Associação

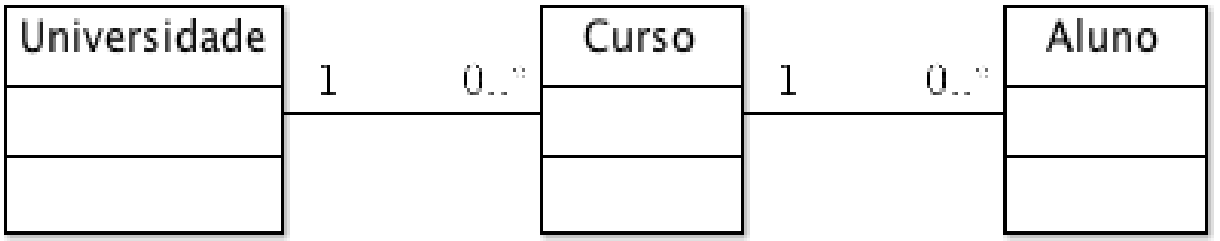
Carlos Bazilio

Depto de Ciência e Tecnologia
Instituto de Ciência e Tecnologia
Universidade Federal Fluminense

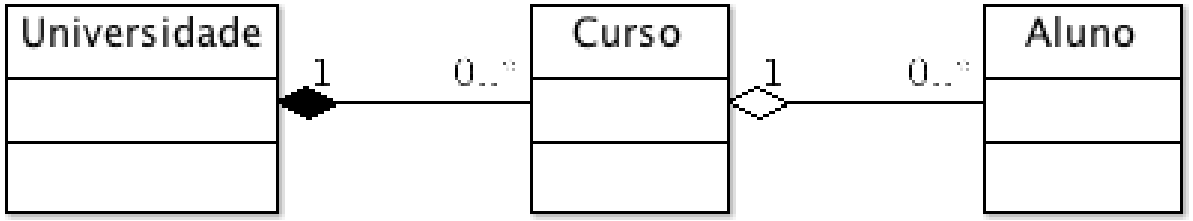
Tópicos Abordados

- Associação
- Agregação
- Composição

Associação



Composição e Agregação



Exemplo

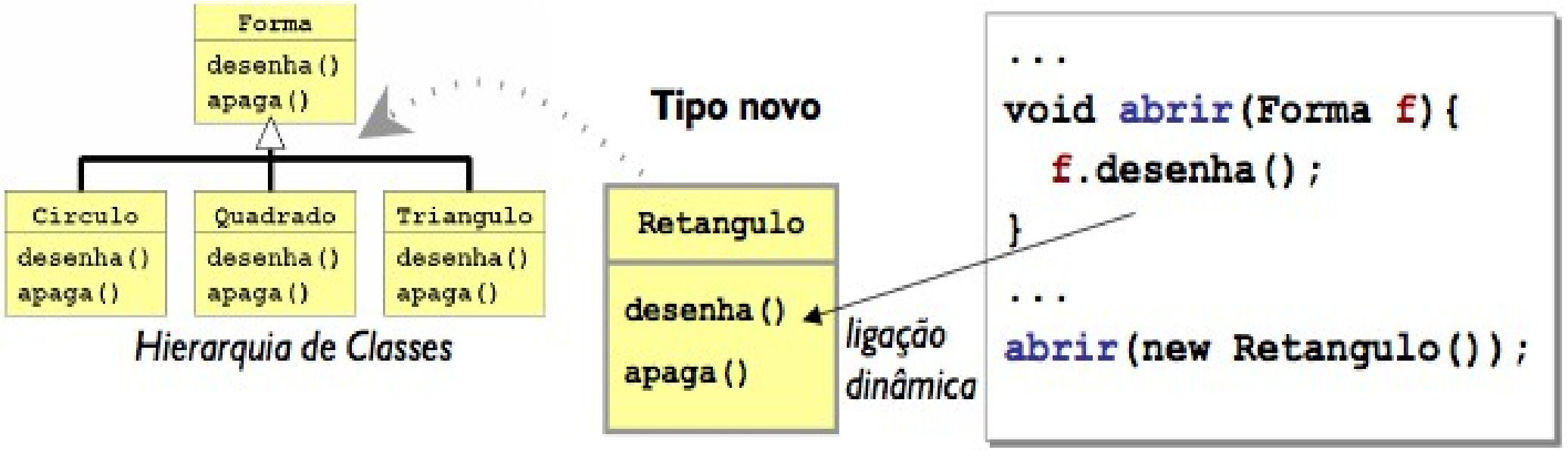
Composição:

```
Final class Car {
  private final Engine engine;
  Car(EngineSpecs specs) {
    engine = new
    Engine(specs);
  }
  void move() {
    engine.work();
  }
}
```

Agregação:

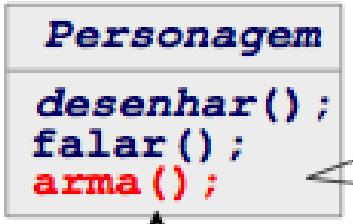
```
Final class Car {
  private Engine engine;
  void setEngine(Engine
    engine) {
    this.engine = engine;
  }
  void move() {
    if (engine != null)
      engine.work();
  }
}
```

Herança


Forma
desenha()
apaga()
Circulo
desenha()
apaga()
Quadrado
desenha()
apaga()
Triangulo
desenha()
apaga()
Hierarquia de Classes
Tipo novo
Retangulo
desenha()
apaga()
ligação
dinâmica
...
void abrir(Forma f){
f.desenha();
}
...
abrir(new Retangulo());

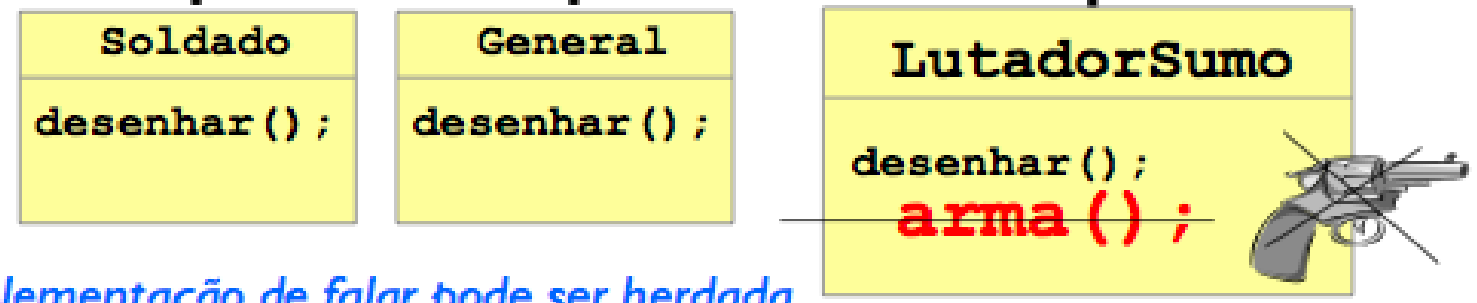
Herança – Problema 1

- *Um novo personagem:*




```
public class LutadorSumo
             extends Personagem {
 public void desenhar(){
    /* desenha o lutador */;
 }
 public void arma(){
   /* deixa corpo em branco? */
 }
}
```

Temos comportamentos que não são comuns a todos os objetos!

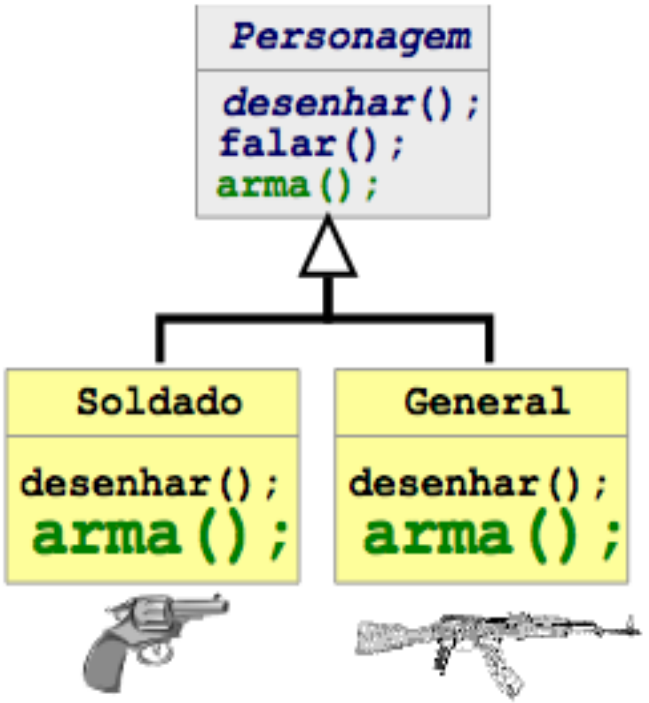


Implementação de falar pode ser herdada da superclasse Personagem.

Desenhar pode ser redefinido, mas...

☒ ***o lutador de SUMÔ não deve atirar!***

Herança – Problema 2



As subclasses podem redefinir métodos para um comportamento específico

```
public class Soldado
        extends Personagem{
 public void desenhar(){
   /* desenha o soldado */;
 }
 public void arma(){
   System.out.print("Tiro")
 }
}
```

```
public class General
        extends Personagem{
 public void desenhar(){
   /* desenha o general */;
 }
 public void arma(){
   System.out.print("Rajada")
 }
}
```

Herança – Problema 2

```
public class UsaPersonagem {
 public static void main(String[] args){
    Personagem p;

    p = new Soldado();
    p.desenha();
    p.arma(); // imprime "Tiro"
}
```

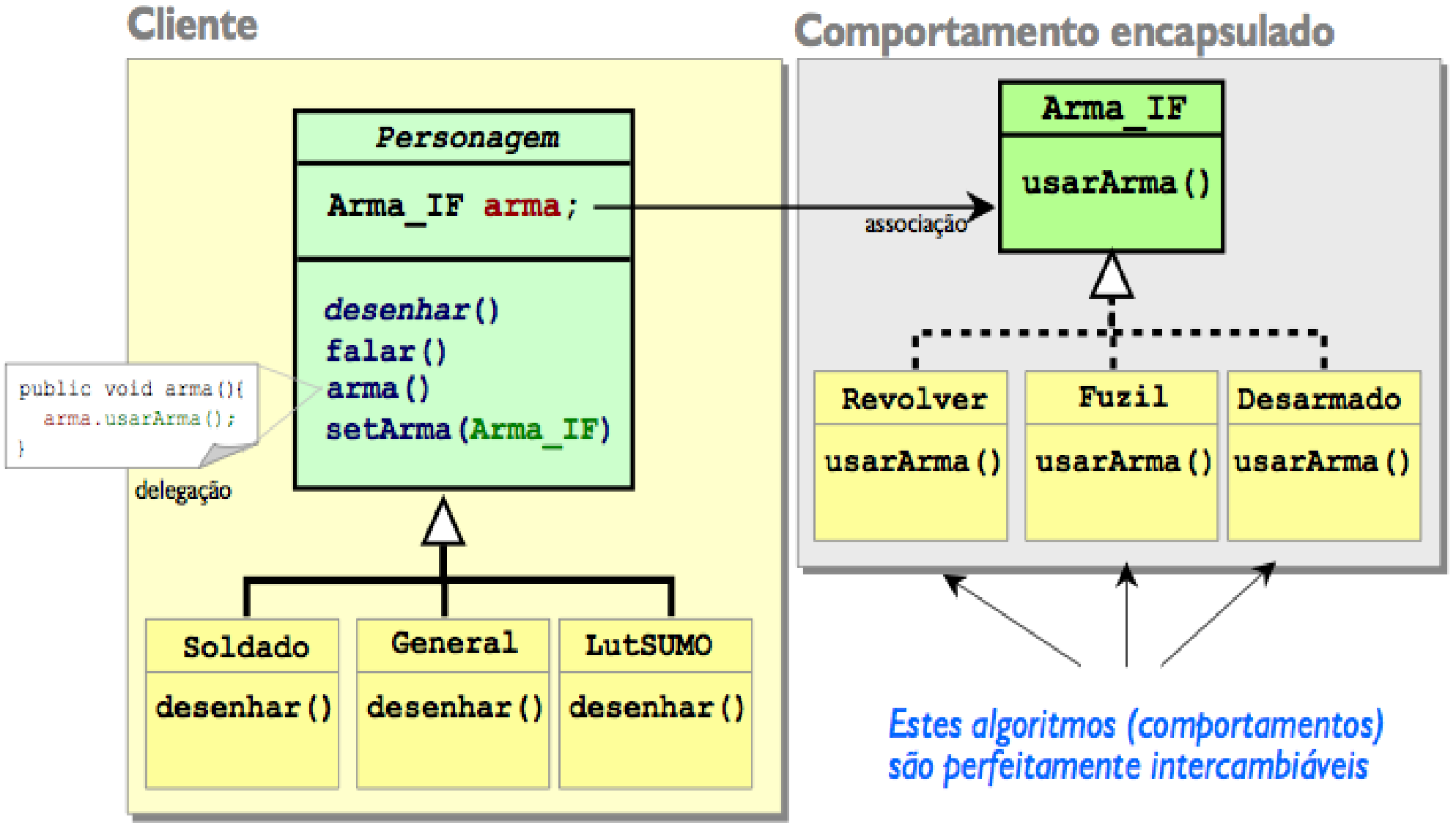
A arma (revólver) está parafusada no código da classe Soldado

```
public class Soldado
        extends Personagem{
 public void desenhar(){
    /* desenha o soldado */;
 }
 public void arma(){
    System.out.print("Tiro")
 }
}
```

```
public void arma(int arma){
    if(arma == 0){
      // imprime "Tiro"
    }else {
      // imprime "Rajada"
    }
}
```

O que acontece quando novas armas surgirem no jogo em cada uma das classes de personagens?

Solução: Composição



Solução: Composição

```
public class UsaPersonagem {
 public static void main(String[] args){
    Personagem p;

    p = new Soldado();
    p.desenhar();
    p.setArma( new Revolver() ); // define arma
    p.arma(); // imprime "Tiro"
    p.setArma( new Fuzil() ); // trocou arma
    p.arma(); // imprime Rajada

    p = new LutadorSumo();
    p.desenhar();
    p.setArma( new Desarmado() ); // define arma
    p.arma(); // imprime "Desarmado"

}
```

Referências

- https://www.thoughtworks.com/pt/insights/blog/composition-vs-inheritance-how-choose
- https://www.wikiwand.com/en/Composition_over_inheritance
- https://stackoverflow.com/questions/49002/prefer-composition-over-inheritance