

Programação Modular: Lista 1

Entrega no dia 16 de outubro de 2019 às 17h

Professor Flavio Bevilacqua

Antônio Vasconcellos Chaves

Engenharia da Computação
Pontifícia Universidade Católica
do Rio de Janeiro
Rio de Janeiro, RJ 22451-900
antoniovasconcelloschaves@gmail.com

João Pedro Paiva

Ciência da Computação
Pontifícia Universidade Católica
do Rio de Janeiro
Rio de Janeiro, RJ 22451-900
joaopedrordepaiva@gmail.com

Pedro Moreira Costa

Engenharia da Computação
Pontifícia Universidade Católica
do Rio de Janeiro
Rio de Janeiro, RJ 22451-900
pedromoreiramcosta@gmail.com

Questão 1

Explique com um exemplo o conceito de callback.

Solução

Callback é o termo usado para quando ao ter os dados requisitados, enviados pelo cliente, ainda há requisitos não preenchidos que não são obrigatórios. Por exemplo, em um cadastro, o servidor precisa das informações (nome, sobrenome e CPF) do cliente. O cliente esquece de preencher o CPF. Então o servidor volta/gera para interface com um aviso de que aquele campo é de preenchimento obrigatório.

Questão 2

Apresente um requisito funcional bem formulado, derivado de um requisito não funcional, diferente do exemplo de login visto em aula.

Solução

A necessidade que a solução seja descentralizada, por exemplo. Pois torna necessária a implementação de módulos para estabelecer a conexão entre os nós do sistema distribuído.

Questão 3

Um bom acoplamento resulta em um bom encapsulamento. Certo, errado, tipo assim, justifique.

Solução

Errado. Ambos estão relacionados mas não têm influência um sobre o outro. O acoplamento pode ser feito com diversos parâmetros por função e conectores complexos e, mesmo assim, manter a integridade da estrutura a ser protegida e ter fácil manutenção se tudo que está relacionado à parte protegida encontra-se no mesmo local.

Questão 4

Nem toda coesão funcional é lógica. Certo, errado, tipo assim (depende), justifique.

Solução

Errado. Como a coesão funcional implica que os elementos interdependem em torno de uma funcionalidade podemos afirmar que também interdependem em torno do conceito lógico associado a ela.

Questão 5

É possível existir mais de um módulo de definição para um módulo de implementação? Porque alguém faria isso?

Solução

Sim. Para alterar a visibilidade dos métodos e atributos, ou seja, das funções e variáveis presentes naquele módulo.

Questão 6

Faça modelo, exemplo e assertivas de uma matriz dinâmica tridimensional que armazena árvores binárias que referenciam listas de matrizes genéricas bidimensionais.

Solução

Figura 1: Modelo físico da matriz dinâmica tridimensional que armazena árvores binárias que referenciam listas de matrizes genéricas bidimensionais.

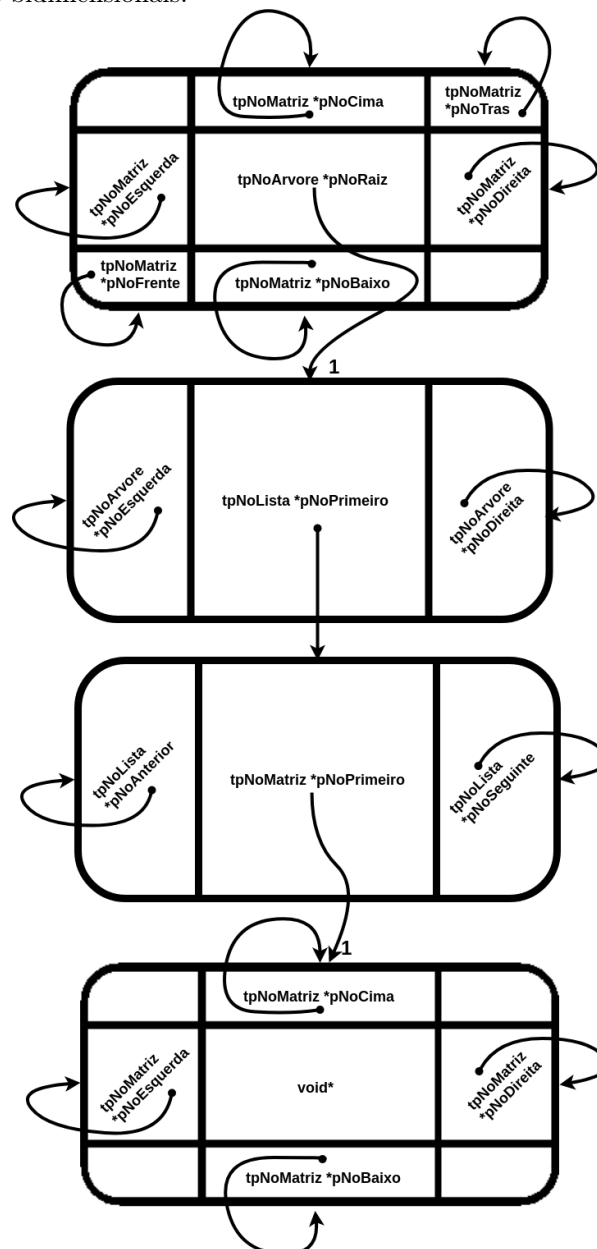
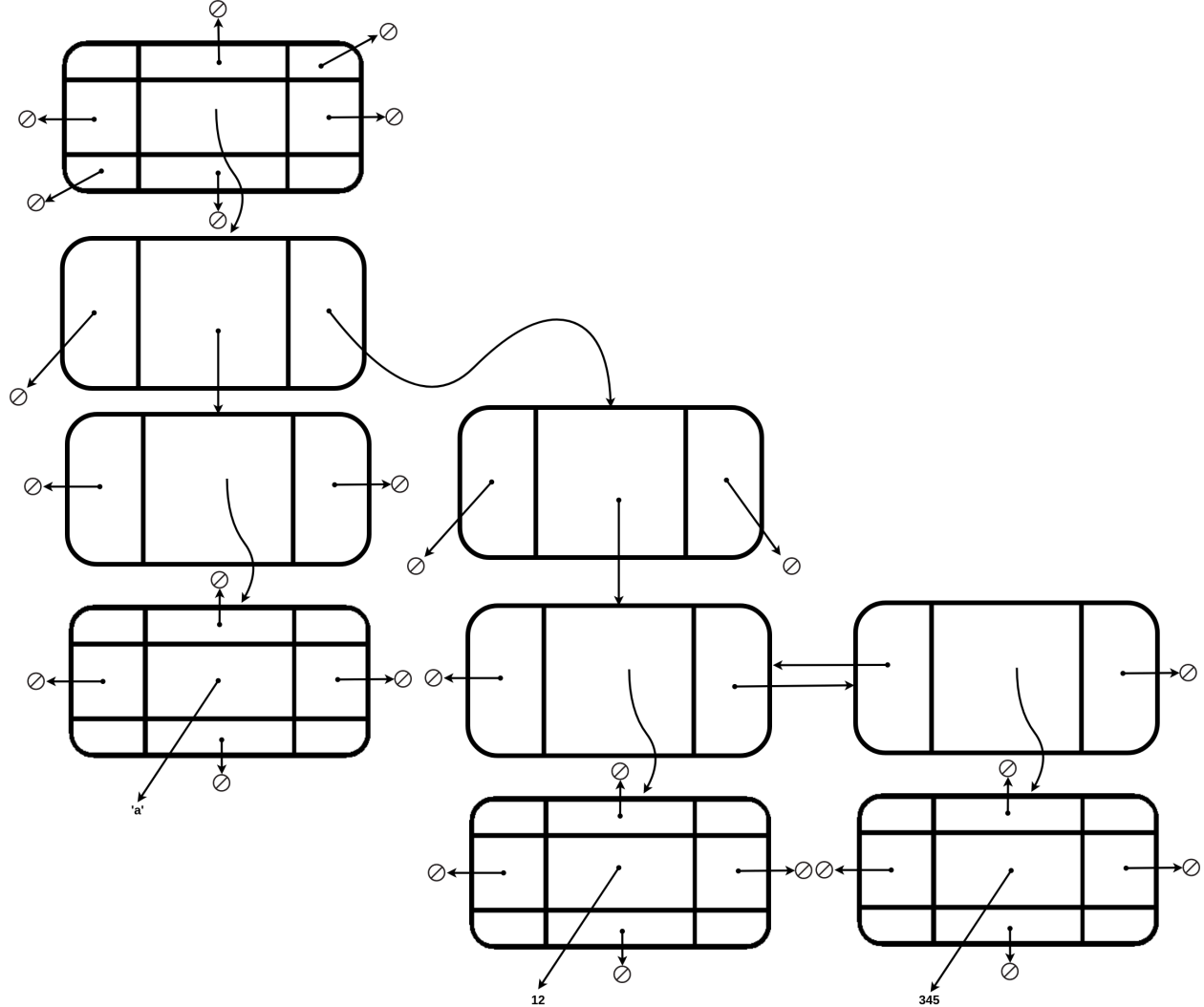


Figura 2: Exemplo físico da matriz dinâmica tridimensional que armazena árvores binárias que referenciam listas de matrizes genéricas bidimensionais.



- $\forall \text{ nó } pNo \in \text{matriz dinâmica tridimensional} : pNo \rightarrow pNoDireita \neq NULL \implies pNo \rightarrow pNoDireita \rightarrow pNoEsquerda == pNo$
- $\forall \text{ nó } pNo \in \text{matriz dinâmica tridimensional} : pNo \rightarrow pNoEsquerda \neq NULL \implies pNo \rightarrow pNoEsquerda \rightarrow pNoDireita == pNo$
- $\forall \text{ nó } pNo \in \text{matriz dinâmica tridimensional} : pNo \rightarrow pNoCima \neq NULL \implies pNo \rightarrow pNoCima \rightarrow pNoBaixo == pNo$
- $\forall \text{ nó } pNo \in \text{matriz dinâmica tridimensional} : pNo \rightarrow pNoBaixo \neq NULL \implies pNo \rightarrow pNoBaixo \rightarrow pNoCima == pNo$
- $\forall \text{ nó } pNo \in \text{matriz dinâmica tridimensional} : pNo \rightarrow pNoFrente \neq NULL \implies pNo \rightarrow pNoFrente \rightarrow pNoTras == pNo$
- $\forall \text{ nó } pNo \in \text{matriz dinâmica tridimensional} : pNo \rightarrow pNoTras \neq NULL \implies pNo \rightarrow pNoTras \rightarrow pNoFrente == pNo$

- \forall nó $pNo \in$ árvore binária : a referência para filho à esquerda de um nó tem como destino a raiz da sub-árvore à esquerda
- \forall nó $pNo \in$ árvore binária : a referência para filho à direita de um nó tem como destino a raiz da sub-árvore à direita
- \forall nó $pNo \in$ árvore binária : o conjunto de nós alcançáveis a partir da raiz da sub-árvore à esquerda é disjunto do conjunto de nós alcançáveis a partir da raiz da sub-árvore à direita
- \forall nó $pNo \in$ árvore binária : não existe caminho na árvore que leve de $pNo \rightarrow pNoEsquerda$ ou $pNo \rightarrow pNoDireita$ a pNo
- \forall nó $pNo \in$ lista : $pNo \rightarrow pNoDireita \neq NULL \implies pNo \rightarrow pNoDireita \rightarrow pNoEsquerda == pNo$
- \forall nó $pNo \in$ lista : $pNo \rightarrow pNoEsquerda \neq NULL \implies pNo \rightarrow pNoEsquerda \rightarrow pNoDireita == pNo$
- \forall nó $pNo \in$ matriz genérica bidimensional : $pNo \rightarrow pNoDireita \neq NULL \implies pNo \rightarrow pNoDireita \rightarrow pNoEsquerda == pNo$
- \forall nó $pNo \in$ matriz genérica bidimensional : $pNo \rightarrow pNoEsquerda \neq NULL \implies pNo \rightarrow pNoEsquerda \rightarrow pNoDireita == pNo$
- \forall nó $pNo \in$ matriz genérica bidimensional : $pNo \rightarrow pNoCima \neq NULL \implies pNo \rightarrow pNoCima \rightarrow pNoBaixo == pNo$
- \forall nó $pNo \in$ matriz genérica bidimensional : $pNo \rightarrow pNoBaixo \neq NULL \implies pNo \rightarrow pNoBaixo \rightarrow pNoCima == pNo$

Questão 7

Explique como é possível alterar o idioma utilizado nas telas de uma aplicação sem esse ter sido previsto ao longo do desenvolvimento e sem haver a necessidade de compilar e linkar novamente a aplicação.

Solução

Podemos acoplar todo texto que será exibido no programa em uma biblioteca dinâmica (DLL), que é ligada em tempo de execução. Assim, caso futuramente seja uma necessidade o uso de outro idioma, basta escrever uma nova biblioteca referente ao mesmo, sem que seja preciso recompilar a aplicação.

Questão 8

Dê exemplo de uma hipótese e explique em que isto auxilia no desenvolvimento de uma aplicação.

Solução

A validação de CPFs será feita no front-end da aplicação. Por esta hipótese sabe-se que não será necessário validar CPFs no back-end pois o mesmo foi feito no front-end.

Questão 9

É possível utilizar variáveis globais como itens de interface fornecidos por terceiros? Mostre como (caso sim).

Solução

Questão 10

Explique se existe diferença entre requisito inverso e restrição.

Solução

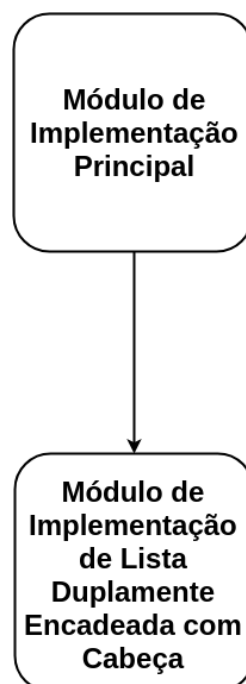
Existe sim diferença entre requisito inverso e restrição. Requisito inverso é aquilo que não será feito. Por exemplo, "Não implementaremos o login porque não foi pedido". Restrições são regras que restringem as alternativas de solução de um problema. Por exemplo, "A aplicação deve ser redigida em C++".

Questão 11

Um programa deve ser capaz de ler um documento de texto e criar um índice remissivo para cada substantivo encontrado. No índice remissivo é apresentada a lista de páginas e que a palavra é encontrada. Elabore a arquitetura modularizada deste programa (tal como foi vista na disciplina) considerando a criação de um tipo abstrato de dados para a estrutura principal a ser acoplada na aplicação. Neste tipo abstrato de dados deve ser utilizada a estrutura Lista Duplamente Encadeada com Cabeça.

Solução

Figura 3: Arquitetura modularizada do programa que lê um documento de texto e cria um índice remissivo para cada substantivo encontrado.



Questão 12

Uma estrutura de chamadas de funções pode ser simultaneamente recursiva direta, indireta e gerar dependência circular entre módulos. Comente essa afirmação.

Solução

Um arco de chamadas não pode ser simultaneamente recursivo direto e indireto, mas pode ser recursivo indireto com trechos de recursão direta. Dependência circular possui mais relação com os módulos dos quais funções são chamadas no arco de chamadas do que as próprias funções chamadas. Um arco recursivo indireto pode gerar dependência circular, quando pelo menos uma função de outro módulo é chamada entre chamadas de uma mesma função. Mas um arco recursivo direto não pode gerar dependência circular.

Questão 13

Dê um exemplo de função morta existente nos seus trabalhos. Justifique sua resposta.

Solução

Questão 14

Dado o seguinte requisito funcional: "Após o término da entrada das notas dos alunos na funcionalidade 'Apresentar Situação Final', a Média Final é calculada e apresentada na tela.". Avalie a necessidade da criação de um requisito inverso. Caso seja necessário, apresente este requisito e explique sua necessidade. Caso não seja necessário, explique a razão de não especificar este requisito.

Solução

Requisitos inversos são incluídos, geralmente, para eliminar ambiguidades. No requisito apresentado, não está claro qual Média Final deve ser calculada. Pode ser a Média Final de cada aluno ou de todos os alunos juntos. Também não está claro como a média deve ser calculada. Um requisito que poderia ser adicionado para esclarecer é: "esta aplicação não calcula a média de todos os alunos juntos, a média de cada aluno será calculada por $(G1 + G2 + 2 \cdot T1) / 4$ ".

Questão 15

Especifique as assertivas de entrada e saída de uma função que é armazenada no cabeça de uma lista para ser utilizada na exclusão do conteúdo apontado por um nó.

Solução

Questão 16

A validação de requisitos é feita pela equipe técnica junto com o cliente. Certo/Errado. Justifique.

Solução

Certo. Após elicitar do cliente aquilo que ele deseja, documentar a elicitação e verificar se aquilo que o cliente deseja é factível, o Analista de Requisitos valida os requisitos documentados com o cliente.

Questão 17

Explique o que um gerente de configuração de software faz em um processo de desenvolvimento.

Solução

O gerente de configuração de software maneja baselines e ambientes de configuração.

Questão 18

"O relatório deve ter no máximo dez linhas por página". Requisito funcional, não funcional, restrição ou hipótese? Justifique.

Solução

Requisito não funcional. Pois descreve algo que deve ser feito, mas não está relacionado a uma funcionalidade específica.

Questão 19

Mostre através de um exemplo como é possível com um único módulo de definição gerar interfaces personalizadas para cada módulo cliente.

Solução

O seguinte código de um módulo de definição:

```
#if defined (EXEMP_OWN)
    #define EXEMP_EXT

#else
    #define EXEMP_EXT extern

#endif

EXEMP_EXT int vetor[7]

#if defined (EXEMP_OWN)
    ={1,2,3,4,5,6,7}

#else
    ;

#endif
```

gera

```
int vetor[7]={1,2,3,4,5,6,7}
```

com este trecho presente no módulo cliente:

```
#define EXEMP_OWN

#include "M1.h"
#undef EXEMP_OWN
```

e gera

```
extern int vetor[7];
```

com o seguinte trecho presente em outro módulo cliente:


```
|| #include "M1.h"
```

Questão 20

Um valor somente declarado pode estar também definido? Certo/errado. Justifique.

Solução

****NUM MESMO MÓDULO!!****

Questão 21

É possível declarar sem definir? Porque alguém faria isso?

Solução

Sim. Para fazer referência a uma variável definida externamente, em outro módulo, que será ligada em tempo de linkedição, caso se trate de uma biblioteca estática, ou de execução, tratando-se de uma biblioteca dinâmica.

Questão 22

O encapsulamento sempre protege os dados. Certo/errado. Justifique.

Solução

Errado. Por mais que o encapsulamento tenha sido perfeito, ainda pode-se acessar o espaço de memória de algum dado e mexer nele, já que C é uma linguagem de baixo nível.