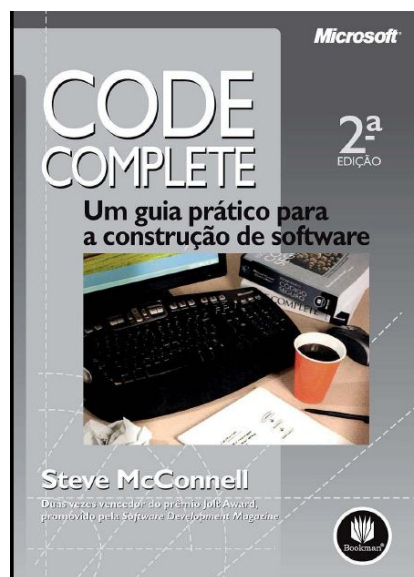


# Construção de Software

## 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. 12 – Tipos de**  
**Dados Fundamentais,**  
**pág. 315 a 344**

## Números em Geral

- Evite os números mágicos, substituindo-os por constantes
- Os únicos valores que podem ocorrer no código são 0 e 1
- Antecipe os erros de divisão por zero
- Torne óbvias as conversões de tipo de dados
- Evite comparações de variáveis com tipo de dados misto
- Preste atenção nos alertas gerados pela compilação do código

Prof. Rubens de Castro Pereira

3

## Números Inteiros

- Verifique a divisão de inteiros:  $7/10$  não é 0,7
- Verifique o estouro de inteiros
- Verifique o estouro de resultados intermediários

Prof. Rubens de Castro Pereira

4

## Números Inteiros

**Tabela 12-1** Intervalos de diferentes tipos de inteiros

Tipo de inteiro	Intervalo
8 bits com sinal	-128 a 127
8 bits sem sinal	0 a 255
16 bits com sinal	-32.768 a 32.767
16 bits sem sinal	0 a 65.535
32 bits com sinal	-2.147.483.648 a 2.147.483.647
32 bits sem sinal	0 a 4.294.967.295
64 bits com sinal	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
64 bits sem sinal	0 a 18.446.744.073.709.551.615

### Exemplo em Java de estouro de resultados intermediários

```
int termA = 1000000;
int termB = 1000000;
int product = termA * termB / 1000000;
System.out.println( "( " + termA + " * " + termB + " ) / 1000000 = " + product );
```

Prof. Rubens de Castro Pereira

5

## Números em Ponto Flutuante

- Evite adições e subtrações com números que tenham grandezas muito diferentes:  
1.000.000 + 0,1 provavelmente será 1.000.000
- Evite comparações de igualdade, usando comparação por intervalos
- Antecipe os erros de arredondamento
- Verifique o suporte da linguagem e da biblioteca para tipos de dados específicos

Prof. Rubens de Castro Pereira

6

## Caracteres e *Strings*

- Evite caracteres e strings mágicos, substituindo-os por constantes
- Cuidado com os erros de valor limite do índice em arrays
- Saiba como a linguagem e seu ambiente suportam Unicode
- Decida-se sobre uma estratégia de internacionalização/localização no início do tempo de vida de um programa
- Se houverão vários idiomas, use Unicode

Prof. Rubens de Castro Pereira

7

## *Strings* em C

- Identifique a diferença entre ponteiros e array de caracteres por meio do uso (sinal de igualdade ou convenção de nomes)
- Declare *strings* do estilo C de forma a terem um comprimento igual a `CONSTANT + 1`
- Inicie as *strings* como nulas para evitar *strings* infinitas (`\0` é o delimitador de fim)
- Use array de caracteres em vez de ponteiros
- Use `strncpy()` em vez de `strcpy()` para evitar strings infinitas (ausência do delimitador nulo)

Prof. Rubens de Castro Pereira

8

## Strings em C

**Exemplo em C de boas declarações de *string***

A *string* é declarada como tendo comprimento `NAME_LENGTH + 1`.

As operações sobre a *string* usando `NAME_LENGTH` estão aqui...

...e aqui.

```

/* Declara a string com um comprimento de "constant+1".
   Em todos os outros locais no programa, é usado "constant"
   em vez de "constant+1". */
char name[ NAME_LENGTH + 1 ] = { 0 }; /* string de comprimento NAME_LENGTH */

...

/* Exemplo 1: configura a string com todos os elementos sendo 'A', usando a
   constante NAME_LENGTH como o número de letras 'A' que podem ser copiadas.
   Note que NAME_LENGTH é usado, em vez de NAME_LENGTH + 1. */
for ( i = 0; i < NAME_LENGTH; i++ )
    name[ i ] = 'A';
...

/* Exemplo 2: copia outra string na primeira usando
   a constante como o comprimento máximo que pode ser copiado. */
strncpy( name, some_other_name, NAME_LENGTH );

```

Prof. Rubens de Castro Pereira

9

## Variáveis Booleanas

- Use variáveis booleanas para documentar seu programa/classe
- Use variáveis booleanas para simplificar testes complicados

### Exemplo em Java de teste booleano no qual o objetivo não está claro

```

if ( ( elementIndex < 0 ) || ( MAX_ELEMENTS < elementIndex ) ||
    ( elementIndex == lastElementIndex )
    ) {
    ...
}

```

### Exemplo em Java de teste booleano no qual o objetivo está claro

```

finished = ( ( elementIndex < 0 ) || ( MAX_ELEMENTS < elementIndex ) );
repeatedEntry = ( elementIndex == lastElementIndex );
if ( finished || repeatedEntry ) {
    ...
}

```

## Tipos Enumerados

- Permite a cada membro de uma classe de objetos ser descrito por um nome
- Use tipos enumerados para dar legibilidade, segurança e facilidade de modificação
- Use tipos enumerados como uma alternativa às variáveis booleanas
- Verifique a existência de valores inválidos
- Defina a primeira e a última entrada de uma enumeração para usar como limites de um loop
- Reserve a primeira entrada do tipo enumerado como sendo inválida (valor 0)
- Defina precisamente como os elementos *first (primeiro)* e *last (ultimo)* devem ser usados no padrão de codificação do projeto e utilize-os consistentemente
- Cuidado com as armadilhas da atribuição de valores explícitos aos elementos de uma enumeração

Prof. Rubens de Castro Pereira

11

## Tipos Enumerados

Exemplo na linguagem C/C++:

```
typedef enum {  
    INVALIDA                = 0 ,  
    INCLUIR                 = 1 ,  
    ALTERAR                 = 2 ,  
    EXCLUIR                 = 3 ,  
    CONSULTAR_ELEMENTO      = 4 ,  
    CONSULTAR_LISTA         = 5  
} Operacao;
```

Prof. Rubens de Castro Pereira

12

## Constantes Nomeadas

- Use constantes nomeadas em declarações de dados
- Evite literais, mesmo aquelas consideradas "seguras"
- Simule constantes com variáveis ou classes com escopo apropriado
- Use constantes consistentemente

Prof. Rubens de Castro Pereira

13

## Constantes Nomeadas

### Exemplo em Visual Basic de código ilegível

```
For i = 1 To 12
    profit( i ) = revenue( i ) - expense( i )
Next
```

### Exemplo em Visual Basic de código legível

```
For i = 1 To NUM_MONTHS_IN_YEAR
    profit( i ) = revenue( i ) - expense( i )
Next
```

### Exemplo em Visual Basic de código ainda mais legível

```
For month = 1 To NUM_MONTHS_IN_YEAR
    profit( month ) = revenue( month ) - expense( month )
Next
```

### Exemplo em Visual Basic de código muito legível

```
For month = Month_January To Month_December
    profit( month ) = revenue( month ) - expense( month )
Next
```

Prof. Rubens de Castro Pereira

14

# Arrays

- Certifique-se de que todos os índices do array estejam dentro dos limites do array
- Verifique as extremidades dos arrays
- Se um array for multidimensional, certifique-se de que seus subscritores (índices) sejam usados na ordem correta:

**notas[i][j] versus notas[j][i]**

- Cuidado com o cruzamento de índices  
**notas[j]** no lugar de **notas[i]**
- Em C, use a macro `ARRAY_LENGTH()` para trabalhar com arrays:

**Exemplo em C de definição de uma macro `ARRAY_LENGTH()`**

```
#define ARRAY_LENGTH( x ) (sizeof(x)/sizeof(x[0]))
```

**Exemplo em C de uso da macro `ARRAY_LENGTH()` para operações de array**

```
ConsistencyRatios[] =
{ 0.0, 0.0, 0.58, 0.90, 1.12,
  1.24, 1.32, 1.41, 1.45, 1.49,
  1.51, 1.48, 1.56, 1.57, 1.59 };
...
for ( ratioIdx = 0; ratioIdx < ARRAY_LENGTH( ConsistencyRatios ); ratioIdx++ );
...
15
```

É aqui que a macro é usada.

# Variáveis

## Leitura extra aula:

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

Parte III – Variáveis:

- Cap. 12 – Tipos de Dados Fundamentais, pág. 315 a 344