

Lista 1

Projeto de Sistema de Software de Clínica Veterinária

Instruções Gerais. Leiam com bastante atenção.

- a) Esta lista de exercícios corresponde à primeira parte (40%) da avaliação que compõe a nota da N1. A segunda parte (60%) da avaliação é a prova individual. Esta lista tem três seções, sendo: A) Exercícios sobre o Projeto; B) Exercícios de Abstração; C) Exercícios sobre o Artigo;
- b) Os exercícios devem ser apresentados na mesma ordem dos enunciados, havendo assim uma sequência lógica. Vale ressaltar que os exercícios da Seção A são referentes ao mesmo projeto de sistema de software, portanto deve haver coerência entre eles. Cada exercício que não for apresentado dentro de uma sequência lógica será devidamente anulado e o desconto será de 0,5 ponto para tal;
- c) Para cada exercício em branco será descontado 1,0 ponto para tal; para cada exercício incompleto, ilegível ou que não atenda o enunciado, será descontado 0,5 ponto para tal;
- d) As listas com respostas suspeitas de plágio serão devidamente anuladas e “zeradas”. Para cada exercício que apresentar a resposta igual em duas ou mais listas será devidamente anulado e o desconto será de 1,0 ponto para tal;
- e) A lista deve ser realizada em grupo de 4 a 5 alunos. Um aluno poderá ser convidado durante a aula para resolver exercícios com o objetivo de validar a lista realizada;
- f) Os diagramas devem ser construídos em alguma ferramenta CASE; a lista deve ser entregue no formato impresso e legível até às 22h00 de 13/04/2020;
- g) Os slides dos capítulos 4, 5, 7 e 8, disponibilizados no SIGA, podem apoiar a realização da maioria dos exercícios desta lista;
- h) Exceções sobre as regras supracitadas devem ser tratadas com o próprio professor antecipadamente.

Analise o Cenário e o Diagrama de Casos de Uso (DCU) do Projeto de Sistema de Software de Clínica Veterinária apresentados logo abaixo e realize os exercícios na sequência.

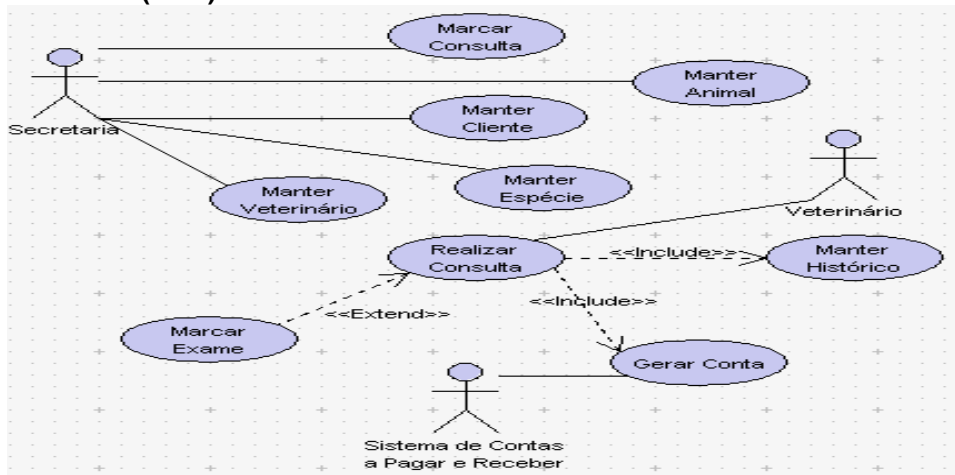
Cenário

Um cliente dirige-se a Clínica onde marca uma **consulta** com a **secretária**, fornecendo suas **informações pessoais e do animal** que deseja tratar. Se o cliente ou o animal não estiver cadastrado no sistema ou possua algum dado que precise ser atualizado, a secretária deve **atualizar o cadastro**. Em cada consulta, o cliente deve informar os **sintomas aparentes do animal**, os quais devem ser registrados. Um **tratamento** pode ser **encerrado** em apenas uma consulta, quando se tratar de algo simples ou **pode se arrastar por muitas consultas**, dependendo do **diagnóstico** do veterinário. Durante uma consulta, o veterinário pode marcar **exames** para o animal, a serem trazidos na consulta seguinte. O **pedido dos exames** e seus **resultados** devem ser registrados no **histórico de tratamentos do animal**. A manutenção das consultas é responsabilidade exclusiva do veterinário que a realizou. Após cada sessão, **o histórico da consulta deve ser atualizado e gera-se uma conta a ser paga** pelo cliente que é controlado pelo **Sistema de Contas a Pagar e Receber** que é **integrado** ao sistema da clínica. É responsabilidade de a secretária manter os cadastros de clientes, animais, veterinários e espécies atualizados.

Regras de Negócio

- RN01: Clientes com 2 pets ou mais cadastrados devem ter desconto de 5% nas consultas;
- RN02: Em caso de cancelamento, a consulta deve ser cancelada até 24 horas antes do horário agendado, caso contrário haverá uma taxa de R\$50,00;
- RN03: Após 5 consultas, deve ser aplicado 20% de desconto na próxima consulta;
- RN04: O pagamento pode ser realizado por meio de cartão de débito, cartão de crédito, transferência bancária ou dinheiro. Quando o pagamento for à vista, deve ser aplicado 10% de desconto no valor total da conta.

Diagrama de Casos de Uso (DCU)



Seção A: Exercícios sobre o Projeto.



- 1- Especifique textualmente o caso de uso Manter Cliente (CSU01), apresentando os fluxos (cenários) principal, alternativo e de exceção, de acordo com o template disponibilizado.
- 2- Especifique textualmente o caso de uso Manter Animal (CSU02), apresentando os fluxos (cenários) principal, alternativo e de exceção, de acordo com o template disponibilizado. O caso de uso Manter Espécie deve ser adequado e especificado como caso de uso incluído (<<include>>) do caso de uso principal, isto é, o CSU02.
- 3- Especifique textualmente o caso de uso Manter Veterinário (CSU03), apresentando os fluxos (cenários) principal, alternativo e de exceção, de acordo com o template disponibilizado.
- 4- Especifique textualmente o caso de uso Marcar Consulta (CSU04), apresentando os fluxos (cenários) principal, alternativo e de exceção, de acordo com o template disponibilizado.
- 5- Especifique textualmente o caso de uso Realizar Consulta (CSU05), apresentando os fluxos (cenários) principal, alternativo e de exceção, de acordo com o template disponibilizado. Os casos de uso Manter Histórico e Gerar Conta devem ser especificados como casos de uso incluídos (<<include>>) do caso de uso principal, isto é, o CSU05. Deve surgir um novo caso de uso denominado Solicitar Exame como caso de uso estendido do caso de uso principal, isto é, o CSU05.
- 6- Especifique textualmente o caso de uso Marcar Exame (CSU06), apresentando os fluxos (cenários) principal, alternativo e de exceção, de acordo com o template disponibilizado. Este caso de uso deve ser adaptado e especificado como caso de uso principal, retirando o mesmo como caso de uso estendido (<<extend>>) do caso de uso Realizar Consulta (CSU05).
- 7- Modele um novo 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 no mínimo quatro atributos e dois 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 no mínimo quatro atributos e dois 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 no mínimo quatro atributos e dois 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 no mínimo quatro atributos e dois 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 no mínimo quatro atributos e dois 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 no mínimo quatro atributos e dois métodos. Faça também o protótipo de interface de usuário para a classe <<boundary>> do CSU006.
- 14- Modele um diagrama de sequência para o CSU01. Se a interação for complexa, definam os quadros de interação.
- 15- Modele um diagrama de sequência para o CSU02. Se a interação for complexa, definam os quadros de interação.
- 16- Modele um diagrama de sequência para o CSU03. Se a interação for complexa, definam os quadros de interação.
- 17- Modele um diagrama de sequência para o CSU04. Se a interação for complexa, definam os quadros de interação.
- 18- Modele um diagrama de sequência para o CSU05. Se a interação for complexa, definam os quadros de interação.
- 19- Modele um diagrama de sequência para o CSU06. Se a interação for complexa, definam os quadros de interação.
- 20- 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.
- 21- Qual é a classe de entidade mais coesa e a menos coesa do diagrama de classes de projeto? Justifique a tua resposta.
- 22- Qual é a classe de entidade mais acoplada e a menos acoplada do diagrama de classes de projeto? Justifique a tua resposta.
- 23- 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.
- 24- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar as relações de gen/espec e as operações polimórficas.
- 25- As relações de gen/espec modeladas violam o Princípio de Liskov? Justifique a tua resposta.
- 26- Quais restrições {OCL} sobre gen/espec são aplicáveis nas relações modeladas? Justifique a tua resposta.
- 27- Modele três classes enumeradas e utilize as mesmas como tipos de atributos. Justifique a existência de cada uma das classes enumeradas modeladas.
- 28- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar as três classes enumeradas.
- 29- Modele seis membros estáticos, sendo três atributos e três métodos. Justifique a criação de existência de cada um dos membros estáticos modelados.
- 30- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar os seis membros estáticos.
- 31- 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.
- 32- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar as dependências estruturais.

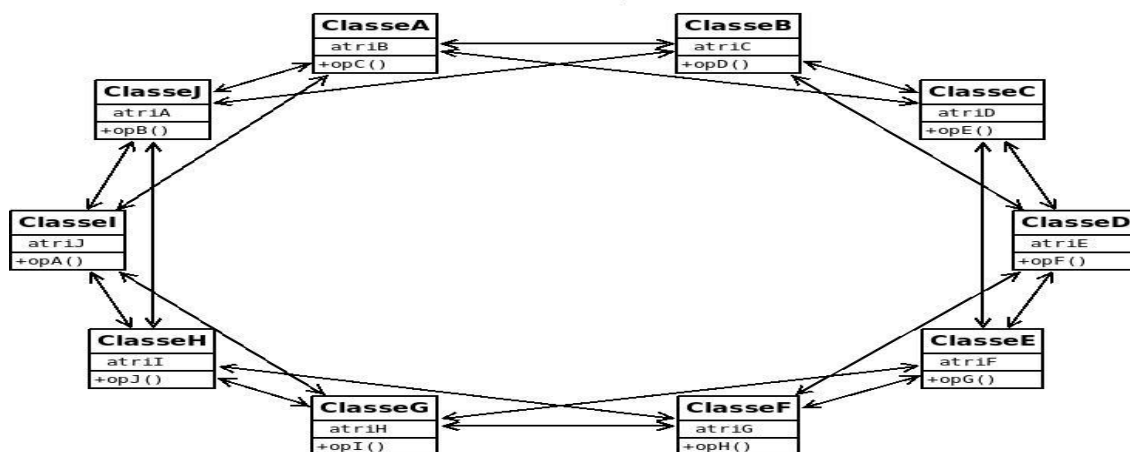
- 33- Modele concomitantemente as dependências não estruturais por parâmetro e por variável local entre as classes de controle e entidade. Explique a vantagem e desvantagem desse tipo de dependência.
- 34- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar as dependências não estruturais por parâmetro e por variável local.
- 35- 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.
- 36- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar as classes parametrizadas com a estrutura <List> e <Set>.
- 37- Modele a estrutura <TreeSet>. Justifique a razão dessa estrutura no seu diagrama. Represente graficamente como essa árvore trabalharia em tempo de execução.
- 38- Apresente a estrutura básica de código em JAVA, C# ou C++ para implementar a estrutura de dados <TreeSet>.

Seção B: Exercícios de Abstração.

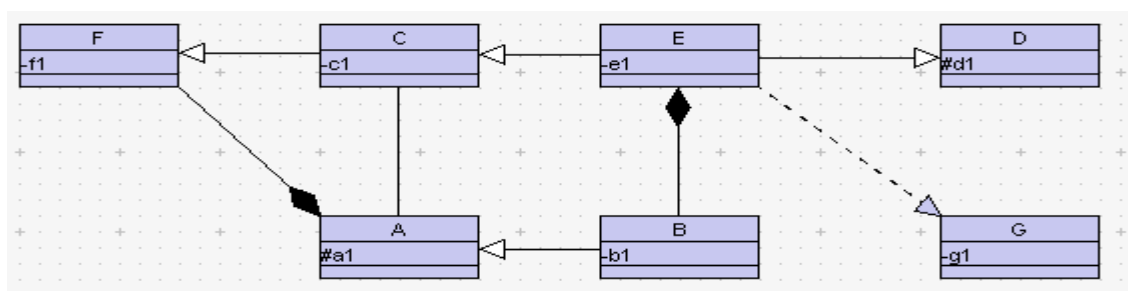
- 39- A partir do caça objetos abaixo, encontre no mínimo cinco objetos e modele uma relação todo-parte, abstraindo o nome de uma ave na classe todo e os cinco objetos encontrados como partes deste todo. Explique a solução apresentada.

A	O	L	C
P	S	O	V
E	A	A	O
O	C	I	B

- 40- 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.



- 41- Identifique cada um dos relacionamentos utilizados no diagrama de classes abaixo, inclusive se são simétricos ou assimétricos. Quais atributos a instância de cada classe pode visualizar? Justifique a tua resposta.



Seção C: Exercícios sobre o Artigo “The Elephant and the Blind Programmers” de Grady Booch. Este artigo também está disponibilizado no SIGA.

- 42- Elabore um jogo de palavras cruzadas contemplando 18 termos identificados no artigo.
- 43- Descreva as características de cada um dos programadores cegos.
- 44- Qual é a relação da história apresentada com o projeto de desenvolvimento de software?
- 45- Produza uma charge, a mão, abordando a história do elefante e dos cinco programadores. Explique o significado da sua charge.