Especificação do Projeto de Desenvolvimento de Sistema de Software de Museu

Lista 1

Instruções Gerais:

- Os exercícios devem ser apresentados na mesma ordem dos enunciados e devem conter uma sequência lógica. Os exercícios são referentes ao mesmo sistema, portanto deve haver coerência entre eles. Os exercícios que não estiverem numa sequência lógica serão devidamente descontados.
- A Lista deve ser realizada em grupo de 4 a 6 integrantes, mas apenas um aluno precisa entregar a Lista por e-mail. Favor copiar os outros integrantes do grupo em cada e-mail enviado. Exceções devem ser tratadas com o próprio professor com antecedência.
- Os slides dos capítulos 4, 5 e 8 podem apoiar a realização da maioria dos exercícios desta Lista.
- Os diagramas devem ser construídos em alguma ferramenta CASE, mas a Lista deve ser entregue no formato digital no e-mail pwvendramel@gmail.com em um único arquivo PDF até às 07h00 de 08/09/2017.
- Para cada exercício em branco, incompleto ou que não atenda o enunciado, será subtraído 1 ponto da Nota de Listas conforme explicado no primeiro dia de aula. Em determinados casos, o desconto pode ser de 0,5 ponto.
- Listas com respostas suspeitas de plágio serão devidamente anuladas e “zeradas”. Os exercícios com respostas iguais entre grupos diferentes serão anulados e descontados. O aluno poderá ser convidado para resolver alguma questão durante a aula com o objetivo de validar os exercícios da Lista.

Analise os Requisitos Funcionais de um Projeto de Desenvolvimento de Software de Museu apresentados logo abaixo e realize os exercícios na sequência.

RF01: Gestão de Visita
Este requisito deve gerenciar o fluxo de visitantes do museu por meio de um cartão magnético fornecido para o visitante na entrada, sendo possível armazenar os dados básicos de cada visitante para envio de convite de exposição ou evento.

RF02: Gestão de Acervo
Este requisito deve gerenciar as obras presentes no museu por meio de um cadastro unificado de acervo físico (obras em geral) e virtual (Documentos Históricos), onde poderão ser realizadas consultas das obras.

RF03: Gestão de Exposição
Este requisito deve gerenciar as exposições, temporárias ou permanentes do museu, permitindo listar as obras exibidas em cada exposição, além de mostrar a salas reservadas e o máximo de visitantes.

RF04: Gestão de Restauração
Este requisito deve gerenciar o processo de restauração de obras, desde a abertura da solicitação de serviço, duração e andamento até a conclusão da restauração da obra.

RF05: Gestão de Evento
Este requisito deve gerenciar as informações do evento a ser realizado, assim como quem será o responsável e o número de visitantes para o evento em questão.

RF06: Gestão de Venda de Souvenir
Este requisito deve gerenciar a loja de souvenirs, desde o seu estoque até a aquisição de novos produtos para a loja. Fornece relatórios de vendas para o gerente deste setor.

Para um melhor entendimento do domínio da aplicação, o estudante deve realizar visitas virtuais nos diversos museus no endereço http://www.saopaulo.sp.gov.br/conhecasp/cultura_museus. Outra visita virtual pode ser realizada no endereço http://www.louvre.fr/en.

Parte A: Exercícios sobre o Projeto

1- Especifique textualmente um caso de uso (CSU01) para o RF01, apresentando os fluxos (cenários) principal, alternativo e de exceção. Utilize um template que seja inteligível. Possíveis casos de uso <<extend>> ou <<include>> podem ser especificados junto com o caso de uso principal.

2- Especifique textualmente um caso de uso (CSU02) para o RF02, apresentando os fluxos (cenários) principal, alternativo e de exceção. Utilize um template que seja inteligível. Possíveis casos de uso <<extend>> ou <<include>> podem ser especificados junto com o caso de uso principal.

3- Especifique textualmente um caso de uso (CSU03) para o RF03, apresentando os fluxos (cenários) principal, alternativo e de exceção. Utilize um template que seja inteligível. Possíveis casos de uso <<extend>> ou <<include>> podem ser especificados junto com o caso de uso principal.

4- Especifique textualmente um caso de uso (CSU04) para o RF04, apresentando os fluxos (cenários) principal, alternativo e de exceção. Utilize um template que seja inteligível. Possíveis casos de uso <<extend>> ou <<include>> podem ser especificados junto com o caso de uso principal.

5- Especifique textualmente um caso de uso (CSU05) para o RF05, apresentando os fluxos (cenários) principal, alternativo e de exceção. Utilize um template que seja inteligível. Possíveis casos de uso <<extend>> ou <<include>> podem ser especificados junto com o caso de uso principal.

6- Especifique textualmente um caso de uso (CSU06) para o RF06, apresentando os fluxos (cenários) principal, alternativo e de exceção. Utilize um template que seja inteligível. Possíveis casos de uso <<extend>> ou <<include>> podem ser especificados junto com o caso de uso principal.

7- Modele um Diagrama de Casos de Uso (DCU) com base nas especificações textuais.

8- Modele uma VCP para o CSU01, utilizando a categorização BCE. A classe de controle deve apresentar dois métodos no mínimo e as classes de entidade devem apresentar seus devidos atributos e métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU01.

9- Modele uma VCP para o CSU02, utilizando a categorização BCE. A classe de controle deve apresentar dois métodos no mínimo e as classes de entidade devem apresentar seus devidos atributos e métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU02.

10- Modele uma VCP para o CSU03, utilizando a categorização BCE. A classe de controle deve apresentar dois métodos no mínimo e as classes de entidade devem apresentar seus devidos atributos e métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU03.

11- Modele uma VCP para o CSU04, utilizando a categorização BCE. A classe de controle deve apresentar dois métodos no mínimo e as classes de entidade devem apresentar seus devidos atributos e métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU04.

12- Modele uma VCP para o CSU05, utilizando a categorização BCE. A classe de controle deve apresentar dois métodos no mínimo e as classes de entidade devem apresentar seus devidos atributos e métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU05.

13- Modele uma VCP para o CSU06, utilizando a categorização BCE. A classe de controle deve apresentar dois métodos no mínimo e as classes de entidade devem apresentar seus devidos atributos e métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU06.

14- Modele um Diagrama de Classes de Projeto a partir das VCPs modeladas e mantenha a utilização da categorização BCE. Os devidos atributos e métodos devem continuar sendo exibidos. As multiplicidades dos relacionamentos devem ser exibidas.

15- Qual é a classe de entidade mais coesa e a menos coesa do diagrama de classes de projeto? Justifique a tua resposta.

16- Qual é a classe de entidade mais acoplada e a menos acoplada do diagrama de classes de projeto? Justifique a tua resposta.

17- Modele duas relações de gen/espec e ative o princípio de polimorfismo universal de inclusão em cada uma delas. Justifique a razão de existência de cada gen/espec e das operações polimórficas. As relações de gen/espec violam o Princípio de Liskov? Justifique a tua resposta.

18- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar as relações de gen/espec e as operações polimórficas.

19- Modele três classes enumeradas e utilize as mesmas como tipos de atributos. Justifique a existência de cada uma das classes enumeradas modeladas.

20- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar as três classes enumeradas.

21- Modele seis membros estáticos, sendo três atributos e três métodos. Justifique a criação de existência de cada um dos membros estáticos modelados.

22- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar os seis membros estáticos.

23- Transforme todos os relacionamentos de associação ou agregação entre as classes de entidade e todos os relacionamentos de associação entre as classes de fronteira e controle para dependências estruturais. Explique a vantagem e desvantagem desse tipo de dependência.

24- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar as dependências estruturais.

25- Transforme todos os relacionamentos de associação entre as classes de controle e entidade para dependências não estruturais por parâmetro. Explique a vantagem e desvantagem desse tipo de dependência.

26- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar as dependências não estruturais por parâmetro.

27- Transforme todos os relacionamentos de associação entre as classes de controle e entidade para dependências não estruturais por variável local. Explique a vantagem e desvantagem desse tipo de dependência.

28- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar as dependências não estruturais por variável local.

Parte B: Atividade de Abstração

Analise o diagrama de classes abaixo. Descreva a coesão e acoplamento desse diagrama. Modele um diagrama de classes que seja melhor em termos de coesão e acoplamento e explique o porquê do novo diagrama de classes ser mais vantajoso do que o abaixo.