

Técnicas de Programação

Documentação de Código

Profa. Elaine Venson

elainevenson@unb.br

Conceitos

- Documentação externa x documentação no nível de código
- Nível de código
 - O código como documentação
 - Comentários no código

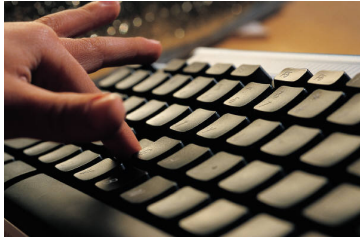
O Código como documentação

- Um bom **estilo de programação** é a forma mais eficiente de documentação em nível de código
 - Boa estrutura
 - Abordagens diretas e compreensíveis
 - Nomes de variáveis e rotinas significativos
 - Uso de constantes em vez de literais
 - Layout limpo
 - Minimização da complexidade das estruturas de dados e fluxos de controle

Frase

“Don’t document bad code - rewrite it”

(Kernighan e Plauger, 1978)



Técnicas para escrever código

- Objetivo: código legível e auto-explicativo

Documentar é preciso?



Documentação com comentários

- Repetição do código
 - Apenas mais linhas para ler
- Explicação do código
 - Código precisa ser melhorado
- Marcadores
 - Comentário que não deve ficar no código
 - Padronizar e revisar



Documentação com comentários

- **Resumo** do código

- Poucas linhas, uma ou duas sentenças
- Melhor que repetir o código
- Ajuda porque é mais rápido que ler o código

- **Descrição da intenção**

- Comentários no nível do problema em vez da solução
- Entender a intenção do programador no código original é a maior dificuldade para quem mantém código

Comentários ineficientes

- Comentar muito é tão ruim quando documentar pouco
- Quando é necessário muito esforço:
 - O estilo do comentário é ruim
 - O programador não entendeu adequadamente o que o programa faz

Comentários ineficientes

```
// Variable      Meaning
// -----
// xPos ..... XCoordinate Position (in meters)
// yPos ..... YCoordinate Position (in meters)
// ndsCmptng..... Needs Computing (= 0 if no computation is needed,
//                               = 1 if computation is needed)
// ptGrdTtl..... Point Grand Total
// ptValMax..... Point Value Maximum
// psblScrMax..... Possible Score Maximum
```

```
/******
 * class:  GigaTron (GIGATRON.CPP)                                *
 *                                              *
 * author: Dwight K. Coder                                *
 * date:   July 4, 2014                                    *
 *                                              *
 * Routines to control the twenty-first century's code evaluation *
 * tool. The entry point to these routines is the EvaluateCode()  *
 * routine at the bottom of this file.                        *
 *****/
```

Comentários efetivos

- Estilo de comentário
 - Definir estilo que seja fácil de manter
- Utilizar processo de programação com pseudo-código
- Integrar documentação ao processo de desenvolvimento
 - Deixar para depois aumenta o esforço
 - Comentários devem sair de forma natural
 - Dificuldade de comentar é um alerta

Numero ótimo de comentários

- Estudos realizados por Capers Jones na IBM indicaram que a densidade de **um comentário a cada 10 sentenças** parecia ser o número ótimo para clareza do código

Técnicas para documentar

- Comentar linhas individuais
- Comentar parágrafos de código
- Comentar declaração de variáveis
- Comentar estruturas de controle
- Comentar rotinas
- Comentar classes, arquivos e programas

Técnicas para documentar

- Comentar linhas individuais
- Comentar parágrafos de código
- Comentar declaração de variáveis
- Comentar estruturas de controle
- Comentar rotinas
- Comentar classes, arquivos e programas

Comentar linhas individuais

- Necessário em poucos casos
- Recomendável nas situações:
 - Um única linha extramente complicada
 - Registrar um erro já ocorrido em uma linha
 - Documentar declaração de variável

Comentar linhas individuais

- Problemas com comentários em final de linha
 - Quando o código da linha aumenta, o comentário fica deslocado
 - Tendência de apenas repetir o código
 - Se o comentário se aplicar a várias linhas de código, não representa a quais linhas se refere

Comentar linhas individuais

- Exemplo em Java de documentação de variáveis:

```
int boundary = 0;           // upper index of sorted part of array
String insertVal = BLANK; // data elmt to insert in sorted part of array
int insertPos = 0;          // position to insert elmt in sorted part of array
```

Técnicas para documentar

- Comentar linhas individuais
- Comentar parágrafos de código
- Comentar declaração de variáveis
- Comentar estruturas de controle
- Comentar rotinas
- Comentar classes, arquivos e programas

Comentar parágrafos de código

- Uma ou duas sentenças que documentam um grupo de linhas de código
- É uma das formas mais úteis de documentação com comentários

Comentar parágrafos de código

- Recomendações

- Escrever comentários sobre a **intenção** do código
 - Pensar como se fosse dar um nome de função para o conjunto de linhas
- Verificar se o parágrafo não deveria ser transformado em rotina
- Escrever comentário sobre o **por que** em vez de **como**
- Utilizar o comentário para preparar o leitor para o que vem a seguir
- Documentar conjunto de linhas quando for utilizado algum **artifício** por motivos de performance. Deixar claro o ganho obtido
- Evitar abreviações

Comentar parágrafos de código

- Recomendações (cont)

- Diferenciar comentários **principais** e **secundários**
 - Por exemplo, usar “. . .” antes do texto dos comentários secundários
- Comentar problemas em rotinas de biblioteca
 - Justificar as linhas com tratamento especial para o caso problemático
- Justificar violações do estilo de programação
- Não comentar código complicado, reescrever
 - Exceção para manutenção

Comentar parágrafos de código

- Um bom exemplo

```
// swap the roots  
oldRoot = root[0];  
root[0] = root[1];  
root[1] = oldRoot;
```

Exemplo de bom
comentário em Java:
apresenta a intenção

Técnicas para documentar

- Comentar linhas individuais
- Comentar parágrafos de código
- Comentar declaração de variáveis
- Comentar estruturas de controle
- Comentar rotinas
- Comentar classes, arquivos e programas

Comentar declaração de variáveis

- Descrever aspectos da variável que o nome não consegue representar
- Recomendações
 - Comentar unidade para variáveis numéricas
 - Comentar o intervalo permitido para valores numéricos
 - Comentar códigos ou usar enumeradores
 - Comentar limites dos valores de entrada
 - Ou usar assertivas
 - Documentar flags em nível de bit
 - Documentar variáveis globais

Comentar declaração de variáveis

- Exemplo em Visual Basic

```
Dim cursorX As Integer ' horizontal cursor position; ranges from 1..MaxCols
Dim cursorY As Integer ' vertical cursor position; ranges from 1..MaxRows

Dim antennaLength As Long ' length of antenna in meters; range is >= 2
Dim signalStrength As Integer ' strength of signal in kilowatts; range is >= 1

Dim characterCode As Integer ' ASCII character code; ranges from 0..255
Dim characterAttribute As Integer ' 0=Plain; 1=Italic; 2=Bold; 3=BoldItalic
Dim characterSize As Integer ' size of character in points; ranges from 4..127
```

Técnicas para documentar

- Comentar linhas individuais
- Comentar parágrafos de código
- Comentar declaração de variáveis
- Comentar estruturas de controle
- Comentar rotinas
- Comentar classes, arquivos e programas

Comentar estruturas de controle

- Para estruturas de decisão como **if** e **case**, pode-se comentar o motivo da decisão e um resumo do resultado obtido
- No caso de **loops**, pode ser indicado o seu propósito
- Recomendações:
 - Incluir o comentário antes da estrutura de controle
 - Comentar o final da estrutura de controle
 - Especialmente em loops longos ou estruturas aninhadas
 - Loops com comentários ao final podem indicar necessidade de melhorar o código

Comentar estruturas de controle

- Exemplo em C++

```
// copy input field up to comma
while ( ( *inputString != ',' ) && ( *inputString != END_OF_STRING ) ) {
    *field = *inputString;
    field++;
    inputString++;
} // while -- copy input field

*field = END_OF_STRING;

if ( *inputString != END_OF_STRING ) {
    // read past comma and subsequent blanks to get to the next input field
    inputString++;
    while ( ( *inputString == ' ' ) && ( *inputString != END_OF_STRING ) ) {
        inputString++;
    }
} // if -- at end of string
```

Técnicas para documentar

- Comentar linhas individuais
- Comentar parágrafos de código
- Comentar declaração de variáveis
- Comentar estruturas de controle
- Comentar rotinas
- Comentar classes, arquivos e programas

Comentar rotinas

- Recomendações:

- Manter os comentários próximos do código que descrevem
- Descrever cada rotina em uma ou duas sentenças no início da mesma
 - Dificuldades em descrever uma rotina podem indicar que o projeto não está bom
- Documentar parâmetros na declaração
- Utilizar ferramentas como o Javadoc
- Distinguir parâmetros de entrada e de saída
- Documentar suposições de interfaces
 - Uso do Assert

Comentar rotinas

- Recomendações (cont):
 - Comentar limitações da rotina
 - Documentar os efeitos globais da rotina
 - Documentar a origem dos algoritmos utilizados
 - Usar comentários para marcar partes do programa
 - `/**` -> marca início de rotinas em Java e C++
 - `@palavra-chave` -> classifica tipo específico de informação do comentário

Comentar rotinas

- Exemplo comentário no formato para Javadoc

```
/**
 * ... <description of the routine> ...
 *
 * @param dataToSort  elements to sort in locations firstElement..lastElement
 * @param firstElement index of first element to sort (>=0)
 * @param lastElement  index of last element to sort (<= MAX_ELEMENTS)
 */

public void InsertionSort(
    int[] dataToSort,
    int firstElement,
    int lastElement
)
```


Técnicas para documentar

- Comentar linhas individuais
- Comentar parágrafos de código
- Comentar declaração de variáveis
- Comentar estruturas de controle
- Comentar rotinas
- Comentar classes, arquivos e programas

Comentar arquivos, classes e programas

- Recomendações para **arquivos**:

- Descreva no início o propósito e conteúdo do arquivo
 - Se o arquivo contém mais de uma classe, explicar o motivo
 - Se houve divisão em múltiplos arquivos, explicar
- Incluir informações do autor: nome, e-mail, telefone
- Incluir tag de controle de versão
- Incluir avisos legais
 - Copyright, confidencialidade
- Dar ao arquivo um nome relacionado com o seu conteúdo

Comentar arquivos, classes e programas

- Recomendações para **classes**:

- Descrever a abordagem de projeto utilizada
- Descrever limitações, suposições de uso, etc
- Comentar a interface da classe
 - A interface deve conter as informações que alguém precisa saber para utilizar a classe
- Não documentar detalhes de implementação na interface

Comentar arquivos, classes e programas

- Recomendações para **programas**:
 - O **Paradigma do Livro**
 - Estudo publicado por **Paul Oman and Curtis Cook** em 1990
 - Analogia com um livro para organizar a documentação do programa
 - Prefacio: comentários introdutórios
 - Sumário: lista de classes, arquivos e rotinas (capítulos)
 - ...