Lab-1 Semana 07

GRASP

**General Responsibility Assignment Software Patterns (or Principles)**

WebQuest

**Introdução**

GRASP consists of guidelines for assigning responsibility (method) to classes and objects in [object-oriented design](http://en.wikipedia.org/wiki/Object-oriented_design). The following 9 different patterns and principles are used in GRASP: 1. Controller, 2. Creator, 3. Indirection, 4. Information Expert, 5. High [Cohesion](http://en.wikipedia.org/wiki/Cohesion_(computer_science)), 6. Low [Coupling](http://en.wikipedia.org/wiki/Coupling_(computer_science)), 7. [Polymorphism](http://en.wikipedia.org/wiki/Polymorphism_(object-oriented_programming)), 8. Protected Variations, and 9. Pure Fabrication.

All these patterns answer some [software](http://en.wikipedia.org/wiki/Software) problem, and in almost every case these problems are common to almost every [software development](http://en.wikipedia.org/wiki/Software_development) project. These techniques have not been invented to create new ways of working, but to better document and standardize old, tried-and-tested [programming](http://en.wikipedia.org/wiki/Computer_programming) principles in object-oriented design.

Computer scientist [Craig Larman](http://en.wikipedia.org/wiki/Craig_Larman), who proposes GRASP (vide Recurso Secundário 1 abaixo), states that "the critical design tool for software development is a mind well educated in design principles. It is not the [UML](http://en.wikipedia.org/wiki/Unified_Modeling_Language) or any other technology."

**Tarefa**

Conhecer, ver exemplos e exercitar os conceitos de GRASP

**Processo**

1. Empregar no máximo 100min para realizar esta atividade.
2. 3 alunos vão ser atribuídos aleatoriamente a cada pattern.
3. [35m – Atividade sozinho]
   1. Cada aluno deve então ler a fonte do GRASP no Wikipedia (vide Recurso Primário 1 abaixo), bem como o link “See also” correspondente.
   2. Como resultado dessa leitura, apresentar um resumo de no máximo 2 páginas com suas palavras, explicando o pattern e qualquer termo relacionado relevante para o entendimento; relacionar com algum conceito visto nas aulas anteriores; apresentar também exemplo simples de sua aplicação em Java, fazendo uso de diagrama UML e código (antes e depois da aplicação do princípio, se for o caso e se for conveniente)!
   3. Para concretizar o documento anterior, vale procurar qualquer outro site, quando a fonte citada e os links associados não fornecerem material suficiente para o entendimento, esclarecimento e exemplificação do pattern.
   4. Ao final da atividade, esse resumo, identificado com o nome do aluno, deverá ser colocado, juntamente com os outros documentos gerados na atividade, no TIDIA.
4. [35m – Atividade em tripla]
   1. Os alunos responsáveis pelo mesmo pattern se reunem, explicam o que entenderam sobre o pattern, estudam os materiais dos outros e derivam apenas um material de até 2 páginas. Não vale escolher um dos três como representando o trio. Tem que fazer um merge com informações dos 3 alunos.
   2. Gerar até 5 slides representando o documento gerado no item anterior.
   3. Depois do tempo encerrado, não podem mais dedicar-se à produção dos slides! Pois devem se concentrar na atividade seguinte!
   4. Colocar o novo resumo e a apresentação ppt, ambas identificadas com os nomes do trio, no nome de um dos autores na atividade do TIDIA.
5. [30m – Apresentação]
   1. Três grupos de patterns vão ser sorteados para apresentação dos slides produzidos.
   2. Cada apresentação terá 5 minutos!
   3. Serão sorteados 3 alunos dos outros grupos para avaliar o que não ficou claro na apresentação de cada grupo; o grupo apresentador então tenta esclarecer as dúvidas levantadas. Esse processo todo deverá levar no máximo 5 min.
6. Os professores irão disponibilizar todos os resumos e apresentações ppts dos trios. Esses documentos + os resumos particulares dos alunos servirão para avaliação dos alunos em atividades de sala de aula!

**Recursos Primários**

1. GRASP:

<http://en.wikipedia.org/wiki/GRASP_(object-oriented_design)>

**Recursos Secundários**

1. Larman, Craig (2005) [2004]. Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd ed.). New Jersey: Prentice Hall. ISBN 0-13-148906-2.