Leituras pré-aula - Refatoração - CES28

Não se assustem com a quantidade, são textos pequenos!!!

(*) significa leitura obrigatoria
(z) "zapeie", olhe os assuntos, olhe rapidamente o que tem por ai, navegue por algo que te interesse
(c) curiosidade

O site basico ((*)leia a pagina de apresentação do site) eh
http://www.refactoring.com/
(*) que contem uma definicao de refactoring
http://martinfowler.com/bliki/DefinitionOfRefactoring.html
(z) e vem do Martin Fowler (de uma zapeada no site dele, interessante ver os assuntos)
http://martinfowler.com/
(z) Inclusive varios artigos sobre refactoring
http://martinfowler.com/tags/refactoring.html

(*) Essa eh uma boa definicao e contextualizacao de codesmell. Cheiraremos bastante código!
http://martinfowler.com/bliki/CodeSmell.html

Bem, para limpar o “smell” do código, precisa refatorar
Na aula iremos mostrar alguns dos exemplos do catalogo de refatorações
(*) e voces devem escolher alguns pra ler antes da aula
http://www.refactoring.com/catalog/
Podem ler outros, a maioria são simples, e há checkbox do lado para selecionar por assunto, mas tenho algumas sugestões:
Alguém perguntou como inicializar um valor depois do construtor e garantir que ele não mude depois
http://www.refactoring.com/catalog/removeSettingMethod.html
Este nós já vimos:
http://www.refactoring.com/catalog/encapsulateCollection.html
Este veremos de forma mais geral com outro nome, e aqui esta um exemplo simples super-resumido. Sabem o que são as setas e porque elas são importantes?
http://www.refactoring.com/catalog/hideDelegate.html
Quando falamos de acoplamento estático, que pode ser usado para trocar facilmente as implementações das classes concretas, alguém comentou que diferentes implementações poderiam ter diferentes interfaces. Aqui há idéias parecidas sobre como acessar classes de terceiros que não podem ser mudadas. Quando elas são melhores ou piores?
http://www.refactoring.com/catalog/?filter=tags-vendor-libraries,books-radio-appear
Existem refatorações no catalogo que são uma a inversa da outra. Porque?
http://www.refactoring.com/catalog/?filter=tags-associations,books-radio-appear

talvez alguns consigam encontrar uma solução melhor para a Q1 da prova (criar itens de venda, acessar clientes, etc). Tem varias soluções.
Uma boa pergunta: cada uma destas refatorações é simples. Porque existe o catálogo então?

(c) para quem gosta de saber de onde vem os nomes. “Forth” eh uma linguagem.
http://martinfowler.com/bliki/EtymologyOfRefactoring.html

(* mas não precisa ate terça, pode ser depois, mas antes da prova)
Bons slides sobre quando fazer refactoring durante o processo de desenvolvimento. Talvez comentemos na aula.
http://martinfowler.com/articles/workflowsOfRefactoring/

(* também não precisa ate terça, só antes da prova)
outra boa pergunta: Qual a diferença entre:

1. http://martinfowler.com/bliki/BranchByAbstraction.html
2. http://www.refactoring.com/catalog/extractInterface.html
3. Acoplamento Abstrato
   (sem contar que ainda veremos mais um conceito parecido em alguns padrões de projeto (e.g. Strategy) no segundo semestre)