



Lista de Questões Retiradas de Provas Parciais 1

1. Modelagem Orientada a Objetos

Prova Parcial 1 de Programação III 2013/1

(6 pontos – classes: 2pts, atributos: 1pt, associações: 2pts, herança: 1pt)

Considere a descrição abaixo para um problema do mundo real para o qual se deseja construir uma solução computacional:

Uma empresa de prestação de serviços em informática possui um quadro de funcionários que podem ser alocados a projetos. Esses projetos são projetos de desenvolvimento de software de clientes que estão terceirizando suas atividades de desenvolvimento. Os projetos podem ser estipulados por serviços (por exemplo, a fase de programação do processo de desenvolvimento de um sistema) ou por preço unitário (por exemplo, contratação de um programador ao preço de R\$ 15,00 a hora e de um analista júnior a R\$ 20,00 a hora). Cada projeto tem um e somente um cliente, sendo que os clientes da empresa são sempre pessoas jurídicas, das quais se deseja saber razão social, CNPJ, endereço, telefone e pessoa de contato.

Funcionários ocupam cargos específicos (analista sênior, analista júnior, programador, etc.) e recebem salários condizentes com os mesmos. Assim, de um cargo deseja-se saber nome e salário. De cada funcionário, por sua vez, deseja-se saber: nome, endereço, telefone, RG e CPF. De um projeto deseja-se saber o cliente, o nome do projeto, a data de contratação do projeto e a data de encerramento do projeto. Quando um funcionário é alocado a um projeto, tem de se registrar a data de início da alocação. O mesmo deve acontecer quando um funcionário é desligado do projeto.

Os projetos por serviços estabelecem os serviços a serem executados, a data de início (para cada serviço) e o valor a ser pago na conclusão de cada um destes. Quando um serviço for concluído no âmbito de um projeto por serviço, deve-se registrar, ainda, a data de conclusão, quando será solicitado o pagamento do serviço junto ao cliente. Contudo, o controle de pagamentos está fora do escopo deste sistema. Por fim, os possíveis serviços oferecidos pela empresa são previamente estabelecidos e de um serviço deseja-se saber nome, descrição e tipo de profissional (cargo) necessário para desempenhá-lo.

Os contratos por preço unitário especificam os tipos dos profissionais (cargos) a serem cedidos, o valor da hora de trabalho a ser cobrado do cliente para cada um dos tipos de profissionais contratados e a carga horária mensal de alocação de cada um deles. Ainda que não seja responsabilidade deste sistema controlar pagamentos, o sistema deve ser capaz de informar, dado um mês, quanto a empresa tem a receber de um cliente pelos funcionários cedidos.

Desenhe um diagrama de classes da UML que represente o domínio do problema acima, considerando o nível de abstração de análise de requisitos. Tome por base apenas a descrição acima. Não é necessário modelar operações das classes.

Prova Parcial 1 de Programação III 2013/2

(6 pontos – classes: 2pts, atributos: 1pt, associações: 2pts, herança: 1pt)

Considere a descrição abaixo para um problema do mundo real para o qual se deseja construir uma solução computacional:

A empresa distribuidora de material de escritório DME deseja informatizar suas atividades. A empresa recebe produtos de fornecedores e vende no atacado para seus clientes. Tanto fornecedores quanto clientes são pessoas jurídicas, das quais deseja-se registrar razão social, CNPJ, telefones e pessoa de contato (este último opcional). Entretanto, dos clientes deseja-se saber, ainda, o endereço para entrega de mercadorias e dos fornecedores os produtos que ele pode fornecer.

Sobre produtos, deseja-se saber o código de barras, a descrição e o valor atual para venda (deseja-se também manter um histórico dos preços antigos). Produtos são fornecidos em lotes, sendo que de um lote deseja-se saber o código, a data de validade, a quantidade de produtos e o fornecedor. Um lote contém produtos de um único tipo, sendo que um produto pode ser adquirido em diversos lotes diferentes.

Clientes efetuam pedidos. Para cada pedido, deseja-se saber o número, a data, a forma de pagamento (opcional) e a quantidade de cada um dos produtos envolvidos.



Desenhe um diagrama de classes da UML que represente o domínio do problema acima, considerando o nível de abstração de análise de requisitos. Tome por base apenas a descrição acima. Não é necessário modelar operações das classes.

Prova Parcial 1 de PAC / Programação III 2014/1

(6 pontos – classes: 2pts, atributos: 1pt, associações: 2pts, herança: 1pt)

Considere a descrição abaixo para um problema do mundo real para o qual se deseja construir uma solução computacional:

A padaria do senhor Oak está tecnologicamente atrasada. O sistema de controle de conta é à moda antiga, papel e caneta. A padaria possui uma grande clientela local, pois o Sr. Oak permite que os clientes anotem “na conta” os produtos que estão levando e efetuem o pagamento no final do mês (pagamento fiado). Ele consegue fazer isso pois também negocia com os fornecedores prazo para pagar suas próprias compras no início do mês seguinte. Acontece que a cada fim do mês é sempre o mesmo problema: o senhor Oak tem dificuldades para calcular quanto tem a receber e quanto tem a pagar somente olhando as anotações em papel. Cansado desta situação, ele deseja um sistema para obter um melhor gerenciamento.

O sistema deve manter um cadastro de fornecedores e dos clientes que pagam fiado (os que não pagam fiado não precisam de cadastro). Dos clientes deseja-se registrar código identificador, nome, endereço, telefone e desde quando o cliente compra na padaria (data do seu cadastro). Clientes podem ser de dois tipos: pessoas físicas ou pessoas jurídicas (empresas). De pessoas físicas deseja-se registrar o CPF, enquanto de pessoas jurídicas deseja-se registrar CNPJ e número de inscrição estadual. Dos fornecedores deseja-se registrar código identificador, nome, endereço, telefone, CNPJ e pessoa de contato. Fornecedores são sempre pessoas jurídicas.

Os produtos são cadastrados no sistema com as seguintes informações: código, descrição, estoque mínimo, quantidade atual em estoque, valor de custo e percentual de lucro. O valor de venda de um produto é calculado com base no valor de custo e percentual de lucro.

Ao longo do mês, os funcionários da Padaria Oak registram cada compra de produtos dos fornecedores e cada venda de produtos para clientes. Das compras efetuadas pela padaria junto aos fornecedores são registrados o número da nota fiscal, o fornecedor, a data da compra, o tipo de produto comprado e a quantidade. Para simplificar, assuma que para cada tipo de produto comprado é gerada uma nota fiscal separada. O valor pago pode ser calculado multiplicando o valor de custo do produto e a quantidade comprada.

Das vendas efetuadas pela padaria aos clientes são registrados o cliente, a data da venda, o produto vendido, a quantidade vendida e o meio de pagamento. Novamente, o valor de cada venda pode ser calculado a partir o valor de venda do produto e a quantidade vendida. Os meios de pagamento aceito são: dinheiro, cheque, cartão de débito/crédito, ticket alimentação e fiado.

Desenhe um diagrama de classes da UML que represente o domínio do problema acima, considerando que ele será implementado em Java. Tome por base apenas a descrição acima. Não é necessário modelar operações das classes. Para datas, utilize a classe `java.util.Date`. Observe as regras de encapsulamento ao definir os atributos.

Prova Parcial 1 de PAC 2014/2

(5 pontos – classes: 1pt, atributos: 1pt, associações: 2pts, herança: 1pt)

Considere a descrição abaixo para um problema do mundo real para o qual se deseja construir uma solução computacional:

Uma revista de informática, a EngeSoft, deseja um novo sistema para gerenciar suas atividades. EngeSoft é publicada mensalmente, sendo que uma edição tem diversos artigos, todos versando sobre um mesmo tema. Por exemplo, a edição deste mês é sobre o tema “Qualidade de Software”, tendo oito artigos. De uma edição deseja-se saber o volume, número, data prevista de publicação, tema e artigos submetidos.

Autores submetem artigos para uma edição específica. De um artigo deseja-se saber os autores e o título. Os autores devem informar, além de seus nomes, e-mails e as instituições a que pertencem, com endereço. Para artigos com mais de um autor, deve ser indicado um autor como contato. Para avaliar os artigos submetidos à publicação, a EngeSoft possui um conjunto de colaboradores que avaliam artigos (chamados revisores). Dos revisores deseja-se saber o nome, e-mail, instituição e temas para os quais está habilitado a avaliar artigos.

Essas informações são usadas para distribuir os artigos para os colaboradores. Cada artigo é obrigatoriamente avaliado por três revisores, todos habilitados ao tema da edição correspondente, que atribuem notas de 0 a 10 (com até uma casa decimal) a três itens: originalidade, conteúdo e apresentação. Com base nessas avaliações é que se



decide se um artigo será publicado ou não. Para simplificar, considere que um revisor é obrigado a avaliar um artigo quando este lhe é atribuído (ou seja, não pode haver cancelamento de atribuição).

Artigos que já foram avaliados pelos seus três revisores estão prontos para a seleção de quais artigos serão publicados na edição em questão, caso contrário encontram-se ainda em avaliação. Apenas quando todos os artigos submetidos para uma edição tiverem sido avaliados é que a seleção pode ser efetuada. Esta seleção é feita pelo editor-chefe da edição, escolhido previamente no conjunto de colaboradores. Finda a seleção, sabem-se quais artigos foram selecionados para publicação e quais foram rejeitados.

A diretoria da revista EngeSoft gostaria de um sistema para auxiliar a tarefa do editor-chefe de cada edição. O sistema deverá, dadas as avaliações dos artigos, calcular as médias das avaliações e apresentar ao editor-chefe um relatório dos artigos avaliados, por ordem de média das avaliações (dados da edição e o nome do editor-chefe deve constar no cabeçalho do relatório).

Desenhe um diagrama de classes da UML que represente o domínio do problema acima, considerando que ele será implementado em Java. Tome por base apenas a descrição acima. Não é necessário modelar operações das classes. Para datas, utilize a classe `java.util.Date`. Observe as regras de encapsulamento ao definir os atributos.

Prova Parcial 1 de PAC 2015/2

(5 pontos – classes: 1pt, atributos: 1pt, associações: 2pts, herança: 1pt)

Considere a descrição abaixo para um problema do mundo real para o qual se deseja construir uma solução computacional:

Com a vastidão de formas de entretenimento que temos hoje, é complicado manter em mente o que já se viu e o que ainda se quer ver, bem como lembrar quem está com sua cópia do livro “O Senhor dos Anéis”, seu Blu-ray do último filme do Tarantino ou seu DVD da última temporada de “Game of Thrones”.

Propomos um sistema capaz de inventariar as “mídias” de entretenimento – considerando livros, filmes ou séries de TV¹ – com as quais o usuário tem algum tipo de relação, dentre as quais: (a) possui uma cópia; (b) já “consumiu” a mídia (ou seja, leu o livro ou assistiu o filme / a temporada da série), independente de possuir ou não uma cópia; (c) deseja consumir a mídia (também independente de ter ou não uma cópia).

Além da relação do usuário com a mídia e o nome da mesma, deseja-se registrar as fichas técnicas de cada uma, a saber: (a) para livros: nomes dos autores, número de páginas e gênero; (b) para filmes: diretor (considere que há somente um), atores principais (cujo registro é opcional), gênero e duração; (c) para temporadas de séries: atores principais (também opcional), gênero, duração, número da temporada e nome da série. Para filmes e séries, deseja-se também registrar o formato da mídia: DVD, Blu-ray ou cópia digital (arquivo no computador).

O gênero é a categoria usada pelo mercado para classificação de filmes, séries, livros e outras mídias, por exemplo: ação, comédia, ficção científica, etc. Quando mais de um gênero é aplicável (ex.: ação e comédia), o sistema se limitará a registrar apenas um.

Ao marcar uma mídia como emprestada, o sistema deve permitir registrar a data de empréstimo, uma data de devolução e para quem a mídia foi emprestada. O sistema, então, ajudaria a controlar não só o que está emprestado, mas, dentre estes, o que já deveria ter sido devolvido e a quem devemos cobrar a devolução. Por fim, como algumas mídias são desejadas e ainda não se possui uma cópia, deseja-se registrar o preço das mesmas.

Com tais informações, espera-se que o sistema seja capaz de gerar para o usuário alguns relatórios úteis como, por exemplo: listagem das mídias emprestadas, indicando com quem está, se já deveriam ter sido devolvidas e, neste caso, quantos dias de atraso acumulam; listagem das mídias por autor/diretor/ator; wishlist, ou seja, lista de mídias que se deseja consumir, mas que ainda não se possui, e seus respectivos preços, dando ideias de possíveis presentes que o usuário deseja ganhar no futuro; estatísticas gerais, como horas gastas com mídias já consumidas, horas necessárias para consumir os demais itens, total de mídias por tipo e por gênero, quantidade de temporadas de cada série que se possui e que já se consumiu.

¹ Para séries de TV, considera-se uma temporada inteira da série, pois geralmente as mídias (DVDs, Blu-Rays) são vendidas desta maneira. O sistema não deve registrar se o usuário assistiu um ou outro episódio de uma série, mas sim temporadas inteiras.



Universidade Federal do Espírito Santo

Centro Tecnológico

Departamento de Informática

Programação III / PAC
(INF 09331 / INF 09324)

Prof. Vítor E. Silva Souza

Desenhe um diagrama de classes da UML que represente o domínio do problema acima, considerando que ele será implementado em Java. Tome por base apenas a descrição acima. Não é necessário modelar operações das classes. Para datas, utilize a classe `java.util.Date`. Observe as regras de encapsulamento ao definir os atributos.

2. Java, sem envolver o conceito de herança

Prova Parcial 1 de PAC 2014/1 (3 pontos)

Considere o programa Java ao lado (considere que ele encontra-se no método `main()` de uma classe pública). A caixa de texto logo abaixo mostra uma possível execução deste programa, sendo que os trechos em *itálico* representam dados digitados pelo usuário.

É possível ver que o programa recebe do usuário o nome de uma disciplina e o número de alunos e, em seguida, lê também o nome e a média de cada aluno e associa-os à disciplina. Por fim, o programa solicita ao objeto da classe `Disciplina` que imprima o resultado da mesma. Para o resultado, a nota mínima para aprovação é 5,0.

O código-fonte, no entanto, encontra-se incompleto, referenciando classes `Aluno` e `Disciplina` que não são apresentadas. Escreva as classes faltantes, além de uma classe `Matricula` para associar um aluno a uma média (pois não faz sentido armazenar a nota nem em `Aluno` nem em `Disciplina`, de modo que o programa acima produza a execução ilustrada ao lado.

Observação: o método `matricular()` em `disciplina` retorna um número inteiro que indica a posição do aluno matriculado na pauta daquela disciplina. Em seguida, o método `atribuirMediaFinal()` informa a posição da pauta onde a média, também passada como parâmetro, deve ser registrada. Ou seja, ao atribuir uma nota a um aluno em uma disciplina, se usa a posição dele na pauta, e não o nome dele ou qualquer outra informação.

```
Scanner scanner = new Scanner(System.in);
String nome = scanner.nextLine();
int numAlunos = scanner.nextInt();
scanner.nextLine(); // Despreza o \n.
Disciplina disc = new Disciplina(nome,
numAlunos);

for (int i = 0; i < numAlunos; i++) {
    nome = scanner.nextLine();
    double media = scanner.nextDouble();
    scanner.nextLine(); // Despreza o \n.

    Aluno aluno = new Aluno(nome);
    int pos = disc.matricular(aluno);
    disc.atribuirMediaFinal(pos, media);
}

disc.imprimirResultado();
scanner.close();
```

```
PAC
3
Huguinho
9.5
Zezinho
4.2
Luisinho
6.0
PAC:
Huguinho: nota 9.5
Zezinho: nota 4.2
Luisinho: nota 6.0
2 aprovado(s) e 1 reprovado(s)
```

Prova Parcial 1 de PAC 2014/2 (3 pontos)

Analise o programa Java ao lado (considere que o código encontra-se no método `main()` de uma classe pública). As caixas de texto na margem direita da página mostram uma possível execução deste programa: acima os dados digitados pelo usuário, abaixo a saída produzida pelo programa.

Implemente as três classes que faltam para o programa funcionar como demonstrado:

```
Scanner scanner = new Scanner(System.in);
int num = scanner.nextInt();
scanner.nextLine(); // Despreza o \n.

CarrinhoDeCompras c = new
CarrinhoDeCompras(num);
for (int i = 0; i < num; i++) {
    String nome = scanner.nextLine();
    double preco = scanner.nextDouble();
    scanner.nextLine(); // Despreza o \n.
    int qtd = scanner.nextInt();
    scanner.nextLine(); // Despreza o \n.
    c.adicionar(nome, preco, qtd);
}
```

```
3
Papel A4
11.4
3
Fita adesiva
4.99
2
Caneta
2.09
5
```

```
No carrinho:
* 3x Papel A4
* 2x Fita adesiva
* 5x Caneta
Total: 54.630000
```



Universidade Federal do Espírito Santo

Centro Tecnológico

Departamento de Informática

Programação III / PAC
(INF 09331 / INF 09324)

Prof. Vítor E. Silva Souza

Produto representa um produto vendido na loja; ProdutoNoCarrinho representa um produto adicionado ao carrinho em uma certa quantidade; e CarrinhoDeCompras representa o carrinho de compras do cliente. Seu código deve ser o mais OO possível.



3. Java, envolvendo o conceito de herança

Prova Parcial 1 de Programação III 2014/1 (2 pontos)

Escreva em Java uma classe que representa um carro. Tal classe deve conter como atributos obrigatórios (valores definidos na construção das instâncias) o nome do carro e a quantidade de litros que tem em seu tanque de combustível. Instâncias dessa classe devem poder receber chamadas a um método andar(double d) que recebe uma distância d em km e imprime na tela: “<nome> andou <d> km em <h> horas e agora possui <l> litros de combustível”, substituindo os elementos entre < e > pelos dados daquele carro. Considere que carros desta classe andam sempre a 100 km/h e fazem 10 km por litro;

Escreva uma outra classe que represente carros turbinados. A diferença do carro turbinado para o carro descrito acima é que ele anda sempre a 150 km/h e faz 6 km/l.

Escreva um programa que peça ao usuário que informe um nome para um carro normal, um nome para um carro turbinado, uma quantidade de combustível e uma distância em km. Seu programa deve, então, criar um carro normal e outro turbinado com seus respectivos nomes, com a mesma quantidade de combustível e chamar o método andar() nos dois carros, passando a mesma distância em km.

Não é necessário se preocupar com uma interface amigável com o usuário. Em outras palavras, assuma que o usuário sabe como funciona o programa e não precisa que sejam impressas instruções na tela.

Exemplo:

```
Go1
Ferrari
40
150
Go1 andou 150.0 km em 1.5 horas e agora possui 25.0 litros de combustível
Ferrari andou 150.0 km em 1.0 horas e agora possui 15.0 litros de combustível
```

Prova Parcial 1 de PAC 2015/2 (3 pontos)

Escreva uma classe que represente alunos matriculados na disciplina de PAC 2015/2 e que permita registrar o nome do aluno e suas notas nas avaliações feitas durante o período. Sua classe deve herdar da classe Pessoa, ao lado. A classe pessoa não pode ser modificada.

Escreva um programa que leia os dados de N alunos e depois imprima na tela um relatório similar ao demonstrado ao lado (a média para passar direto é 7,0):

A leitura dos N alunos pode ser feita de duas formas: (a) ler um inteiro N e depois ler N alunos; ou (b) lê um aluno, pergunta se o usuário gostaria de inserir outro aluno e, se for o caso, retorna e lê mais um aluno, e assim por diante.

Observe os princípios da orientação a objetos ao escrever sua solução.

```
public class Pessoa {
    private String nome;
    public Pessoa(String nome) {
        this.nome = nome;
    }
    public String toString() {
        return nome;
    }
}
```

```
Nome,Média,Situação
Fulano,6.00,Prova Final
Cicrano,10.00,Aprovado
```