

Projeto de Sistema de Software de Farmácia

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 impressa. Não aceitas exceções na redefinição dos grupos ou mesmo sobre a participação dos integrantes;
- Os diagramas devem ser construídos em alguma ferramenta CASE de forma inteligível, devendo ser entregue até dia 13/09/2019;
- Os slides dos capítulos 4, 5, 7 e 8 podem apoiar a realização da maioria dos exercícios desta Lista;
- Para cada exercício em branco, incompleto ou que não atenda o enunciado, será subtraído 1,0 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 questões durante a aula com o objetivo de validar os exercícios da Lista;
- As Listas que não respeitarem as instruções gerais supracitadas não serão aceitas e, consequentemente, a lista será “zerada”;

Analise a Especificação do Projeto de Desenvolvimento de Sistema de Software de Farmácia e realize os exercícios na sequência.

Requisitos Funcionais

RF01: Manter o cadastro de clientes normais e especiais;

RF02: Manter o cadastro de medicamentos;

RF03: Gerenciar a abertura e o encerramento do caixa;

RF04: Gerenciar a venda de medicamentos com registro de pagamento em dinheiro ou cartão e com possibilidade de concessão de desconto e emissão de nota fiscal eletrônica;

RF05: Gerenciar o estoque, permitindo entrada e saída. A entrada é realizada na compra de produtos e a saída está relacionada com a venda dos produtos

Regras de Negócio

RN01: todo aposentado tem direito a 20% de desconto nas compras de qualquer medicamento. Para isso é necessário o cadastro dos aposentados, conforme RF01;

RN02: num mesmo dia, o caixa pode ser aberto e encerrado quantas vezes forem necessárias, conforme RF03;

RN03: quando o pagamento for realizado em dinheiro, o cliente terá desconto de 5%, exceto os aposentados conforme RN01;

RN04: será necessário manter dois tipos de usuário: atendente e gerente. O gerente pode executar todas as funções do sistema; o atendente não pode encerrar o caixa nem conceder descontos especiais.

RN05: clientes que apresentarem a carteira do plano de saúde terá um desconto de 5% em cada medicação. Produtos de higiene, perfumaria e outros não terão descontos

Parte A: Exercícios sobre o Projeto

- 1- Especifique textualmente o Caso de Uso (CSU01) para o RF01, apresentando os fluxos (cenários) principal, alternativo e/ou de exceção, de acordo com o template disponibilizado. Vale lembrar que uma ou mais regras de negócio podem fazer parte da especificação do caso de uso.
- 2- Especifique textualmente o Caso de Uso (CSU02) para o RF02, apresentando os fluxos (cenários) principal, alternativo e/ou de exceção, de acordo com o template disponibilizado. Vale lembrar que uma ou mais regras de negócio podem fazer parte da especificação do caso de uso.
- 3- Especifique textualmente o Caso de Uso (CSU03) para o RF03, apresentando os fluxos (cenários) principal, alternativo e/ou de exceção, de acordo com o template disponibilizado. Vale lembrar que uma ou mais regras de negócio podem fazer parte da especificação do caso de uso.
- 4- Especifique textualmente o Caso de Uso (CSU04) para o RF04, apresentando os fluxos (cenários) principal, alternativo e/ou de exceção, de acordo com o template disponibilizado. Vale lembrar que uma ou mais regras de negócio podem fazer parte da especificação do caso de uso.
- 5- Especifique textualmente o Caso de Uso (CSU05) para o RF05, apresentando os fluxos (cenários) principal, alternativo e/ou de exceção, de acordo com o template disponibilizado. Vale lembrar que uma ou mais regras de negócio podem fazer parte da especificação do caso de uso.
- 6- Modele um Diagrama de Casos de Uso (DCU) com base nas especificações textuais.
- 7- 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 no mínimo quatro atributos e dois métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU01.
- 8- 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 no mínimo quatro atributos e dois métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU02.
- 9- 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 no mínimo quatro atributos e dois métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU03.
- 10- 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 no mínimo quatro atributos e dois métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU04.
- 11- 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 no mínimo quatro atributos e dois métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU05.
- 12- Modele um diagrama de sequência para o CSU01. Se a interação for complexa, definam os quadros de interação.
- 13- Modele um diagrama de sequência para o CSU02. Se a interação for complexa, definam os quadros de interação.
- 14- Modele um diagrama de sequência para o CSU03. Se a interação for complexa, definam os quadros de interação.
- 15- Modele um diagrama de sequência para o CSU04. Se a interação for complexa, definam os quadros de interação.
- 16- Modele um diagrama de sequência para o CSU05. Se a interação for complexa, definam os quadros de interação.
- 17- Modele um Diagrama de Classes de Projeto a partir das VCPs modeladas e dos diagramas de sequência, e mantenha a utilização da categorização BCE. Os devidos atributos e métodos devem continuar sendo exibidos. As multiplicidades dos relacionamentos devem ser apresentadas.

- 18- Qual é a classe de entidade mais coesa e a menos coesa do diagrama de classes de projeto? Justifique a tua resposta.
- 19- Qual é a classe de entidade mais acoplada e a menos acoplada do diagrama de classes de projeto? Justifique a tua resposta.
- 20- 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.
- 21- Apresente a estrutura básica de código em JAVA, Python ou C++ para implementar as relações de gen/espec e as operações polimórficas.
- 22- As relações de gen/espec modeladas violam o Princípio de Liskov? Justifique a tua resposta.
- 23- Quais restrições {OCL} sobre gen/espec são aplicáveis nas relações modeladas? Justifique a tua resposta.
- 24- Modele três classes enumeradas e utilize as mesmas como tipos de atributos. Justifique a existência de cada uma das classes enumeradas modeladas.
- 25- Apresente a estrutura básica de código em JAVA, Python ou C++ para implementar as três classes enumeradas.
- 26- Modele seis membros estáticos, sendo três atributos e três métodos. Justifique a criação de existência de cada um dos membros estáticos modelados.
- 27- Apresente a estrutura básica de código em JAVA, Python ou C++ para implementar os seis membros estáticos.
- 28- Transforme todos os relacionamentos de associação ou agregação entre as classes de entidade e todos os relacionamentos entre as classes de fronteira e controle para dependências estruturais. Explique a vantagem e desvantagem desse tipo de dependência.
- 29- Apresente a estrutura básica de código em JAVA, Python ou C++ para implementar as dependências estruturais.
- 30- Modele concomitantemente as dependências não estruturais por parâmetro e por variável local entre as classes de controle e modelo. Explique a vantagem e desvantagem desse tipo de dependência.
- 31- Apresente a estrutura básica de código em JAVA, Python ou C++ para implementar as dependências não estruturais por parâmetro e por variável local.
- 32- Modele concomitantemente as classes parametrizadas com a estrutura <List> e <Set> para resolver o lado muitos dos relacionamentos. Para cada classe parametrizada modelada, justifique o motivo de ter escolhido o tipo de estrutura de dados.
- 33- Apresente a estrutura básica de código em JAVA, Python ou C++ para implementar as classes parametrizadas com a estrutura <List> e <Set>.
- 34- Modele a estrutura <TreeSet>. Justifique a razão dessa estrutura no seu diagrama. Represente graficamente como essa árvore trabalharia em tempo de execução.
- 35- Apresente a estrutura básica de código em JAVA, Python ou C++ para implementar a estrutura de dados <TreeSet>.

Parte B: Exercícios de Abstração

Os exercícios de abstração desta lista devem ser feitos com base no artigo “No Silver Bullet: essence and accidents of software engineering” de Frederick P. Brooks Jr. Esse artigo também está disponibilizado no SIGA em partes, pois não foi possível fazer o upload de arquivo maior que 2 MB.

- 36- Elabore um jogo de palavras cruzadas contemplando 15 termos identificados no artigo.

- 37- Por que o autor afirma que não existe bala de prata?
- 38- Brooks aborda dois termos importantes em seu artigo: essência e acidente. O que significa cada um desses termos? Dê um exemplo de essência e outro de acidente.
- 39- Quais são as dificuldades essenciais apresentadas no artigo? Alguma bala de prata resolve essas dificuldades? Justifique a tua resposta.
- 40- O paradigma orientado a objetos é uma bala de prata para resolver a questão da essência de software? Justifique a tua resposta.
- 41- A UML é uma bala de prata para resolver a questão da essência de software? Justifique a tua resposta.
- 42- Ferramenta CASE é uma bala de prata? Justifique a tua resposta.
- 43- Por que se compara um software a um lobisomem? Produza uma charge manualmente contemplando o paradigma orientado a objetos e a história do lobisomem.