

Trabalho Prático do Módulo de Java

Enunciado

1. Âmbito e objectivos da actividade

- a) Avaliar se os conteúdos da formação foram satisfatoriamente assimilados pelos formandos;
- b) Apresentar uma experiência de projecto em contexto laboral, com prazos de entrega, na qual os formandos precisam garantir o consenso, para selecção do modelo de implementação;
- c) Proporcionar uma experiência que permita colocar em prática os conceitos adquiridos em contexto de formação, num exercício de natureza prática;

2. Dimensão dos grupos

Os grupos devem ter uma dimensão mínima de 4 elementos e máxima de 6

3. Condições da entrega

- a) Os trabalhos devem submetidos em ficheiro zip, com a seguinte nomenclatura: F16_numeroEmpregado1_numeroEmpregado2_numeroEmpregadoX_MMAAAA.zip em que MM são os dois dígitos do mês e AAAA são os 4 dígitos do ano, da edição a que correspondem;
- b) No interior do zip deverá constar um ficheiro *.txt com o nome e número de colaborador de cada um dos elementos do grupo;
- c) O ficheiro deverá ser enviado por e-mail para: nuno.mestrinho@capgemini.com, até às **23h59 de dia 25/10**. A apresentação fica agendada para dia **29/10** em horários a indicar, para cada grupo;
- d) Após a data, não serão permitidas entregas, excepto se por motivo de força maior;

4. Critérios de avaliação

- a) O trabalho será avaliado numa escala de 0 a 100;
- b) O trabalho terá discussão/apresentação obrigatória,

- c) Todos os elementos do grupo terão exactamente a mesma nota, mesmo que algum elemento se destaque pela qualidade das suas intervenções, na discussão do mesmo;
- d) A nota do grupo sofrerá penalização se algum dos elementos não conseguir fundamentar ou clarificar as decisões tomadas pela equipa, sejam elas técnicas e/ou funcionais. Isto significa que devem garantir, que todos os elementos possuem o mesmo nível de know-how sobre o trabalho;
- e) Entende-se por questões natureza funcional:
 - Qualidade dos comentários ao código produzido;
 - Clareza funcional da aplicação;
 - Interface intuitivo (diferente de bonito – isto é, se é de fácil utilização);
 - Devolução de mensagens de erro adequadas ao utilizador, relativamente a comportamentos inválidos e/ou acções não permitidas;
- f) Entende-se por questões de natureza técnica:
 - Bom mecanismo de logging de erros;
 - Controlo adequado de excepções;
 - Controlo adequado de execuções com inserção de dados inválidos, garantindo que as mesmas não interrompem a execução de forma incontrolada;
 - Utilização das estruturas de dados adequadas aos requisitos;
- g) Os grupos devem ter em consideração no código produzido, as melhoras práticas de programação e os standards e convenções definidas para programação em JAVA;
- h) A distribuição da pontuação de 0 a 100 será efectuada da seguinte forma:

Critério	Percentagem atribuída
Implementação das funcionalidades solicitadas e/ou resposta correcta às questões colocadas e/ou indicação clara e precisa do que é necessário construir	35
Aplicação executável	5
Controlo adequado de excepções, erros e/ou dados de entrada inválidos	15
Codificação de acordo com as convenções e standards para a linguagem JAVA (ex.: JavaBeans Standards)	10
Programação modular/Encapsulamento/Herança	20
Utilização adequada de estruturas de dados	5
Nível de conhecimento dos elementos do grupo equilibrado	5
Código correctamente comentado e indentado	5

- i) O trabalho terá a ponderação de 60% da avaliação final do módulo para a edição de Setembro (formação contínua) e de 100% para a edição de Junho (uma vez que, neste último caso, a formação não foi contínua).

5. Apresentação dos trabalhos

29/10 em horários a indicar

6. Temas omissos, dúvidas ou questões

Os temas que não estejam detalhados neste enunciado (que não se pretendeu exaustivo) e que se traduzam em dúvidas, omissões ou constituam questões a esclarecer devem ser submetidas para o endereço: nuno.mestrinho@capgemini.com

7. Enunciado

Devido a um problema no servidor de email da empresa “TUDO ACONTECE, SA” perderam-se todos os registos de calendário, de todos os colaboradores. Por sorte, um dos administradores tinha efectuado uma exportação dos dados constantes na base de dados, no dia imediatamente anterior, para um ficheiro json, semelhante ao que se encontra abaixo:



ficheiro.json

Todavia, como diz o ditado, um azar nunca vem só – a exportação não se encontra completa, uma vez que no ficheiro apenas constam os eventos de calendário em que cada um dos colaboradores é o respectivo organizador estando em falta os eventos em que os mesmos são apenas convidados.

O Administrador de Sistemas da “TUDO ACONTECE, SA” receoso de ser despedido, por não ter mantido um backup da base de dados decidiu contratar uma equipa da Capgemini para, a partir deste ficheiro json, construir uma aplicação com interface em Java Swing/web, que por carregamento/leitura do ficheiro consiga construir em memória o modelo de hierarquia de classes e disponibilizar métodos para obter a seguinte informação/funcionalidades:

- a) **Carregar ficheiro** – Carrega os dados do ficheiro json em memória, por leitura do ficheiro. Em alternativa poderá ser desenvolvido um método que simule o carregamento

dessa mesma informação, instanciando e declarando uma hierarquia de classes que represente essa mesma informação (a primeira opção é valorizada, mas nenhuma delas tira pontos);

- b) **Obter Meetings por colaborador (como Organizer)** – Deve permitir, por indicação do número de colaborador, devolver uma lista de eventos do tipo Meeting, para um determinado colaborador, com indicação da data, hora, local e idEvento, assim como os nomes e números de colaborador dos participantes em forma de tabela. A lista de participantes deve ser disponibilizada na última coluna dessa tabela, com os nomes separados por vírgula;
- c) **Obter Appointment por colaborador** – Lista de eventos do tipo Appointment, para um determinado colaborador, com indicação do idEvento, data, hora e local em formato de tabela;
- d) **Obter Meetings por colaborador (como Participante)** – Deve permitir, por indicação do número de colaborador, devolver uma lista de eventos do tipo Meeting, para um determinado colaborador, com indicação da data, hora e local, assim como os nomes e números de colaborador dos restantes participantes/organizer em formato tabela. A lista de participantes deve ser disponibilizada na última coluna dessa tabela, com os nomes separados por vírgula;
- e) **Permitir eliminar participantes de um determinado evento** – deverá existir um ecrã/funcionalidade que permita, por indicação do idEvento e número de colaborador, alterar a lista de participantes de um determinado evento (adição ou eliminação). Esta alteração só pode realizar-se se o número de colaborador indicado for o Organizer do evento. Da mesma forma, só deverá ser possível inserir números de colaborador que existam em sistema. Adicionalmente deve ser possível alterar a data e a hora do evento
- f) **Guardar alterações** – Após alteração da lista de participantes e/ou da data e hora deverá ser possível guardar novamente uma exportação dos dados em memória, para um ficheiro json, de forma a serem visíveis as alterações efectuadas. Em alternativa poderá ser construído um método que imprime a informação actual, após a alteração (a primeira opção é valorizada, mas nenhuma delas tira pontos);

Notas adicionais:

- 1) Os Meetings possuem lista de participantes, os Appointments não;
- 2) O idEvento é único, pelo que não podem existir dois eventos no calendário com o mesmo idEvento;

- 3) Podem existir eventos sobrepostos, em salas diferentes ou não, para o mesmo colaborador. O sistema deverá notificar ao registar a marcação de um evento, que existe conflito;
- 4) Devem ser utilizados, sempre que aplicável e/ou desejável, conceitos de herança, polimorfismo, overriding, overloading e encapsulamento;
- 5) Mesmo que o código não se encontre implementado, a correcta explicação do raciocínio, das decisões tomadas, dos controlos funcionais que devem ser tidos em conta e/ou do pseudo-código criado serão valorizados – e quando adequadamente fundamentados atribuída a cotação por completo;
- 6) As valorizações que se indicam, por implementação dos casos de maior complexidade de implementação, apenas se aplicam nas situações em que os grupos tenham perdido pontos, até ao máximo de 50% da cotação perdida, com excepção dos 5 pontos que são retirados pela não execução da aplicação – que não podem ser recuperados.
Isto é, se a aplicação executa e um grupo perdeu 20 pontos em toda a implementação, se tiver implementado:
 - 1 das situações mais complexas – recupera 50% de 20 pontos perdidos, que são depois divididos por 2 = 5 – fica com 85%;
 - 2 das situações mais complexas – recupera 50% de 20 pontos perdidos = 10 – fica com 90%;
- 7) Se existir algum tipo de dúvida ou questão na interpretação do ficheiro json, podem enviar email que terei todo o gosto em detalhar.



Sobre a Capgemini

Líder mundial em serviços de consultoria, tecnologia e transformação digital, a Capgemini encontra-se na vanguarda da inovação e aborda toda a amplitude de soluções para os seus clientes nas áreas da cloud, digital e plataformas. Com base nos seus 50 anos de história e profundo conhecimento das indústrias e sectores onde actua, a Capgemini ajuda os seus clientes a alcançarem os seus objectivos através de uma variedade de serviços, desde a estratégia às operações. A Capgemini é motivada pela convicção de que o valor da tecnologia vem de, e através, das pessoas. É uma empresa multicultural com 200 mil colaboradores em mais de 40 países. O Grupo reportou em 2017 uma receita global de 12,8 mil milhões de euros.



People matter, results count.

A informação contida neste documento é confidencial e propriedade do Grupo Capgemini.
Copyright © 2018 Capgemini. Todos os direitos reservados.