**Engenharia de Software**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Digitado por:** | **Professor:** | **Versão** | **Ano** |
| Ricardo Kim | Aluizio Saiter | 1.0 | 2017 |

**ÍNDICE**

[1. Introdução 3](#_Toc483177250)

[2. Requisitos 4](#_Toc483177251)

[2.1. Análises 4](#_Toc483177252)

[2.2. Ator (Entidades) 4](#_Toc483177253)

[2.3. Descrevendo os casos de uso 4](#_Toc483177254)

[3. Processos 5](#_Toc483177255)

[4. Observações em Aula 7](#_Toc483177256)

[5. Representação de classes 8](#_Toc483177257)

[5.1. Lei de Demeter 8](#_Toc483177258)

[5.2. Representação 9](#_Toc483177259)

# Introdução

* Normas IEEE
  + Requisito
  + Modulação
  + Teste
  + Formal
  + Method
  + Estimative
* Processo
* Requisitos
* Modulação
* Teste

# Requisitos

## Análises

* Funcionalidades (Obsoleto)
  + Análise
  + Estrutura
  + SADT
  + SA
* Objetivos
  + **Casos de uso**
  + Objeto interação com usuário
* A/D
* Compilação
* Teste
* Uso

## Ator (Entidades)

* Pessoas
* Sistema
* Sensores
  + Solicitante
  + Aprovador

Ator papel ao qual usuário interage com o sistema.

1. Requisitos = Ator + Casos de uso
2. Diagrama dos casos de uso
3. Específicação do User Case

Representação do **Caso de uso**

Ex: Caixa 24

* Caso de uso
  + Sacar
  + Consultar
* Ação dos “atores”
  + Correntista
    - Sacar
    - Consultar
  + Sistema do Caixa
  + Sistema da conta corrente

## Descrevendo os casos de uso

* Caixa Branca 🡪 Detalhes
* Caixa Cinza 🡪 Regras de negócio
* Caixa Preta 🡪

# Processos

* Empresa executa processo
  + Automatizar processos
    - Ganhos
      * Qualidade
      * Velocidade
      * Economia
* Desenvolver sistema
  + Interrompe comunicações
  + Desenvolver somente processos bons
* Requisitos do processo
  + Caso de uso (representa objetivo)
* Especificações dos casos de uso
  + Nome 🡪 verbo infinitivo + substantivo (ex: Consultar Saldo)
  + Breve Descrição 🡪 Descrição resumida
  + Fluxo Básico 🡪 Tudo relacionado às solicitações mais comuns
    - 1
    - 2
    - N
  + Fluxo Alternativo 🡪 Todas às exceções (não tratar BUGs físicos)
    - 1
    - 2
    - N...
  + Condições especiais
  + Pré condição
  + Pós Condição

Exemplos:

* Fluxo Básico
  + 1
    - Correntista
      * Insere cartão
    - Sistema caixa 24h
      * Lê cartão
      * Verifica se o cartão pertence ao pool de bancos
      * Verifica se data de validade do cartão está expirado
      * Solicita senha
  + 2
    - Correntista
      * Informa senha
    - Sistema caixa 24h
      * Verifica se senha está correta
      * Devolve cartão
      * Solicita operação desejada
  + 3
    - Correntista
      * Escolhe operação de consultar saldo
    - Sistema caixa 24h
      * Pergunta se o sistema do banco solicitado, se está online.
  + 4
    - Sistema Conta Corrente
      * Informa estar online
    - Sistema 24h
      * Envia dados do cliente e solicita saldo.
  + 5
    - Sistema Conta Corrente
      * Retorna o saldo
    - Sistema caixa 24h
      * Apresenta o salto para o correntista
      * Pergunta se correntista quer imprimir saldo
  + 6
    - Correntista
      * Solicita impimir slado
    - Sistema caixa 24 h
      * Imprime saldo.
      * Sistema pergunta se correntista deseja realizar outra operação.
  + 7
    - Correntista
      * Informa que não deseja realizar mais operações
    - Sistema caixa 24h
      * Retorna ao estado inicial.

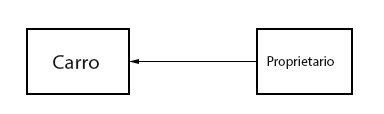
# Observações em Aula

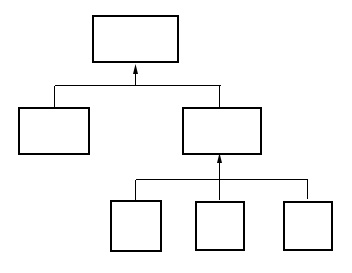
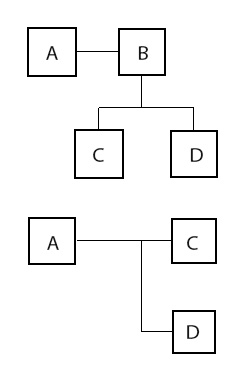
Não existe caso de uso que seja pré-requisito outro caso de uso

# Representação de classes

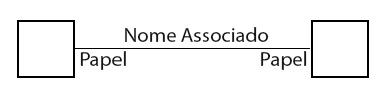
## Lei de Demeter

Objeto fala com ele próprio ou com quem ele tem relacionamento.

* 
  + Possui (mais básica) Reflexiva.

* https://i.stack.imgur.com/WiZCk.png
  + Parte de Composição
* https://i.stack.imgur.com/XkFft.png
  + Parte de Agregação
* 
  + Herança 🡨 Tudo que esta nos pais é herdado
  + Relacionamento, Atributos, Métodos.
* 
  + Classe virtual
  + Herda
    - Relacionamento
    - Atributos
    - Métodos
  + Cardinalidade quantidade de objetos possíveis de criar.

## Representação



# Principio básico do teste (comprometimento)

* Incompletude do teste
* Existência de falhas
  + Falhas não listadas podem significar que o teste é ineficiente
* Assumir o custo de parada?
  + tolerância a falhas?

Observações

* (gráfico em o²)
* Maior tempo de teste, maior custo, menor risco

Teste de caixa preta (especificação dos parâmetros de teste) (garantir a especificação):

Teste somente de especificação (necessário caso contrario não há teste de caixa preta).

* Caso de teste baseado na especificação
* Particiona mento de equivalência
* Analise de valor limite
* Tabela de Decisão e Ação.

Teste de caixa branca (análise do código) (normalmente é o primeiro) (não usar em teste de integração):

Teste somente quando eu tiver o código

* Grafo de fluxo de controle
  + Todos os nos
  + Todos os ramos
* Grafo de fluxo de dados
  + Todo
  + aldef
  + alluse
  + audeupef

Teste de sintaxe (técnica onde ingerir erros de sintaxes propositais (não deve passar erros de sintaxes))

* Erros de sintaxes não devem receber erros de sintaxe.
* Analise de mutantes
  + Quantidade de mutações de código
  + Programa de teste que gere as mutações

Caso de teste:

* Dados de entrada
* Dados de saída esperada

Classe de equivalência (modelo)

|  |  |  |
| --- | --- | --- |
| Variável | Classe de equivalências válidas | Classes de equivalência não válidas |
| X | Intervalo, ou conjunto de valores válidos para a classe de equivalência. | Intervalo, ou o conjunto de valores inválidos para a classe de equivalência. |
| Y | Intervalo, ou os conjuntos de valores válidos para a classe de equivalência. | Intervalor, um conjunto de valores inválidos para a classe de equivalência. |

Exercício 1 entrada de 1 a 10

Normal: entrada entre 1 a 10

Exceção 1: menor que 1

Exceção 2: maior que 10

Exceção 3: caractere

|  |  |  |
| --- | --- | --- |
| Variável | Classe de equivalências válidas | Classes de equivalência não válidas |
| X |  |  |

Exercício 2

Curso normal: o usuário informa uma cadeia de caracteres contento de 1 a 20 caracteres, e o caractere a ser procurado, o sistema informa a posição na cadeia em que o caractere foi encontrado pela primeira vez.

Exceção 1: O caractere informado não está presente na cadeia previamente informada

Exceção 2: A cadeia informada está vazia ou o seu tamanho é maior que 20

Exceção 3: não é informado um caractere a ser procurado ou informado a mais de um caractere.

|  |  |
| --- | --- |
| Classes de equivalências válidas | Classes de equivalências não validas. |
|  |  |

dados de entrada quebrar em grupos e testar mesma partição de entrada mesma partição de saida

# tabelas de condição de ação

[audio]

**Condições**

Condição 1: a = b

Condição 2: b = c

Condição 3: c = a

Condição 4: a + b > c

Condição 5: b + c > a

Condição 6: c + a > b

Ação 1: triangulo isósceles

Ação 2: triangulo equilátero

Ação 3: triangulo escaleno

Ação 4: não é triangulo

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a>b+c | V | F | F | F | F | F | F | F | F | F | F |
| b>c+a | - | V | F | F | F | F | F | F | F | F | F |
| c> a+b | - | F | V | F | F | F | F | F | F | F | F |
| a=b | - | - | - | V | V | V | V | F | F | F | F |
| b=c | - | - | - | V | V | F | F | V | V | F | F |
| c=a | - | - | - | V | F | V | F | V | F | V | F |
| Equi |  |  |  | x |  |  |  |  |  |  |  |
| Iso |  |  |  |  |  |  | x |  | x | x |  |
| Esc |  |  |  |  |  |  |  |  |  |  | x |
| nãoT | x | x | x |  |  |  |  |  |  |  |  |
| Impo |  |  |  |  | x | x |  | x |  |  |  |

[áudio gráfico de fluxo de controle]

Um nó possui código

Um ramo possui um fluxo de controle

Atividade:

Desenhe o grafo de fluxo de controle para o programa de classificação de triangulo

## Complexidade ciclo matica

Desenvolvido por Thomas MacCabe em 1980

Trata na quebra de um programa em função da complexidade

Quesito de legibilidade de código

é possivel usar em testes de caixa branca, usando nos caminhos base (básica)