategoria() = l2.getCategoria())
        e (l1 <> l2) então
            aprovada = verdadeiro
            desafiado = l1
            desafiante = l2
        senão
            aprovada = falso
            desafiado = nulo
            desafiante = nulo
    fim se
fim metodo
```

Luta
- desafiado
- desafiante
- rounds
- aprovada
+ marcarLuta()
+ lutar()

Luta
<ul style="list-style-type: none"> - desafiado - desafiante - rounds - aprovada
<ul style="list-style-type: none"> + marcarLuta() + lutar()

classe Luta

// Métodos

publico metodo lutar()

se (aprovada)

desafiado.apresentar()

desafiante.apresentar()

vencedor = aleatório(0..2)

escolha (vencedor)

caso 0: //Empate

Escreva("Empatou!")

desafiado.empatarLuta()

desafiante.empatarLuta()

caso 1: //Ganhou Desafiado

Escreva(desafiado.getNome())

desafiado.ganharLuta()

desafiante.perderLuta()

caso 2: //Ganhou Desafiante

Escreva(desafiante.getNome())

desafiante.ganharLuta()

desafiado.perderLuta()

fim escolha

senão

Escreva("Luta não pode acontecer")

fim se

fim metodo

FimClasse

Criando o método "lutar"

Aqui é feito as definições de empate, vitória e derrota.


```
// Programa Principal
```

```
L: vetor[0..5]
```

```
L[0] = novo Lutador("Pretty Boy",  
                    "França",  
                    31, 1.75, 68.9,  
                    11, 3, 1)
```

```
L[1] = novo Lutador("Putsript",  
                    "Brasil",  
                    29, 1.68, 57.8,  
                    14, 2, 3)
```

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

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

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

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

```
UEC01 = novo Luta()
```

```
UEC01.marcarLuta(L[0], L[1])
```

```
UEC01.lutar()
```

Luta
- desafiado
- desafiante
- rounds
- aprovada
+ marcarLuta()
+ lutar()

Por fim, é incrementado o programa final.

É feito o instanciamento da luta.

Vamos Programar?

- ▶ Tente fazer os códigos do exemplo apresentado

The background features abstract green geometric shapes, including triangles and polygons, some of which are semi-transparent, creating a layered effect. The shapes are primarily located on the right side and bottom of the frame, with some extending towards the center.

▶ FIM!!