

Aluno: _____

2ª Avaliação Individual - 2013.1

Questão 1) (2,0) Você trabalha na empresa *Everything4Sale Inc.*, desenvolvendo um novo *site* de comércio eletrônico onde as compras efetuadas devem poder ser compartilhadas em diversas redes sociais. Novas redes sociais devem poder ser facilmente suportadas no futuro através de *plugins*. O diagrama abaixo à esquerda foi criado pelo projetista para que o núcleo do *framework* se comunique uniformemente com qualquer rede social. O seu papel é implementar a comunicação com o *Facebook*. Para isso, você descobriu que a biblioteca *RestFB* (www.restfb.com) já realiza estas operações (conforme descrito abaixo à direita), porém utiliza uma API diferente. Proponha uma solução para este problema através da utilização de um *Adapter* de objeto. Apresente o diagrama de classes da sua solução e todo o código-fonte do *Adapter*.

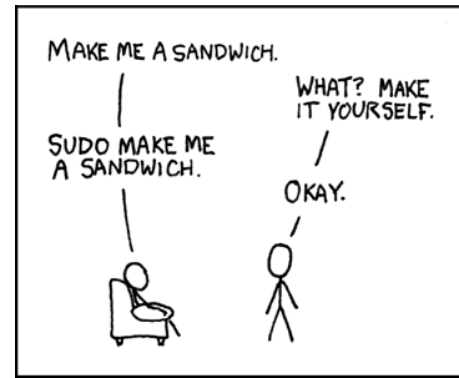
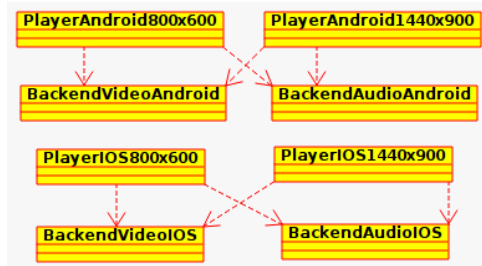
<pre>classDiagram class FrameworkCore { <<client>> } class ISocialNetwork { <<interface>> +publishOrder(user : string, orderMessage : string) } class GooglePlus class Facebook class LinkedIn class Twitter FrameworkCore ..> ISocialNetwork GooglePlus .. > ISocialNetwork Facebook .. > ISocialNetwork LinkedIn .. > ISocialNetwork Twitter .. > ISocialNetwork</pre>	<pre>01. // Publishing a simple message. 02. // FacebookType represents any Facebook Graph Object that 03. // has an ID property. 04. 05. FacebookType publishMessageResponse = 06. facebookClient.publish("user/feed", FacebookType.class, 07. Parameter.with("message", "RestFB test")); 08. 09. out.println("Published message ID: " + 10. publishMessageResponse.getId());</pre>
---	--

Questão 2) (2,0) Você trabalha como desenvolvedor no IFBA e está desenvolvendo um sistema para automação dos processos de negócio da instituição. Para o processo “Requerimento de 2ª Chamada de Avaliação” uma série de ações devem poder ser executadas após a conclusão no processo. Conforme apresentado abaixo à direita, o usuário poderá configurar o sistema em *run-time* para indicar quais ações deverão ser realizadas após a conclusão da tramitação. Novas ações devem poder ser adicionadas como *plugins*. Todas estas operações devem poder ser realizadas sem nenhum tipo de alteração no núcleo do *framework*, ou seja, ele continuará simplesmente sempre invocando a operação *finalizeProcess()*. Apresente o diagrama de classes da sua solução, as APIs das classes envolvidas e indique qual(is) padrão(ões) foi(oram) utilizado(s).

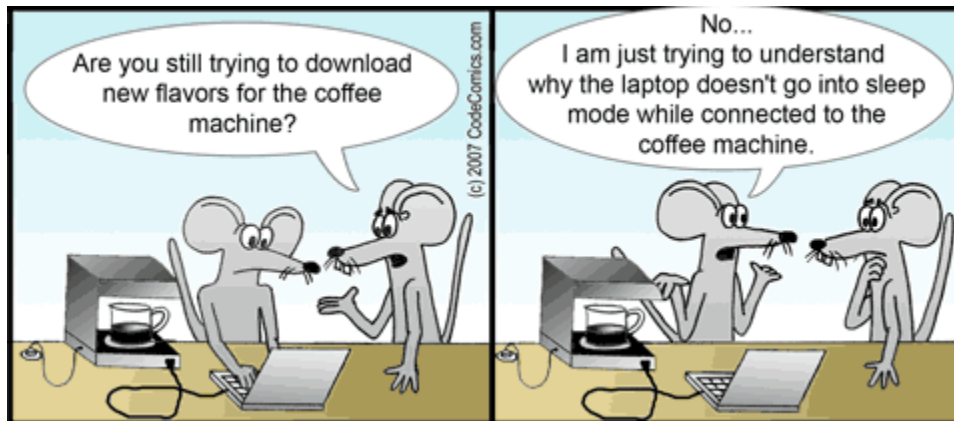
<pre>classDiagram class FrameworkCore { <<client>> } class IBusinessProcess { <<interface>> +finalizeProcess() } FrameworkCore ..> IBusinessProcess</pre>	
--	--

Questão 3) (2,0) Apresente e explique um exemplo inédito e real de uso do *Composite*. Apresente o diagrama de classes da sua solução, o código-fonte da classe *Composite* e o código-fonte do cliente que utiliza o *pattern*.

Questão 4) (2,0) Você está desenvolvendo um aplicativo *cross-platform* para exibição de filmes que requer a utilização de um *player* de vídeo e de um *player* de áudio. O sistema deve executar no *Android* e no *iOS*, em dispositivos com diferentes *form-factors* (resoluções de tela). O diagrama abaixo apresenta a solução atual. Refatore o projeto, apresentando uma versão melhorada deste diagrama de classes (com assinaturas dos métodos) e indique qual(is) padrão(ões) foi(oram) utilizado(s).



Questão 5) (2,0) Explique qual a necessidade de uso dos *smart pointers* e que tipo de benefício se obtém com a sua utilização. Apresente um exemplo de código.



Boa Sorte !