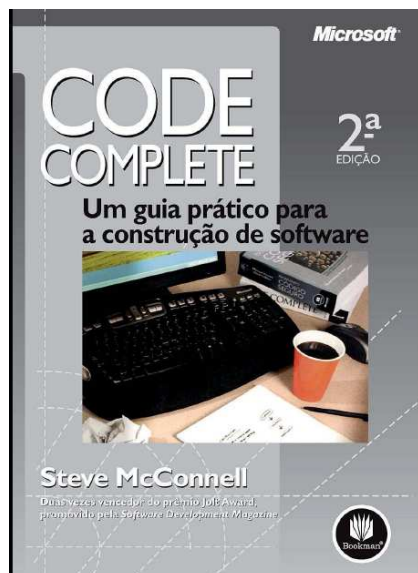


# Construção de Software

## Variáveis

Prof. Rubens de Castro Pereira, Me.  
rubens@inf.ufg.br



**Code Complete**  
**Um guia prático para a**  
**construção de software,**  
**2ª ed., Steven**  
**McConnell**  
**Bookman**

**Cap. 10 – Problemas**  
**Gerais no uso de**  
**Variáveis, pág. 263 a**  
**284**

## Variáveis

### Teste da capacidade de ler e escrever dados:

Avalie e pontue cada um dos termos do próximo com os seguintes valores:

- 0: não conheça o termo
- 0,5: conhece o termo, porém não possui certeza sobre o significado
- 1: conhece o termo e seu significado

Ao final, some os pontos e interprete conforme a tabela após o teste.

Prof. Rubens de Castro Pereira

3

## Variáveis

<input type="text"/> tipo abstrato de dados	<input type="text"/> literal
<input type="text"/> <i>array</i>	<input type="text"/> variável local
<input type="text"/> <i>bitmap</i>	<input type="text"/> tabela de pesquisa
<input type="text"/> variável booleana	<input type="text"/> dados-membro
<input type="text"/> árvore B	<input type="text"/> ponteiro
<input type="text"/> variável de caractere	<input type="text"/> privado
<input type="text"/> classe contêiner	<input type="text"/> sinapse retroativa
<input type="text"/> precisão dupla	<input type="text"/> integridade referencial
<input type="text"/> fluxo alongado	<input type="text"/> pilha
<input type="text"/> tipo enumerado	<input type="text"/> <i>string</i>
<input type="text"/> ponto flutuante	<input type="text"/> variável estruturada
<input type="text"/> <i>heap</i>	<input type="text"/> árvore
<input type="text"/> índice	<input type="text"/> <i>typedef</i>
<input type="text"/> inteiro	<input type="text"/> união
<input type="text"/> lista encadeada	<input type="text"/> encadeamento de valor
<input type="text"/> constante nomeada	<input type="text"/> <i>variant</i>
	<input type="text"/> Total de pontos

4

## Variáveis

- |       |                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-14  | Você é um programador iniciante, provavelmente em seu primeiro ano de ciência da computação na escola ou aprendendo sozinho sua primeira linguagem de programação. Poderá aprender muito lendo um dos livros mencionados na próxima subseção. Muitas descrições de técnicas desta parte do livro se destinam a programadores com conhecimentos avançados; você irá tirar mais proveito delas depois de ter lido um desses livros. |
| 15-19 | Você é um programador intermediário ou um programador experiente que já se esqueceu de muita coisa. Embora grande parte dos conceitos lhe sejam familiares, você também pode tirar proveito da leitura de um dos livros sugeridos a seguir.                                                                                                                                                                                       |
| 20-24 | Você é um especialista em programação. Provavelmente já possui em sua estante os livros sugeridos a seguir.                                                                                                                                                                                                                                                                                                                       |
| 25-29 | Você sabe mais sobre tipos de dados do que eu. Considere a possibilidade de escrever seu próprio livro de computação. (Envie-me um exemplar!)                                                                                                                                                                                                                                                                                     |
| 30-32 | Você é um tremendo trapaceiro. Os termos "fluxo alongado", "sinapse retroativa" e "encadeamento de valor" não se referem a tipos de dados – eu os inventei. Leia a seção "Honestidade intelectual", no Capítulo 33, "Caráter pessoal"!                                                                                                                                                                                            |

Prof. Rubens de Castro Pereira

5

## Variáveis

- Declarações implícitas de variáveis: NÃO UTILIZE!
- Declare todas as variáveis, mesmo que o compilador não exija.
- Use convenções de atribuição de nomes.
- Verifique os nomes das variáveis, evitando declarar variáveis e não as utilizar.

Prof. Rubens de Castro Pereira

6

## Diretrizes para Iniciar Variáveis

- Problemas relacionados a iniciar incorretamente:
  - A variável nunca recebeu um valor
  - O valor da variável está atualizado
  - Parte da variável recebeu um valor e parte não – inicializou alguns atributos (membros) de um objeto e outros não

Prof. Rubens de Castro Pereira

7

## Diretrizes para Iniciar Variáveis

- Inicie cada variável quando ela for declarada.
- Inicie cada variável perto de onde ela é usada pela primeira vez.
- Declare e defina cada variável perto de onde ela é usada pela primeira vez.
- Quando possível, defina constates (final ou const).
- Preste atenção especial em contadores e acumuladores (reinício)
- Inicie os dados-membros de uma classe no construtor.
- Verifique a necessidade de reiniciar as variáveis.
- Inicie constantes uma vez; inicie variáveis com código executável, próximo de onde elas são usadas.
- Use a configuração do compilador que inicia automaticamente todas as variáveis.
- Observe as mensagens de alerta de seu compilador.
- Verifique a validade dos parâmetros de entrada.
- Use um verificador de acesso à memória para procurar ponteiros (quando possível).
- Inicie a memória de trabalho no início do programa ou classe.

Prof. Rubens de Castro Pereira

8

## Diretrizes para Iniciar Variáveis

### Exemplos

```
long matriculaAluno = 0;
bool acaoSelecionada = false;
double valorMensalidade = 0;
float codigoProduto;
...

codigoProduto = -1;
while(!acaoSelecionada) {
    ...
}
```

Prof. Rubens de Castro Pereira

9

## Escopo ou Visibilidade de Variáveis

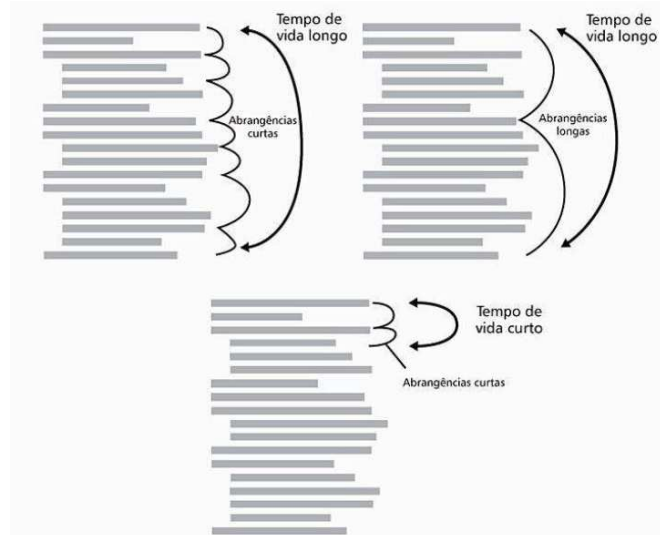
- Escopo ou Visibilidade refere-se ao alcance de uma variável ser conhecida e referenciada por todo o programa ou classe.
- Escopo limitado ou pequeno é conhecido apenas em uma pequena área código fonte.
- Escopo grande é conhecido em muitos lugares do código fonte.

Prof. Rubens de Castro Pereira

10

## Escopo ou Visibilidade de Variáveis

Mantenha  
as variáveis  
“vivas”  
pelo tempo  
mais curto  
possível



**Figura 10-1** “Tempo de vida longo” significa que uma variável está ativa durante o transcorrer de muitas instruções. “Tempo de vida curto” significa que ela está ativa por apenas algumas poucas instruções. “Abrangência” se refere a quanto as referências para uma variável estão próximas.

## Diretrizes Gerais para Minimizar o Escopo de Variáveis

- Inicie as variáveis usadas em um loop imediatamente antes do loop, em vez de iniciá-las no início da rotina que o contém.
- Não atribua valor a uma variável antes que chegue o momento exato em que esse valor será usado.
- Agrupe instruções relacionadas.
- Divida os grupos de instruções relacionadas em rotinas separadas.
- Comece com visibilidade mais restrita e expanda o escopo da variável somente caso seja necessário.

## Diretrizes Gerais para Minimizar o Escopo de Variáveis

### Exemplos

```
int indiceAlunos = 0;
float receitaDiaria = ReceitaDiaria();
double totalReceita = TotalReceita (receitaDiaria);
```

Prof. Rubens de Castro Pereira

13

Exemplos

Grupos de instruções

seção 14.2, "Instruções cuja ordem não importa".

Instruções usando dois conjuntos de variáveis.

#### Exemplo em C++ do uso desordenado de dois conjuntos de variáveis

```
void SummarizeData(...) {
    ...
    GetOldData( oldData, &numOldData );
    GetNewData( newData, &numNewData );
    totalOldData = Sum( oldData, numOldData );
    totalNewData = Sum( newData, numNewData );
    PrintOldDataSummary( oldData, totalOldData, numOldData );
    PrintNewDataSummary( newData, totalNewData, numNewData );
    SaveOldDataSummary( totalOldData, numOldData );
    SaveNewDataSummary( totalNewData, numNewData );
    ...
}
```

Instruções usando oldData.

Instruções usando newData.

#### Exemplo em C++ do uso mais inteligível de dois conjuntos de variáveis

```
void SummarizeData(...) {
    GetOldData( oldData, &numOldData );
    totalOldData = Sum( oldData, numOldData );
    PrintOldDataSummary( oldData, totalOldData, numOldData );
    SaveOldDataSummary( totalOldData, numOldData );
    ...
    GetNewData( newData, &numNewData );
    totalNewData = Sum( newData, numNewData );
    PrintNewDataSummary( newData, totalNewData, numNewData );
    SaveNewDataSummary( totalNewData, numNewData );
    ...
}
```

## Persistência de Variáveis

- Persistência significa o tempo de vida dos dados:
  - Persiste durante a vida de um determinado bloco de código ou rotina.
  - Persiste enquanto se permite, isto é, até serem eliminadas formalmente.
  - Persiste durante toda a vida de um programa ou classe (globais, *static*).
  - Persiste para sempre: armazenadas em um banco de dados ou arquivo.

Prof. Rubens de Castro Pereira

15

## Persistência de Variáveis

- Verifique o valor das variáveis antes de utilizá-las, gerando alertas se necessário.
- Após o uso das variáveis, configure-as com valores “despropositados”. Ex.: configurar um ponteiro como nulo após sua exclusão.
- Escreva código que presuma que os dados não são persistentes.
- Habitue-se a declarar e iniciar todos os dados imediatamente antes de serem usados.

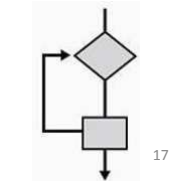
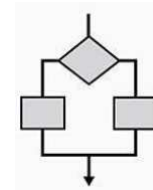
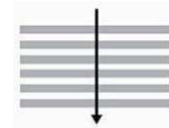
Prof. Rubens de Castro Pereira

16



## Relacionamento entre as Variáveis e Estruturas de Controle

- Dados sequenciais leva a instruções sequenciais em um programa/classe.
- Dados seletivos se transformam em instruções *if* e *case* em um programa/classe.
- Dados iterativos se transformam em estruturas e repetição *for*, *repeat* e *while*.



Prof. Rubens de Castro Pereira

17

## Usando cada Variável para Exatamente um Propósito

- Use cada variável para apenas um propósito.
- Evite variáveis com significados ocultos
- Certifique-se de que todas as variáveis declaradas sejam usadas.

Prof. Rubens de Castro Pereira

18

## Variáveis

### **Leitura extra aula:**

1. Code Complete – Um guia prático para a construção de software, 2ª ed., Steven McConnell, Bookman

### Parte III – Variáveis:

- Cap. 10 – Problemas Gerais no uso de Variáveis, pág. 263 a 284