

Universidade Federal do Ceará – Campus de Quixadá
QXD0058 - Projeto Detalhado de Software
GRASP

Exercício de Sala ou Laboratório 02.5 – Exercício PBL sobre Especialista e Criador – GRASP

* Originalmente proposto em <http://www.lbd.dcc.ufmg.br/colecoes/wei/2007/0013.pdf>

Objetivos:

- Iniciar-se no uso de padrões para desenvolvimento de software;
- Compreender padrões para atribuição de responsabilidade (GRASP); Analisar qual padrão é mais adequado dada uma situação; Aplicar padrões GRASP.

Descrição do Problema:

Existe uma comunidade indígena que é composta de várias tribos. Cada tribo tem um nome, um cacique e fica localizada em uma determinada região. Um índio pertence a uma e somente uma tribo. Cacique também é considerado índio, e só passa a ser cacique quando assume a coordenação de uma tribo. Cada tribo pode ter diversos animais das mais variadas espécies. Frequentemente, acontece uma feira onde animais são trocados entre as tribos pelos seus respectivos caciques. Caso haja um cacique interessado em alguma troca, mas não há outro disposto a realizá-la, logicamente, não poderá ocorrer a troca. Um animal nunca é vendido, pois a comunidade não trabalha com dinheiro. Infelizmente, devido ao grande crescimento da comunidade nos últimos anos, a demanda de troca está muito alta, provocando assim, uma confusão na organização que administra estas trocas. Para isto, foi contratada uma empresa de software para desenvolver um sistema que facilitasse isto. O desenvolvedor responsável pela criação do sistema começou a fazê-lo, porém foi demitido pouco tempo depois por estar fazendo um trabalho muito ruim. O projeto estava sendo mal elaborado, ferindo padrões essenciais de orientação a objetos. O dono da empresa ficou sabendo que você é uma especialista em orientação a objetivos, principalmente, no que diz respeito a padrões, então ele resolveu contratá-lo para resolver tal problema.

Exercício:

- 1) Esboce uma solução em modelo de domínio para a descrição de sistema
- 2) Esboce uma solução em diagrama de classes para o modelo definido de forma a manter baixo hiato representacional
- 3) Aperfeiçoe o código fornecido como solução para o problema levando em conta seu conhecimento de padrões
 - a. O padrão de especialistas está sendo respeitado? Dicas: `Tribo.getNumPatas()` (herança indireta – fazer especialista e inversão de dependência) permite a fácil expansão para animais com número de patas diferentes? Em `Animal.estaNestaTribo(Tribo tribo)` (ver Especialista) é o animal que deve percorrer a lista de tribos para saber se faz parte dela?
 - b. O padrão criador está sendo respeitado? Dica: `Cacique.adicionaAnimalNaTribo(Tribo tribo, String nome, String especie)` (ver Criador) deveria ser o Cacique mesmo a adicionar um animal na tribo?
 - c. É possível melhorar a coesão?
 - d. É possível diminuir o acoplamento?

- e. Como a separar classes de UI de Lógica? Como separar camadas e classes de tratamento específico de comunicação ou armazenamento, por exemplo, usando invenção?