

Instituto Federal de Educação Ciências e Tecnologia da Paraíba – IFPB

Campos Cajazeiras

Curso: Análise e Desenvolvimento de Sistemas

Disciplina: Padrões de Projetos

Professor: Diogo Moreira

Alunos:

Natarajan Rodrigues

Victor Hugo Ribeiro

Wensttay Sá

Relatório de Padrões Utilizados no Exercício 1

Cajazeiras

Março / 2016

Histórico das Versões do Documento

| Data | Versão | Descrição | Autor |
|------------|--------|--|-----------|
| 08/03/2016 | 1.0 | Padrões Comportamentais (State, Strategy e Observer) | Natarajan |

Relatório de Padrões Utilizados no Exercício 1

O presente relatório tem como objetivo explicitar sucintamente a utilização de padrões de projeto comportamentais utilizados na resolução do exercício 1 da disciplina Padrões de Projetos, que consiste no problema da locadora de jogos denominada Pattern Games. Abaixo segue alguns detalhes da implementação da resposta ao exercício necessárias a elucidar os caminhos adotados.

Tipo de solução adotada

O grupo optou por construir um aplicativo web, utilizando Maven para administração de repositórios. Notar a necessidade de rodar versão 1.8 do JDK para pleno uso de alguns detalhes como Comparators disponíveis nessa versão, embora o Maven já possa configurar essa especificação automaticamente.

P.S.: é de praxe que na avaliação da solução seja necessário alterar a data do sistema onde está rodando o aplicativo para verificar se a solução apresentada condiz com o esperado. No entanto, alguns comportamentos indesejados podem acontecer devido a alteração de data “on the fly”; ou seja, se o sistema estiver rodando e com páginas abertas, a alteração do relógio pode gerar alguma inconsistência devido à perda de sessão ou outras mudanças internas relacionadas à administração do Tomcat. A maioria destas alterações foi tratada na forma de redirecionamento do fluxo da aplicação para a página index, mas alguns cenários (envolvendo a mudança de data atual para uma data futura e comportamento de algumas páginas que usam Ajax) não puderam ser contornados. Em tempo, imaginamos que a solução para estes casos é necessária quando se trata de um ambiente não educacional (leia-se “real”), mas acreditamos que foge um pouco do escopo do objetivo do exercício. Por este motivo, ao alterar a data do sistema solicitamos, caso surjam comportamentos inesperados, voltar para tela index utilizando um dos atalhos disponíveis (logo PatternGames do canto esquerdo superior ou link “Escolher outro cliente”).

Bibliotecas Extra Utilizadas

A solução faz uso da biblioteca Caellum disponível em “stella.caelum.com.br” para verificação do CPF dos clientes. Também utiliza a API JavaMail, da Oracle, para envio de emails que notifica aos clientes sobre a devolução de jogos, conforme solicitado.

Padrões Comportamentais na solução

O uso dos padrões comportamentais de projeto estão circunscritos a 3 cenários/objetivos do exercício:

- 1) administrar tipos diferentes de comportamento de locações de jogos (uma comum e outra especial, para fins de semana);
- 2) gerar comportamentos específicos de um jogo de acordo com o estado em que se encontra o mesmo (alugado ou disponível);
- 3) notificar clientes quando o jogo mudar para um estado de disponível;

Respectivamente, os padrões utilizados foram:

1) Strategy: através desse padrão foi possível montar estratégias diferentes do cálculo do valor da locação de acordo com a data em que a mesma ocorre (um para fim de semana e um para os demais dias). Criou-se a interface strategy e as classes finais que a implementam com as especificidades.

2) State: o uso deste padrão possibilitou programar as respostas que ocorriam nas operações de alugar ou devolver os jogos. Cada estado pode realiza as mesmas operações, cada estado a seu modo. A operação de alugar um jogo no estado disponível e devolver um jogo alugado denotam estados de normalidade (apenas alterando o jogo para um estado novo). As tentativas de alugar um jogo já alugado ou devolver um jogo já disponível retornam exceções para cada caso, conforme solicita o exercício.

3) Observer: o uso desse padrão está atrelado ao State. Ao realizar mudar o estado de jogo de alugado para disponível (devolução), clientes cadastrados como observadores são notificados que o jogo está agora disponível na locadora. A solução usando o padrão Observer constituiu em tornar os objetos Cliente como observadores dos objetos Jogo (que eram observáveis, ou seja, eram o alvo de observação, ou objeto “interessante”). A partir daí, a implementação administra a inclusão/exclusão de clientes como observadores de um determinado jogo que, ao ser devolvido, notifica todos os seus interessados através de email.