Técnicas para Análise de Similaridade de Código de Software em Litígios de Propriedade Intelectual

PSI5007 - prof. Volnys Bernal

Ana Maria Mota (anamariamota_68@hotmail.com)

Denise Hideko Goya (dhgoya@ime.usp.br)

Questão Legal

- No Brasil:
  - Autoria de programas: Direito Autoral
  - Registro opcional junto ao INPI
    - Documentos do Programa
- Nos EUA:
  - Critérios de aceitação de prova:
    - Look and Feel
    - Abstraction, Filtration, Comparison
  - Software é patenteável

Similaridade de Código

- O que é similaridade de código?
- Definir similaridade para que propósito?
  - Comparação de programas
    - detectar roubo de propriedade intelectual
    - refatoração / manutenção / reengenharia
    - compactação de arquivos
  - Comparação de outros tipos de documentos
- O propósito influencia a definição e a técnica

Exemplo: Sintaxe Semelhante

```
int sum = 0 ;
void foo (Iterator iter){
   for ( item = first(iter); has more(iter); item = next(iter) ) {
    sum = sum + value ( item ) ;
  }
}
int bar (Iterator iter){
   int sum = 0 ;
  for ( item = first(iter); has more(iter); item = next(iter) ) {
     sum = sum + value ( item ) ;
   }
}
```

Exemplo: Sintaxe Diferente

```
while ((x = pi[t-1]) != '(' && x != '+' && x != '-') {
  posf[j++] = x;
  --t;
}
```

```
while (1) {
  x = pi[t-1];
  if (x == '(' || x == '+' || x == '-')
    break;
  --t;
  posf[j++] = x;
}
```

Tipos de Similaridade e Técnicas

Classificação de Similaridade para Código de Software	
Similaridade Representacional **(Sintática)**	Similaridade Textual
	Similaridade por Métricas
	Similaridade Baseada em Características
	Informação Compartilhada
Similaridade Comportamental **(Semântica)**	Similaridade na Curva de Execução
	Similaridade na Relação entre Entradas e Saídas
	Distância Semântica
	Distância na Equivalência da Abstração
	Grafo de Dependências do Programa

Sintática - Textual

- Algoritmos clássicos para comparação de strings
- Subseqüência Comum mais Longa (LCS)
  - diff (do Unix)
- Diferença de Levenshtein

Diferença de Levenshtein entre "caro" e "clara" é 2:

`caro`	→	`cara`	(substituição)
`cara`	→	`clara`	(inserção)

Sintática - Textual por Token

- Transformação inicial de strings para tokens

a seguinte instrução no código-fonte:

```
a = b==c : v[i] ? v[0]; // comentário
```

é transformada e parametrizada para:

```
v1=v2==v3:v4[v5]?v5[0];
```

- Comparam-se seqüências de tokens
- Hash para otimização
- Estrutura de dados: árvore de sufixos

Sintática - por Métricas

- Contagem de atributos do programa
  - número de chamadas de funções
  - número de variáveis locais usadas ou definidas
  - número de variáveis não-locais usadas ou definidas
  - número de parâmetros
  - número de sentenças
  - número de desvios
  - número de loops
- Fórmula sobre os valores acima: *score*

Sintática - Baseada em Características

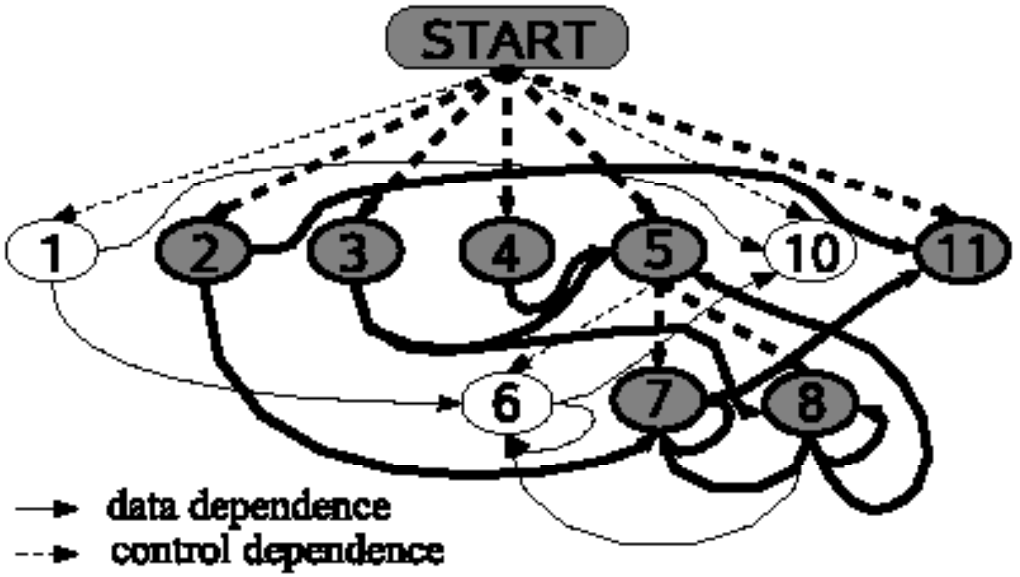
- k-gramas: strings de comprimento k
  - Algoritmo de Karp-Rabin
  - Divide-se o texto em vários k-gramas
  - Calcula-se hash de cada k-grama
  - Seleciona-se, por exemplo, os hashes múltiplos de um número dado: **fingerprint**
- janela de tamanho w: são w hashes consecutivos de k-gramas

Sintática - Informação Compartilhada

- Medida de similaridade a partir da
  - Teoria da Informação de Shannon
    - entropia (baixa entropia $\rightarrow$ alta redundância)
  - Complexidade de Kolmogorov
  - Quantidade de informação compartilhada

Semântica - Grafo de Dependências

```
1   sum = 0
2   mul = 1
3   a = 1
4   b = read()
5   while (a <= b) {
6     sum = sum + a
7     mul = mul * a
8     a = a + 1
9   }
10  write(sum)
11  write(mul)
```



- Objetivo: encontrar subgrafos isomorfos

Ferramentas

Classificação de Similaridade		Ferramentas
Similaridade Representacional **(Sintática)**	Textual	diff, Duploc, dup, siff, Plague, Yap, Jplag, CodeMatch, CCFInder, CloneDr
	por Métricas	Clan, Accuse, Yap3, Faidhi-Robson
	Baseada em Características	Karp-Rabin, Moss
	Informação Compartilhada	Sid
Similaridade Comportamental **(Semântica)**	Curva de Execução	
	Relação entre Entradas e Saídas	
	Distância Semântica	
	Distância na Equivalência da Abstração	
	Grafo de Dependências do Programa	Duplix

Jplag (Textual por Token)

Matches for 943151 & 942261

70.4%

943151 (73.0%)	942261 (67.0%)	Tokens
Jumpbox.java(1-8)	Jumpbox.java(1-10)	11
Jumpbox.java(8-11)	Jumpbox.java(10-13)	11
Jumpbox.java(23-46)	Jumpbox.java(22-44)	27
Jumpbox.java(64-73)	Jumpbox.java(73-82)	14

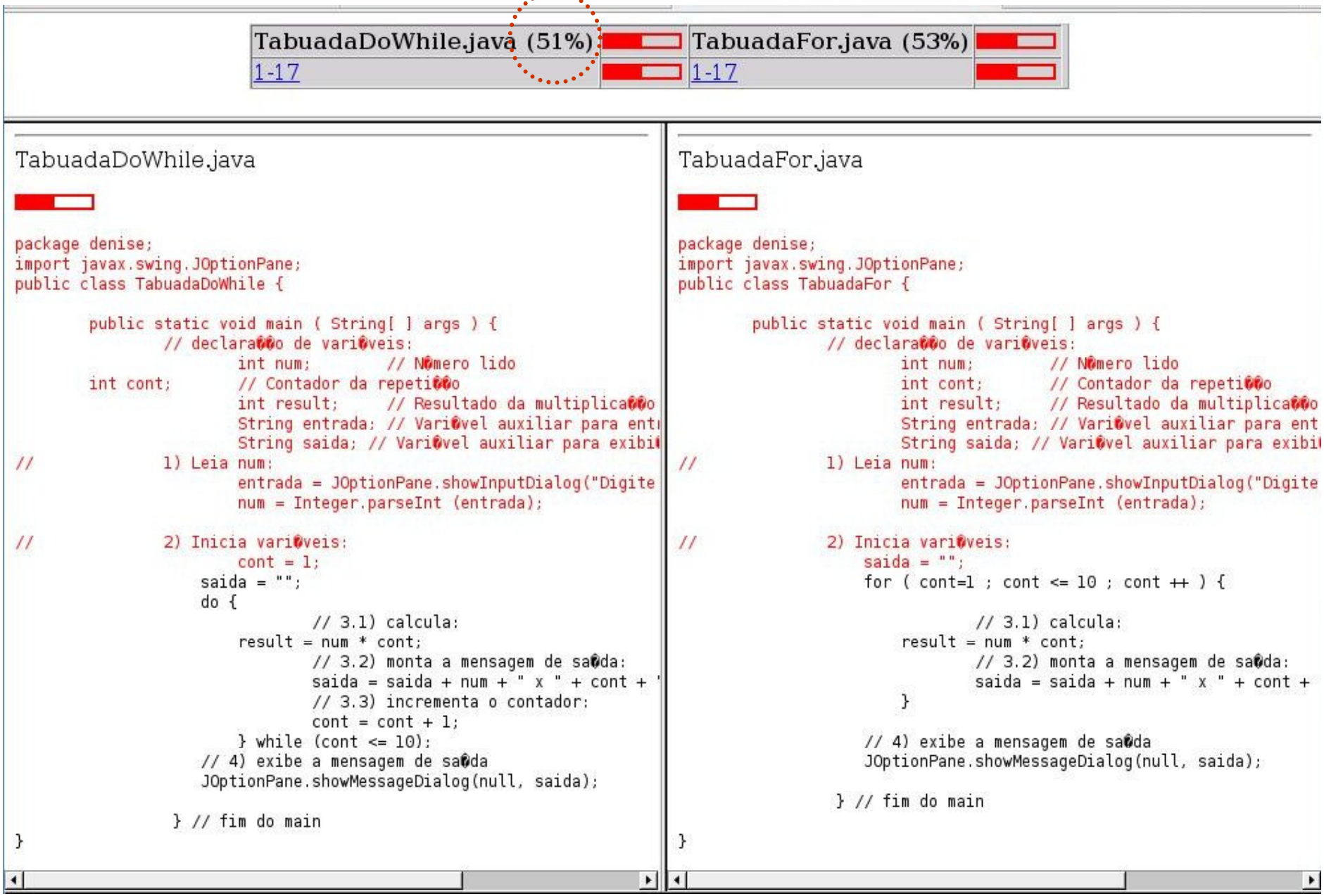
Jumpbox.java

```
import java.util.*;
import java.awt.*;
import java.awt.image.*;

public class Jumpbox extends Frame implements Runnable {

        Image smile,wince,offImage;
        int count = 0,size,x0,y0,xB,yB,currentBox,result=0,colors[]
        Long tempTime;
        long gameTime;
        Random random = new Random();
        Thread animationThread;
        Graphics offGraphics=null;
        boolean happyFace,threadOn=false;

        public Jumpbox(String title, String[] args) {
                super(title);

                if (args.length == 0){
                        tempTime = new Long("15");
                        gameTime = 15;
                }
                else {
                        try {
                                tempTime = new Long(args[0]);
                                gameTime = tempTime.longValue();
                        } catch (NumberFormatException ioe) {
                                System.out.println("ERROR: usage: J
```

Jumpbox.java

```
import java.util.*;
  import java.awt.*;
import java.awt.image.*;

public class Jumpbox extends Frame implements Runnable {

        Image smile, nosmile, offImage;
        Graphics offGraphics = null;
        int sizeImg;
   int nrChange = 0, xOldBox, yOldBox, xBox = 0, yBox = 0, typeBox
           Long gTime;
           long gameTime;
        Random rnd = new Random(System.currentTimeMillis());
        Thread animationThread;
        boolean bSmile, threadRunning = false;

        public Jumpbox(String title, String[] args) {
                super(title);
                if (args.length == 0) {
                        gameTime = 15;
                        gTime = new Long(15);
                } else {
                        try {
                                gTime = new Long(args[0]);
                                gameTime = gTime.longValue();
                        } catch (NumberFormatException ioe) {
                                System.out.println("Jumpbox: usage
                                System.exit(0);
```

Moss (Base em Características)



TabuadaDoWhile.java (51%)	TabuadaFor.java (53%)
1-17	1-17

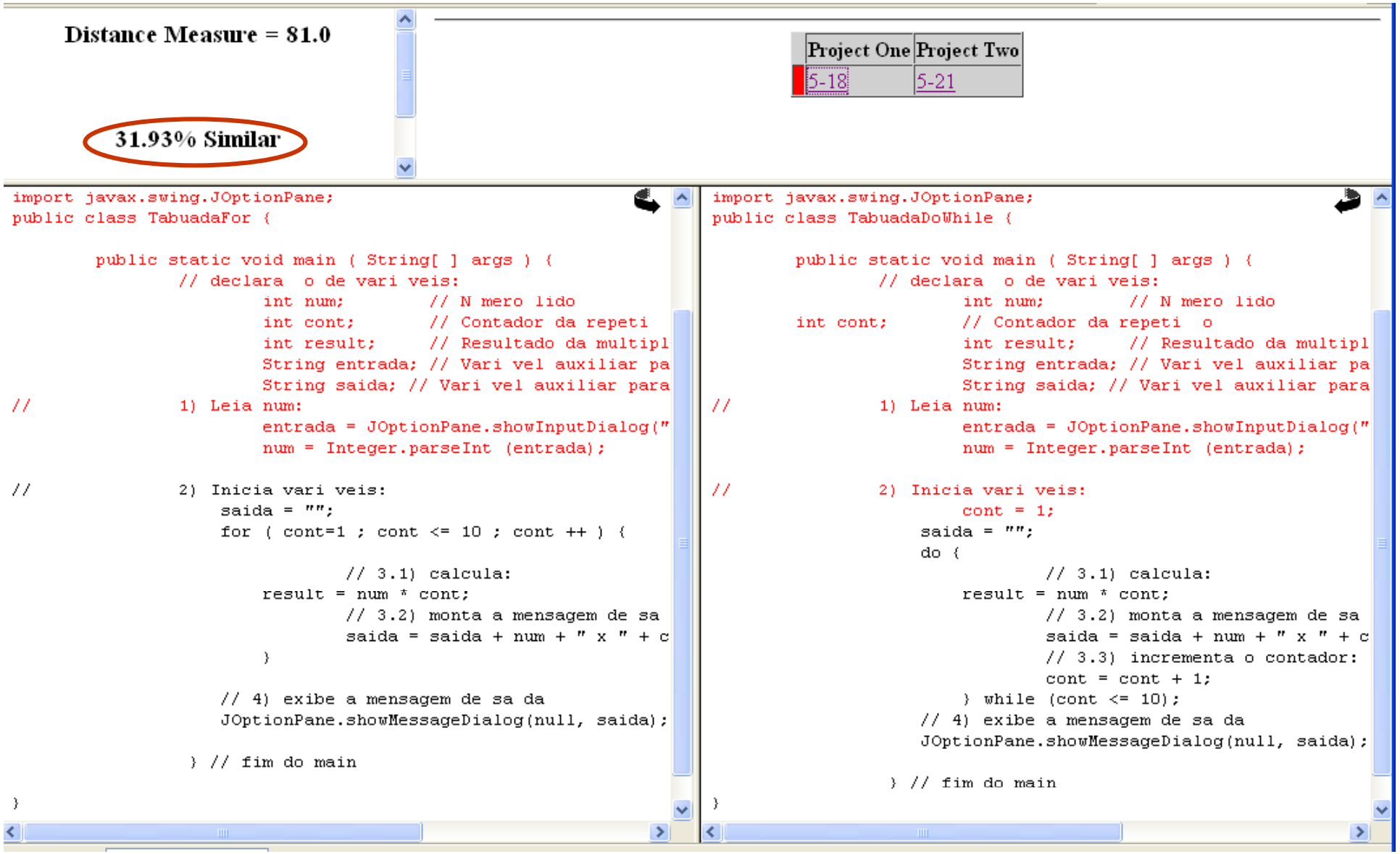
TabuadaDoWhile.java

```
package denise;
import javax.swing.JOptionPane;
public class TabuadaDoWhile {

        public static void main ( String[ ] args ) {
                // declara��o de vari�veis:
                        int num;        // N�mero lido
        int cont;       // Contador da repeti��o
                        int result;     // Resultado da multiplica��o
                        String entrada; // Vari�vel auxiliar para entr
                        String saida; // Vari�vel auxiliar para exibi
//              1) Leia num:
                        entrada = JOptionPane.showInputDialog("Digite
                        num = Integer.parseInt (entrada);

//              2) Inicia vari�veis:
                        cont = 1;
                    saida = "";
                    do {
                                // 3.1) calcula:
                        result = num * cont;
                                // 3.2) monta a mensagem de sa�da:
                                saida = saida + num + " x " + cont + '
                                // 3.3) incrementa o contador:
                                cont = cont + 1;
                        } while (cont <= 10);
                    // 4) exibe a mensagem de sa�da
                    JOptionPane.showMessageDialog(null, saida);

                } // fim do main
}
```

TabuadaFor.java

```
package denise;
import javax.swing.JOptionPane;
public class TabuadaFor {

        public static void main ( String[ ] args ) {
                // declara��o de vari�veis:
                        int num;        // N�mero lido
                        int cont;       // Contador da repeti��o
                        int result;     // Resultado da multiplica��o
                        String entrada; // Vari�vel auxiliar para ent
                        String saida; // Vari�vel auxiliar para exibi
//              1) Leia num:
                        entrada = JOptionPane.showInputDialog("Digite
                        num = Integer.parseInt (entrada);

//              2) Inicia vari�veis:
                    saida = "";
                    for ( cont=1 ; cont <= 10 ; cont ++ ) {

                                // 3.1) calcula:
                        result = num * cont;
                                // 3.2) monta a mensagem de sa�da:
                                saida = saida + num + " x " + cont +
                        }

                    // 4) exibe a mensagem de sa�da
                    JOptionPane.showMessageDialog(null, saida);

                } // fim do main
}
```

SID (Informação Compartilhada)



CodeMatch (CodeSuite) - Texto/Token

Matching comments and strings: (top of page)

File1 Line#	File2 Line#	Comment/String
6	6	declarao de variveis:
7	7	Nmero lido
8	8	Contador da repetio
9	9	Resultado da multiplicao
10	10	Varivel auxiliar para entrada de dados
11	11	Varivel auxiliar para exibio de mensagem
13	13	Digite um nmero:
16	16	2) Inicia variveis:
20	20	3.1) calcula:
22	22	3.2) monta a mensagem de sada:
23	23	\n
27	26	4) exibe a mensagem de sada
30	29	fim do main

Matching instruction sequences: (top of page)

File1 Line#	File2 Line#	Number of matching instructions
1	1	12

Matching identifiers: (top of page)

10	args	cont	denise	entrada	javax	JOptionPane	Leia
main	num	parseInt	result	saida	showInputDialog	showMessageDialog	swing

Matching statements: (top of page)

File1 Line#	File2 Line#	Matched line
1	1	package denise
2	2	import javax.swing.JOptionPane
5	5	public static void main (String[ ] args) {
7	7	int num
8	8	int cont
9	9	int result
10	10	String entrada
11	11	String saida
12	12	/ 1) Leia num:
13	13	entrada = JOptionPane.showInputDialog( )
14	14	num = Integer.parseInt (entrada)
18	17	saida =
21	21	
23	23	
28	27	

C:\Documents and Settings\All Users\Docu
\TabuadaDoWhile.java

Score Compared To File

87 C:\Documents and Settings\All Users\Document

Total number of bytes in files in folder 1 = 953

Total number of bytes in files in folder 2 = 906

Total run time = 0 Seconds

Conclusões

- Perito deve determinar o que comparar
  - abstrair, filtrar, análise manual
- Ferramentas de análise sintática:
  - Versões comerciais: CodeSuite e MossPlus
  - Versões on-line: Jplag, Moss e SID
- Comparativos entre ferramentas existentes: analisar com critério

Perguntas