

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PARAÍBA Campus Campina Grande</p>	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA – CAMPUS CAMPINA GRANDE		
	CURSO:	CURSO SUPERIOR DE TECNOLOGIA EM TELEMÁTICA	
	PERÍODO:	P3	TURMA: N
	DISCIPLINA:	PROGRAMAÇÃO ORIENTADA A OBJETOS	
	PROFESSOR:	CÉSAR ROCHA VASCONCELOS	SEMESTRE LETIVO

NOME:
-------

## PRÁTICA - HERANÇA VERSUS COMPOSIÇÃO

**Obs.:** procure responder a todas as questões desta lista não deixando nenhuma incompleta ou em branco.

- 1) (Eclipse) Crie um projeto no Eclipse de nome **ProjJogo**. Crie um pacote **br.com.jogo** para armazenar as classes. Implemente cada uma das classes e interfaces vistas nos slides (Personagem, Soldado, Arma\_IF, Revolver, General, etc.) com composição. Usando uma ferramenta UML real como o **BlueJ** (<http://www.bluej.org/>) ou **Jude** (<http://www.esm.jp/jude-web/index.html>) ou **ArgoUML** (<http://argouml.tigris.org/>) construa o diagrama parcial das classes (igual aos slides). Execute o programa fazendo testes (atribuindo e trocando armas em cada personagem, por exemplo). O programa deve funcionar normalmente.
- 2) Após terminar a questão anterior, implemente agora algumas mudanças no sistema do jogo:
  - a. Insira no jogo um novo personagem: **DragãoAlado** – verifique se todos os comportamentos definidos em Personagem são comuns ao dragão (um dragão deve falar?). Defina uma nova arma (a seu gosto) para o dragão (fogo, por exemplo)
  - b. Insira agora uma nova arma para os personagens: **Faca**.
  - c. Insira agora um novo método no jogo: **correr()**. Este método deve permitir que um personagem possa correr em caso de perigo. Verifique se todos os personagens podem correr (ou seja, este comportamento é comum a todos?). Ex.: um dragão sai na carreira?
  - d. Insira agora um novo método no jogo: **voar()**. Este método deve permitir que um personagem (em vez de correr) possa voar em caso de perigo. Verifique se todos os personagens são capazes de voar (ou seja, este comportamento é comum a todos?). Ex.: um lutador de sumô pode voar? E o dragão?
  - e. Insira agora uma nova arma para os personagens: uma **Bomba**.
  - f. Ao final, altere todo o diagrama de classes para acomodar as mudanças feitas acima.
- 3) (Eclipse) Crie um projeto no Eclipse de nome **ProjLocadora**. Crie um pacote **br.com.locadora** para armazenar as classes. Implemente um pequeno sistema para uma locadora (DVDs, CDs e Fitas). Identifique (a seu gosto) se há algum código comum que possa ser fatorado (atributos e métodos comuns a todos os produtos, por exemplo). Esta locadora trabalha com três categorias de preços de produtos: *lançamento(R\$10,00)*, *normal(R\$5,00)* e *infantil(R\$3,50)*. O sistema deve permitir que um produto que hoje é cobrado como lançamento, possa virar normal amanhã. Além disso, o sistema deve permitir que os preços possam ser alterados (por ex.: mudar o preço de lançamento de 10 para 12 reais a qualquer momento). O sistema também deve permitir que novos produtos possam ser inseridos futuramente (ex: cartucho de vídeo game). Usando uma ferramenta UML real como o **BlueJ** (<http://www.bluej.org/>) ou **Jude** (<http://www.esm.jp/jude-web/index.html>) ou **ArgoUML** (<http://argouml.tigris.org/>) construa o diagrama de classes. Execute o programa fazendo testes. O programa deve funcionar normalmente.