

Centro Federal de Educação Tecnológica (CEFET-MG)
Departamento de Computação (DECOM) – Campus II

Prof. Eduardo Cunha Campos

Lista 02 de Exercícios

Aluno(a): _____
Matrícula: _____

Questão 01. Em relação à refatoração, assinale com V (verdadeiro) ou F (falso) as afirmações abaixo.

- () O melhor momento para se refatorar um código é durante os testes de aceitação, pois o cliente tem interesse em um código de qualidade.
- () Um dos passos da refatoração é a aplicação dos testes que verificarão sua implementação.
- () Rotinas muito longas e código duplicado são exemplos de *bad smells*.
- () Refatorações são modificações no código que são simples a ponto de não gerarem nenhum efeito prático.
- () Um código que já foi refatorado uma vez não precisará ser refatorado no futuro, pois já atende aos critérios de qualidade exigidos.
- () A refatoração de um código implica apenas a melhoria de sua qualidade interna e não deve afetar sua funcionalidade original.

A sequência correta de preenchimento dos parênteses, de cima para baixo, é:

- A) F–V–F–V–F–V.
- B) V–F–V–F–V–F.
- C) F–V–V–F–F–V.
- D) F–V–F–V–V–F.
- E) V–F–V–F–V–V.

Questão 02. Acerca do tema refatoração de *software*, considere as afirmativas.

- I. A refatoração busca evoluir o projeto e código-fonte de um sistema de *software* para se alcançar alta coesão, isto é, suas classes devem possuir conjuntos extensos de responsabilidades.
- II. A refatoração busca evoluir o projeto e código-fonte de um sistema de *software* para alcançar baixo acoplamento, isto é, a colaboração entre as classes deve ser mantida em um nível mínimo aceitável.
- III. A refatoração é o processo de mudar um sistema de *software* de tal forma que não altere o comportamento externo do código-fonte, embora melhore sua estrutura interna.

Está(ão) correta(s):

- A) I, II e III.
- B) I e II, somente.
- C) I e III, somente.
- D) III, somente.
- E) II e III, somente.

Questão 03. Quanto à refatoração (*refactoring*), analise as afirmativas abaixo e assinale a alternativa correta.

- I. Quando o prazo de entrega está próximo do final, deve-se fazer a refatoração.
- II. Quando o código está instável, e não funciona, é o melhor momento para refatorar.
- III. *EXtreme Programming* tem refatoração como uma de suas práticas.
- IV. Ela consiste em melhorar a estrutura do código de um sistema preservando as suas funcionalidades.

Estão corretas as afirmativas:

- A) Somente a I e II.
- B) Somente a II e III.
- C) Somente a III e IV.
- D) Somente a I e III.
- E) I, II, III e IV.

Questão 04. Discorra sobre 4 técnicas de Reuso de *Software* amplamente adotadas no cenário do desenvolvimento moderno de *software*.

Questão 05. Na programação de computadores, um cheiro de código (do inglês, “*code smell*”) é qualquer característica no código-fonte de um programa que possivelmente indica um problema mais profundo. *Shotgun Surgery* e *Divergent Change* são exemplos de *bad smells*. O que consiste cada um desses *smells* e qual é a diferença entre eles?

Questão 06. O modelo ágil se tornou uma tendência em empresas que precisam agregar mais valor para seus clientes, responder mais rapidamente às mudanças de mercado, diminuir falhas e aumentar a cadência de inovações em seus projetos de *software*. O *Scrum* é uma metodologia de desenvolvimento ágil baseado no conceito de equipes auto-organizadas. Quais são os papéis presentes em equipes *Scrum*? Cite e explique as responsabilidades de cada um dos papéis.

Questão 07. *Refactoring* é o processo que:

- A) implementa todas as funcionalidades da camada de model para depois implementar as camadas de controller e de viewer, nos casos em que a arquitetura MVC é utilizada.
- B) efetua mudanças em um código existente e funcional sem alterar seu comportamento externo, com o objetivo de aprimorar a estrutura interna do código.
- C) inclui funcionalidades extras no código, com o intuito de aprimorá-lo.
- D) aprimora a extração e o refinamento iterativo dos requisitos do produto ainda na fase de planejamento do *software*, sendo considerado um valor na metodologia XP.
- E) estabelece os métodos, um após o outro, para depois definir as classes e suas abstrações e implementar as interfaces.

Questão 08. A Reengenharia de *Software*, conhecida também como renovação ou reconstrução, é o exame e alteração de um sistema de *software*, para reconstituí-lo em uma nova forma, e a subsequente implementação dessa nova forma. Um processo de reengenharia geralmente inclui alguma forma de engenharia reversa, seguida por uma forma de engenharia progressiva ou reestruturação. Descreva 3 tarefas da Reengenharia.

Questão 09. Programação extrema (XP), é considerada uma metodologia ágil pois se ajusta bem a pequenas e médias em desenvolvimento de software com requisitos vagos e em constante mudança. Integração contínua é uma prática que foi introduzida pela XP. O que consiste esta etapa? Cite alguns exemplos de servidores de integração contínua adotados atualmente nos projetos de *software*.

Questão 10. Os *bad smells* presentes no código podem dificultar as atividades de manutenção do código-fonte. Essas estruturas precisam ser refatoradas para melhorar aspectos de legibilidade e compreensibilidade de código. Neste sentido, descreva a ocorrência de 4 tipos de *bad smells* que já foram catalogados na literatura. Para cada um deles, cite e explique qual é a ação de refatoração que deve ser feita para eliminá-lo.