**Logotipo, nome da empresa

Descrição gerada automaticamente**Uma imagem contendo desenho, placar

Descrição gerada automaticamente

## Gamificação com Padrões – Instruções

ITA – Desenvolvimento Ágil com Padrões de Projeto (Coursera)

Aluno: Renato Sampaio Albuquerque

O objetivo dessa tarefa é fazer um componente de gamificação, que armazena diferentes tipos de conquista de um determinado usuário do sistema. Para implementar esse componente, a especificação lista alguns padrões de projeto vistos no curso que precisam ser utilizados. Pede-se também a implementação de um pequeno exemplo para o uso e teste do componente.

A correção do exercício levará em consideração:

* Funcionalidade criada
* Utilização dos padrões solicitados
* Criação dos testes de unidade

**Representação das conquistas**

Crie uma classe chamada ***Achievement*** para representar uma conquista. Cada instância deve possuir uma propriedade "name" que identifica o **Achievement**. Essa classe deve possuir as seguintes subclasses para representar diferentes tipos de conquista:

* ***Points***: possui um valor numérico. Quando for adicionado um ***Achievement*** desse tipo com q quantidade de pontos ganhos, deve ser somado ao anterior. Não devem haver 2 ***Achievement*** do tipo ***Points*** com o mesmo nome para um usuário.
* ***Badge***: representa um objetivo. Não faz sentido adicionar dois **Achievement** desse tipo com o mesmo nome para um mesmo usuário.

A diferença de como cada classe deve ser tratada ao ser adicionada deve estar em cada uma delas! Crie um método na superclasse que delega essa tarefa para um método implementado na subclasse.

**Armazenamento das conquistas**

O componente deve possibilitar diversas formas de armazenamento das informações. Para esse exercício iremos utilizar apenas o armazenamento em memória.

Crie uma interface chamada ***AchievementStorage***, com os métodos "*void addAchievement(String user, Achievement a)*", "*List<Achievement> getAchievements(String user)*" e "*Achievement getAchievement(String user, String achievementName)*". Essa interface terá apenas uma implementação chamada ***MemoryAchievementStorage*** que irá armazenar essas informações em memória.

Crie uma classe chamada ***AchievementStorageFactory*** com o método estático "*AchievementStorage getAchievementStorage()*". Essa classe deve retornar a instância de ***AchievementStorage*** configurada através de outro método "*void setAchievementStorage(AchievementStorage a)*" no início da aplicação. Deve ser utilizado um padrão que permita que exista apenas uma instância de ***AchievementStorage*** sendo utilizada em toda aplicação.

**Invocação da Funcionalidade de Gamification**

A ideia é que a chamada do componente de Gamification ocorra de forma transparente à aplicação. Dessa forma a adição dos Achievements irá ocorrer a partir de um proxy que irá envolver as classes da aplicação. O proxy só deve chamar o componente de gamificação depois que o método original for invocado na classe que está sendo encapsulada.

Para o exemplo, deve ser criado o proxy para a interface chamada ***ForumService*** com os seguintes métodos e regras de gamification:

* void addTopic(String user, String topic) - Deve adicionar 5 pontos do tipo "CREATION". Deve adicionar o bagde "I CAN TALK"
* void addComment(String user, String topic, String comment) - Deve adicionar 3 pontos do tipo "PARTICIPATION". Deve adicionar o badge "LET ME ADD"
* void likeTopic(String user, String topic, String topicUser) - Deve adicionar 1 ponto do tipo "CREATION".
* void likeComment(String user, String topic, String comment, String commentUser) - Deve adicionar 1 ponto do tipo "PARTICIPATION".

O proxy criado deve-se chamar ***ForumServiceGamificationProxy.***

PS: não é preciso criar a implementação da interface, somente o proxy.

**Observando as Conquistas**

Deve haver uma interface chamada ***AchievementObserver*** com o método "*void achievementUpdate(String user, Achievement a)*". Instâncias de classes que implementam essa interface podem ser adicionadas em classes com a interface ***AchievementStorage***. Toda vez que um novo Achievement for recebido, todas as classes desse tipo precisam ser notificadas. Se for do tipo ***Points***, a notificação acontece com a quantidade total de pontos.

Para o exemplo, devem ser implementadas os seguintes observadores:

* Quando a quantidade de pontos do tipo "CREATION" chegar a 100, o usuário deve receber o badge "INVENTOR"
* Quando a quantidade de pontos do tipo "PARTICIPATION" chegar a 100, o usuário deve receber o badge "PART OF THE COMMUNITY"

**Teste da Implementação**

Deve ser feito o teste automatizado que verifica o funcionamento da implementação. Deve-se fazer o teste em cima do proxy criado e adicionando na classe ***AchievementStorage*** os observadores pedidos. Um nova instância de ***MemoryAchievementStorage*** deve ser configurada a cada teste. Deve-se criar um *mock object* para fazer o papel da classe que está sendo encapsulada pelo proxy.

Os seguintes testes devem ser feitos:

* Chamar o método addTopic() e ver se os achievements foram adicionados da forma correta
* Chamar o método addComment() e ver se os achievements foram adicionados da forma correta
* Chamar o método likeTopic() e ver se os achievements foram adicionados da forma correta
* Chamar o método likeComment() e ver se os achievements foram adicionados da forma correta
* Chamar o método addTopic() duas vezes e ver se os pontos foram somados e se o badge está presente apenas uma vez
* Fazer um teste invocando vários métodos e verificar se o resultado é o esperado.
* Fazer o mock lançar uma exceção para algum método e verificar se os Achievements não foram adicionados.
* Atingir 100 pontos de "CREATION" e verificar se o usuário recebe o badge "INVENTOR"
* Atingir 100 pontos de "PARTICIPATION" e verificar se o usuário recebe o badge "PART OF THE COMMUNITY"