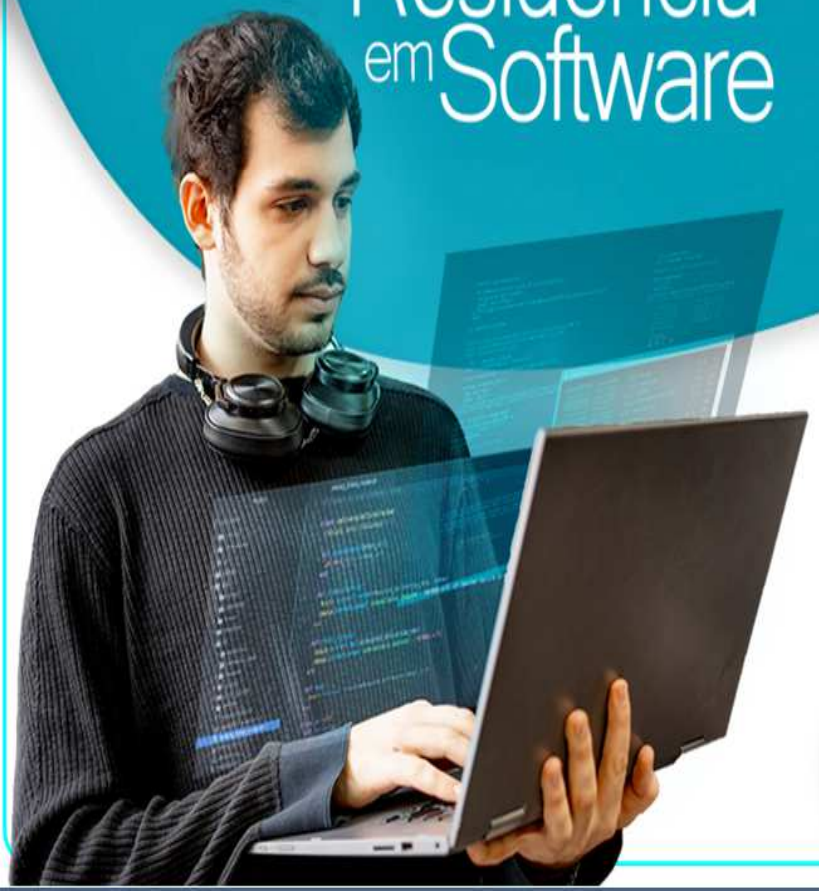
Residência em Software

Residência em Tecnologia da Informação e Comunicação

Primncípios de Programação Orientada a Objetos



Professor:

Alvaro Degas Coelho

Java

INSTITUIÇÃO EXECUTORA

cepedi UESC

COORDENADORA

MCTI FUTURO Softex

APOIO

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Casos de Uso

- Ator
  - Ente (provavelmente humano) que interage com o seu sistema
  - Exemplos
    - O caixa do supermercado. O atendente do hospital. O vendedor (registrando uma venda). O fiscal dos horários da empresa de ônibus. O enfermeiro cadastrando resultados de exames...

Atores podem não serem pessoas

- Ator é qualquer ente que interage: humano é uma possibilidade
  - Exemplo: um sistema de Vendas conversa com um sistema de Controle de Estoque
  - Do ponto de vista do Controle de Estoque, o sistema de vendas é um ator
    - Qual o estoque disponível do produto X?
    - Reduza em N unidades o estoque do produto X
  - E vice versa

Caso de Uso

- "documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo“
  - Ivan Jacobson

Casos de Uso: porque

- Os casos de uso:
  - Descrevem como os usuários interagem com o sistema (as funcionalidades do sistema)
- Dão uma visão externa do sistema
  - O que ele faz, e como

Padrão Expert

- O “amigo” mais íntimo do Caso de Uso
  - E do programador em OO
- Ideia do Padrão Expert
  - Dê a responsabilidade de cumprir uma tarefa para a Classe especialista nela
- Classe “Especialista”?
  - Dona dos dados.
  - Agrega quem é dono dos dados
  - Conversa com quem vai fazer subtarefas
  - ... com a experiência vem a sabedoria!

Um exemplo inicial

- Solicitar matrícula de um estudante (José) numa turma (T666 da disciplina X66-Trabalhos Forçados)
- Início
  - Aluno -> Dados de Login -> Sistema
  - Aluno <- Confirmação Login <- Sistema
  - Aluno -> Seleciona Opcao Matricular -> Sistema
  - Aluno <- Tela de Matrícula <- Sistema
  - Aluno -> solicitaMatricula(Aluno, T666) -> Sistema
  - Aluno <- confirmaSolicitacao() <- Sistema

Observações Importantes

- Ainda não estamos selecionando que classe vai recepcionar cada uma das solicitações do Estudante
  - Usaremos o Padrão Expert
- Não estamos preocupados com situações de erro
  - José pode ter errado sua senha, enviado uma disciplina inexistente, ou não ter o pré-requisito, etc.

Um pequeno Exercício

- Elabore casos de Uso para
  - Abrir uma nova turma para uma dada disciplina, com um certo número de vagas
- Cadastrar professor para alguma turma
- Pode omitir a parte de autenticação (login) e de tratamento de exceções
- Não se esqueça: por enquanto as solicitações são direcionadas ao “Sistema”. Depois debulharemos isso

Voltando ao exemplo do Tribunal

- Vejamos que “Num tribunal há **processos**. São ações movidas por alguém que tem algum interesse, e é dirigida contra alguém que, portanto, também tem interesse.”
- Há algum caso de uso aqui?

Criar Processo

- Caso de Uso Criar (Iniciar?) Processo
- Desprezaremos por enquanto a etapa de login (autenticação)
  - Quem seria o ator?
- Ator -> solicitaCriarProcesso() -> Sistema
- Ator <- telaProcesso() <- Sistema
- Ator -> criarProcesso(partes, dadosProcesso) -> Sistema
- Ator <- confirmaçãoProcessoCriado() <- Sistema

E as Classes envolvidas

- Lembre do Padrão Expert
  - E do método das classes de palavras, se for o caso
- Quem seria a classe responsável por executar *criarProcesso(partes, dadosProcesso)* ?

Residência em Software

E as Classes envolvidas

- Lembre do Padrão Expert
  - E do método das classes de palavras, se achar necessário
- Quem seria a classe responsável por executar *criarProcesso(partes, dadosProcesso)* ?
- Ator -> criarProcesso(partes, dadosProcesso) -> PROCESSO

Residência em Software

Prosseguindo

- “Qualquer pessoa (física ou jurídica) que tem interesse em um processo é chamadas de **Parte**. Tipicamente as partes são uma pessoa de cada lado. Mas pode haver grupos de pessoas em qualquer Parte.”
- Há um caso de uso aqui?
  - Parte é uma pessoa ou um grupo de pessoa
    - Conjunto (Coleção) possivelmente unitária de pessoas?
  - Como um usuário poderia criar uma Parte?

Caso de Uso: criar (definir?) Parte

- Ator -> criarNovaParte() -> Sistema
- Ator <- idParte <- Sistema
- Ator -> solicitaListaPessoas() -> Sistema
- Ator <- listaDePessoas <- Sistema
- Ator -> incluirPessoaNaParte(pessoa, parte) -> Sistema
- Ator <- dadosParte <- Sistema
- Ator -> registrarParte(parte) -> Sistema
- Ator <- confirmacaoParteRegistrada() <- Sistema

E as classes

- Que classe responde por
  - criarNovaParte()
  - solicitaListaPessoas()
  - incluirPessoaNaParte(pessoa, parte)
  - registrarParte(parte)

Caso de Uso: criar (definir?) Parte

- Ator -> criarNovaParte() -> PARTE
- Ator <- idParte <- PARTE
- Ator -> solicitaListaPessoas() -> PESSOA
- Ator <- listaDePessoas <- PESSOA
- Ator -> incluirPessoaNaParte(pessoa, parte) -> PARTE
- Ator <- dadosParte <- PARTE
- Ator -> registrarParte(parte) -> PARTE
- Ator <- confirmacaoParteRegistrada() <- PARTE

Outro Caso de Uso

- “Há casos especiais em que a Parte é o **Estado**: uma autarquia, uma empresa pública, um órgão da administração direta (prefeitura, governadoria, presidência). Estes processos tem características especiais porque tramitam sob vigilância de algum órgão público de **fiscalização**.”

Erro comum: pecado da falsa onisciência

- Provavelmente você já ficou quebrando a cabeça tentando criar um jeito de seu programa “saber” coisas para facilitar a sua execução
  - Isto é bom
- Mas normalmente a responsabilidade de “saber tudo” não recai sobre o sistema
  - Assumir isso sem necessidade é ruim para cronograma, prazos e, principalmente, boa programação

Quem “sabe” se o processo é fiscalizado externamente?

- Provavelmente não é responsabilidade do sistema “saber” isso
  - Assumiremos, aqui, que não é
- Mas é responsabilidade do sistema **REGISTRAR** isso
  - Saber se o processo é fiscalizado
  - Saber por quem o processo é fiscalizado

Voltando ao trecho

- “Há casos especiais em que a Parte é o **Estado**: uma autarquia, uma empresa pública, um órgão da administração direta (prefeitura, governadoria, presidência). Estes processos tem características especiais porque tramitam sob vigilância de algum órgão público de **fiscalização**.”
- Há casos de uso aqui?
- Fiscalização é uma classe? O órgão público que fiscaliza é uma classe?
  - Isto nos remete a Casos de Uso “não escritos?”

Exercício

- Definir o caso de Uso
  - CriarOrgaoFiscalização
- Voltar ao caso de Uso de cadastro de processo
  - Permitir que um processo seja identificado como “fiscalizado”, e seu órgão fiscalizador esteja definido

Para agora e Para casa

- Para agora
  - Observe as classes que já definimos nos casos de uso
    - Construa o diagrama de classes que as inclua até o momento
- Para casa
  - Prossiga com os demais casos de uso do sistema do Tribunal
- Para se divertir: estude mais sobre casos de uso, descrição de casos de uso, diagramas de clasos de uso e, Forte recomendação...
  - DIAGRAMAS DE SEQUÊNCIA