



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AP2 1º semestre de 2013.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

- 1) O que significa uma transição em um diagrama de estados? O que representa uma condição quando associada a esta transição? (Valor: 1,0 ponto; máximo: 10 linhas)

Uma transição representa uma mudança de estado em um objeto, podendo ser automática ou gerada por um evento. Transições não podem ser interrompidas e custam tempo zero. Uma transição pode conter uma condição, também conhecida como condição de guarda, que é uma expressão booleana que indica quando a transição deve ocorrer. Nestes casos, a transição ocorre quando a condição for avaliada como verdadeira.

- 2) Dizemos que um projeto de software deve ser gerenciado em cinco grandes etapas: inicialização, planejamento, execução, controle e fechamento. Explique a relação existente entre as atividades de execução e controle no gerenciamento de projetos de software. (Valor: 2,0 pontos; máximo: 10 linhas)

Na etapa de execução o plano de projeto desenvolvido durante a etapa de planejamento é posto em prática. Porém, devido a características da equipe, do ambiente, dos patrocinadores envolvidos e de outros aspectos que estão fora do controle da equipe de gerenciamento de projeto, o andamento do projeto pode não ser exatamente igual ao esperado no plano. A etapa de controle ocorre em paralelo com a execução e nela a equipe de gerenciamento acompanha o andamento da execução, de modo a identificar o distanciamento entre o que foi planejado e o que está sendo executado. Assim, a etapa de controle reúne informações sobre o andamento

observado do projeto para que o gerente possa mudar o plano de modo a refletir a realidade ou mudar a realidade para tentar seguir o plano.

- 3) Marque Verdadeiro (V) ou Falso (F) para as afirmações abaixo (valor até 2,0 pontos – 0,2 por marcação correta. Atenção: cada 2 erros/2 brancos/ 1 erro, 1 branco eliminam 1 acerto!):

- A. FALSO.
- B. FALSO.
- C. FALSO.
- D. VERDADEIRO.
- E. FALSO.
- F. FALSO.
- G. VERDADEIRO.
- H. FALSO.
- I. FALSO.
- J. FALSO.

- 4) Defina o que representam os conceitos de acoplamento e coesão em software. Apresente os problemas que podem ocorrer quando cada uma destas características é observada no software. Por exemplo, considere o que pode ocorrer quando se apresentam em excesso ou deficiência. Sugira uma métrica que pode ser usada para observar cada conceito e exemplifique, através de um diagrama de classes, o conceito de acoplamento através de classes acopladas e não acopladas. (Valor: 2,0 pontos; máximo 10 linhas)

Acoplamento: Intuitivamente, refere-se ao grau de interdependência entre diferentes pedaços de um projeto

Comentário: Desde que objetos de uma mesma classe possuem as mesmas propriedades, duas classes estão acopladas quando métodos declarados numa classe utilizam métodos ou variáveis de instância de outra classe.

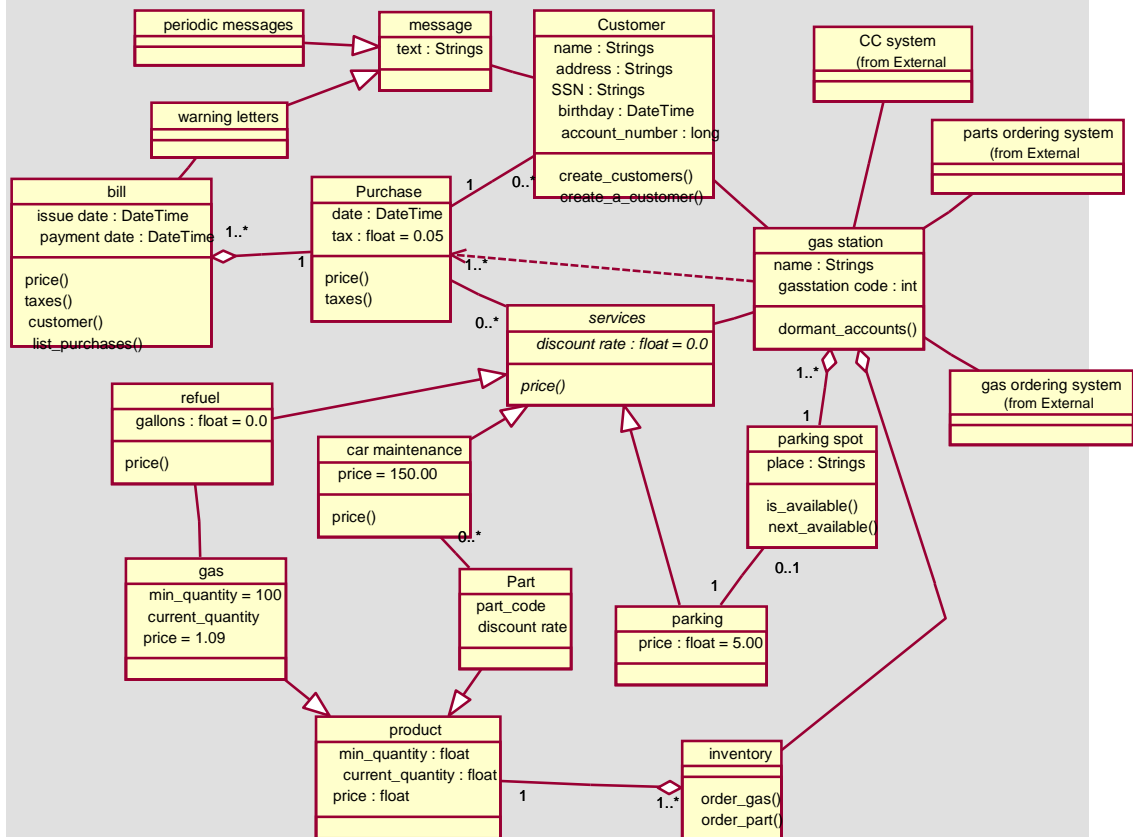
Coesão: Intuitivamente, refere-se à consistência interna dos pedaços de um projeto

Comentário: O grau de similaridade de métodos pode ser visto como o maior aspecto de coesividade das classes/objetos. Se uma classe/objeto tem diferentes métodos executando diferentes operações para um mesmo conjunto de variáveis de instância, a classe é coesa.

O excesso de acoplamento aumenta a dependência entre módulos. Assim, quanto mais acoplado, maior a chance de efeitos colaterais e mais suscetível a falha. A falta de coesão provoca uma situação semelhante, pois baixa coesão indica que o modulo (classe) não consegue resolver suas responsabilidades sem precisar usar recursos de terceiros (neste caso, gerando dependência). Assim, novamente as questões de efeito colateral vêm à tona, e a possibilidade de falha também.

Uma métrica que pode ser usada para medir acoplamento é a CBO (*coupling between objects*). Basicamente ela mede o numero de relacionamentos existente entre as classes que irão instanciar os objetos.

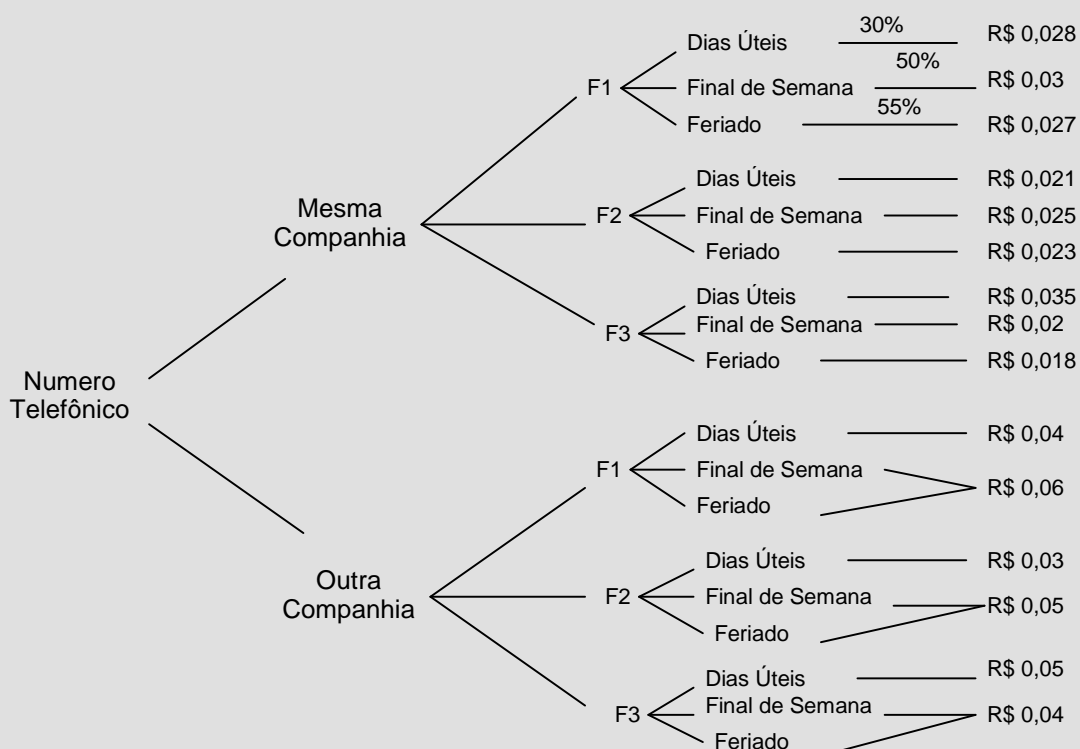
NO diagrama de classes abaixo (não era esperado um diagrama deste tamanho, apenas esta sendo usado para exemplificar o problema), a classe *Gas Station* possui acoplamento alto (8) em comparação com a classe *periodic messages* (1). Portanto, objetos da classe *Gas Station* são altamente acoplados, dependendo de 8 outros objetos para desempenhar suas responsabilidades.



- 5) Utilize a abordagem de grafo de causa e efeito (apresente a árvore e a tabela de decisão) para representar os casos de teste para o seguinte requisito (Valor: 2,0 pontos):

“ Uma companhia telefônica realiza a tarifação das ligações comerciais considerando as categorias de horário F1, F2 e F3. Nos dias úteis as ligações para as categorias F1, F2 e F3 são tarifadas em R\$ 0,04, R\$0,03 e R\$ 0,05, respectivamente. Para os finais de semana e feriados, as tarifas das ligações para F1, F2 e F3 são R\$ 0,06, R\$0,05 e R\$ 0,04 respectivamente. Caso a ligação seja realizada para numero telefônico da mesma companhia, as tarifas tem redução de 30% nos dias úteis, de 50% nos finais de semana e de 55% nos feriados.”

O arranjo do grafo e da tabela podem variar, dependendo da variável escolhida para a representação inicial. Aqui vai uma solução considerando o numero telefônico como partida.



Numero Telefônico	Mesma Companhia									Outra Companhia								
Faixa Tarifaria	F1			F2			F3			F1			F2			F3		
Dia	DU	FS	FE	DU	FS	FE	DU	FS	FE	DU	FS	FE	DU	FS	FE	DU	FS	FE
Tarifa	0,028	0,03	0,027	0,021	0,025	0,023	0,035	0,02	0,018	0,04	0,06	0,06	0,03	0,05	0,05	0,05	0,04	0,04

DU= Dia Útil, FS=Final de Semana, FE=Feriado

- 6) Defina defeito e falha e indique quais técnicas podem ser usadas para identificar cada um deles. (Valor: 1,0 ponto, 10 linhas).

A falha representa um comportamento inesperado do software, ou seja, observa-se um comportamento diferente do que foi especificado. Uma falha pode ser consequência de uma ou mais faltas presentes no software. Usualmente, refere-se a estas faltas como defeitos. Falhas podem ser reveladas pelos testes de software. Da falha se pode chegar aos defeitos. Por sua vez, defeitos podem ser detectados por inspeção do software.