

# PROBLEMA



WAR - Jogo de Estratégia

Vários jogadores um tabuleiro.





# PROBLEMA



WAR - Jogo de Estratégia

Vários jogadores um tabuleiro.

Construir um modelo para este problema na UML  
e implementar em Java





# PROBLEMA



WAR - Jogo de Estratégia. Vários jogadores um tabuleiro.

**Como implementar este jogo, garantindo um único  
tabuleiro para todos os jogadores**



# PROBLEMA



Construir o modelo de uma solução usando a UML com um único tabuleiro para todos os jogadores e implementar em java.





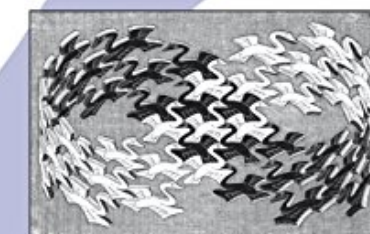
# PADRÕES DE PROJETO

## SINGLETON

**Prof. Me Eugênio Júlio Messala Cândido Carvalho**  
**eugeniojuliomessala@gmail.com**

**Padrões de  
Projeto**

Soluções reutilizáveis de software  
orientado a objetos



ERICH GAMMA  
RICHARD HELM  
RALPH JOHNSON  
JOHN VLISSIDES

**Design Patterns**



# SINGLETON

## ***INTENÇÃO***

Garantir que uma classe tenha somente uma instância e fornecer um ponto global de acesso para a mesma.

# SINGLETON

## ***MOTIVAÇÃO***

É importante para algumas classes ter uma, e apenas uma, instância.

Por exemplo, embora possam existir muitas impressoras em um sistema, deveria haver somente um spooler de impressoras.

# SINGLETON

## ***APLICABILIDADE***

Use o padrão Singleton quando:

- for preciso haver apenas uma instância de uma classe, e essa instância tiver que dar acesso aos clientes através de um ponto bem conhecido;
- a única instância tiver de ser extensível através de subclasses, possibilitando aos clientes usar uma instância estendida sem alterar o seu código.



# SINGLETON

## ***ESTRUTURA***

### **Singleton**

– unicalInstancia : Singleton

– Singleton()

+ getUnicalInstancia() : Singleton



# BIBLIOGRAFIA

- GAMMA, Erich et al. **Padrões de Projeto: soluções reutilizáveis de software orientado a objetos.** Porto Alegre: Bookman, 2007.
- KERIEVSKY, Joshua. **Refatoração para padrões.** Porto Alegre: Bookman, 2008.