

Tarefa 3: Conteúdo 4 - Relacionamento Entre Classes

Instruções para Entrega:

*As respostas devem ser entregues **exclusivamente por meio de repositório no GitHub**. O link do repositório Git deve ser inserido nos comentários do SIGAA, no campo correspondente à atividade.*

Organização recomendada da estrutura do repositório:

Q1/

- a) Criar uma pasta denominada diagrama/ contendo o arquivo da imagem (.jpeg ou .png) com o diagrama UML.*
- b) Criar uma pasta denominada código/ com os arquivos-fonte da aplicação desenvolvida.*
- c) Criar uma pasta denominada resposta/ com o arquivo de texto (.txt, .md ou .pdf) contendo a resposta dissertativa.*

Q1) Problema de Associação: No contexto de um Sistema de Controle de Veículos Pessoais, pode-se modelar, em UML, uma relação de associação entre as classes Pessoa e Carro. Nesse sistema, uma Pessoa pode dirigir diversos Carros ao longo do tempo, enquanto cada Carro é conduzido por apenas uma pessoa por vez. O Carro possui atributos como o modelo e um comportamento dirigir(), mas não armazena informações sobre motoristas anteriores. Já a Pessoa possui nome e o comportamento usarCarro(Carro carro), sendo responsável por registrar os veículos que já conduziu.

- a) Faça o Diagrama de Classe (lucid.app)
- b) Implemente o Código-Fonte em Java
- c) Relacione e explique os conceitos de acordo com seu código

Q2) Problema de Agregação: No Sistema de Gerenciamento de Cursos Online, a modelagem em UML envolve uma relação de agregação entre as classes Curso e Instrutor. Um Curso agrega um ou mais Instrutores, enquanto cada Instrutor pode estar vinculado a diversos cursos, mantendo sua existência independente desses vínculos. Essa característica define uma agregação, pois o ciclo de vida do instrutor não depende do curso ao qual está associado. A classe Curso possui atributos como nome, carga horária, nível e descrição, enquanto a classe Instrutor contém informações como nome, formação, e-mail e anos de experiência. O sistema deve possibilitar o cadastro de cursos, a associação de instrutores a esses cursos e a exibição de informações detalhadas de cada curso, incluindo os instrutores envolvidos, promovendo uma gestão eficiente e reutilizável dos recursos humanos e educacionais da plataforma.

- a) Faça o Diagrama de Classe (lucid.app)
- b) Implemente o Código-Fonte em Java
- c) Relacione e explique os conceitos de acordo com seu código

Q3) Problema de Composição: No Sistema de Gerenciamento de Computadores, a modelagem em UML representa uma relação de composição entre as classes Computador e PlacaMae. Nesse contexto, cada Computador é composto por uma única PlacaMãe, que não existe de forma independente, sendo criada e destruída juntamente com o computador ao qual pertence. A classe Computador armazena atributos como marca, modelo, processador e memória RAM, enquanto a classe PlacaMae contém dados como fabricante, chipset, número de slots e tipo de memória suportada. O sistema deve permitir o cadastro

de computadores completos, incluindo suas respectivas placas-mãe, além de possibilitar a exibição detalhada das informações e a atualização dos dados da placa-mãe diretamente por meio do objeto Computador, refletindo a forte dependência entre essas duas entidades.

- a) Faça o Diagrama de Classe (lucid.app)
- b) Implemente o Código-Fonte em Java
- c) Relacione e explique os conceitos de acordo com seu código

4) Considere um sistema de Gerenciamento de Hospitais, cuja modelagem orientada a objetos reflete diferentes tipos de relacionamentos entre as entidades envolvidas. Um Hospital é composto por diversas Unidades de Atendimento (como UPA, Pronto Socorro, Enfermaria, entre outras), caracterizando uma relação de composição, pois tais unidades não existem de forma independente fora do hospital. Cada Unidade agrega múltiplos Médicos, configurando uma relação de agregação, já que os médicos podem atuar em mais de uma unidade ou até em instituições diferentes. Por fim, a relação entre Médico e Paciente ocorre por meio das Consultas, caracterizando uma associação, pois ambas as entidades são independentes, mas se relacionam durante o atendimento clínico. Essa estrutura facilita a representação precisa das dependências e interações dentro do contexto hospitalar.

- a) Faça o Diagrama de Classe (lucid.app)
- d) Implemente o Código-Fonte em Java
- e) Relacione e explique os conceitos de acordo com seu código