

Java

Notas de Aula
Prof. André Bernardi
andrebernardi@unifei.edu.br





Plano de curso – ECOP05

- **ECOP05** - PROGRAMAÇÃO APLICADA (32h) -
Turma: 01 (2021.2)
- **Local** : RTE **Horário**: 2M23
- **Ementa**:
 - Aplicativos Java. Estruturas de controle. Métodos. Arrays. Programação baseada em objetos. Programação orientada a objetos. Strings. Componentes gráficos. Componentes GUI. Tratamento de exceções. Multithreading. Multimídia. Entrada e saída usual. Entrada e saída em arquivos. Introdução a análise e projetos orientados a objetos



Plano de curso – ECOP05

Início	Fim	Descrição
30/08/2021	30/08/2021	Apresentação da disciplina e Revisão
06/09/2021	06/09/2021	Manipulação de Strings
13/09/2021	13/09/2021	Desenho e Java 2D
20/09/2021	20/09/2021	Manipulação de Arrays e Java 2D
27/09/2021	27/09/2021	Interface Gráfica e Componentes
04/10/2021	04/10/2021	Interface Gráfica - Layout e Painéis
11/10/2021	11/10/2021	Interface Gráfica - Janela e Menus
18/10/2021	18/10/2021	Primeira Avaliação
25/10/2021	25/10/2021	Processamento de Exceções
01/11/2021	01/11/2021	Janelas Internas e Subclasse de JPanel
08/11/2021	08/11/2021	Persistência de dados - Entrada e Saida em Arquivos
22/11/2021	22/11/2021	Multimidia e Threads
29/11/2021	29/11/2021	Arquivos Multimidia e Manipulacao de Imagens
06/12/2021	06/12/2021	Segunda Avaliação
13/12/2021	13/12/2021	Classes para utilização de banco de dados



Avaliações

- Primeiro Bimestre:
 - 18/10/2021 às 7:55h

- Segundo Bimestre:
 - 06/12/2021 às 7:55h

- Prova Extra:
 - 03/01/2022 às 7:55h



Referências

- Notas de aula da disciplina
- Deitel & Deitel – Java How to Program
- Oracle



Característica Básica

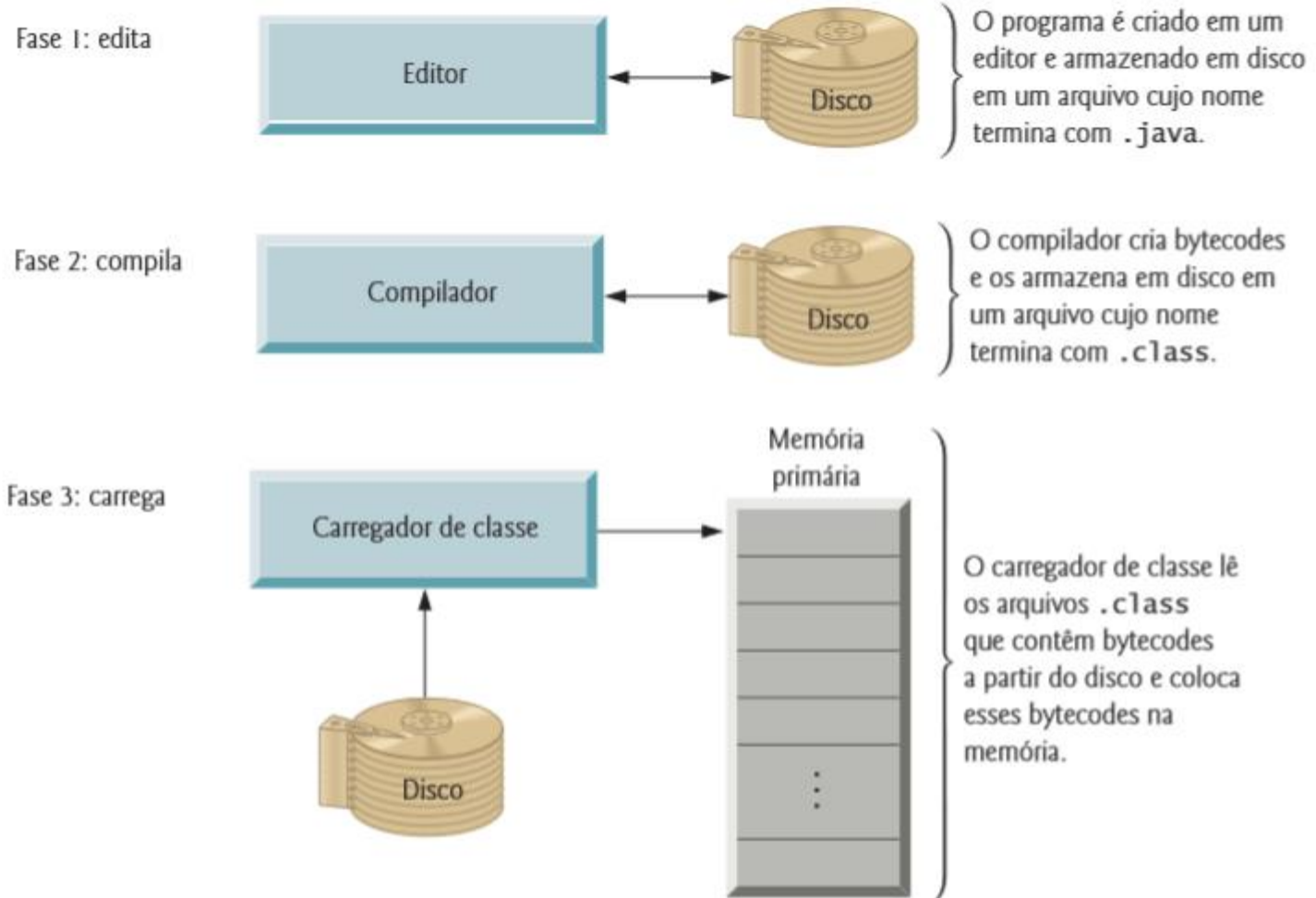
- Capacidade de execução em diferentes plataformas e sistemas operacionais, sem a necessidade de recompilação do código fonte.



Subdivisões da Tecnologia Java

- Java foi oficialmente apresentado em 1995 pelo consorcio Sun.
- Em 1999 surge a chamada segunda edição do Java na qual foram criadas as seguintes denominações:
 - JME – Java Micro Edition
 - JSE – Java Standard Edition
 - JEE – Java Enterprise Edition

Ambiente Java



Fase 4: verifica



Memória
primária



O verificador de bytecode confirma que todos os bytecodes são válidos e não violam restrições de segurança do Java.

Fase 5: executa



Memória
primária



Para executar o programa, a JVM lê os bytecodes e os compila (isto é, traduz) no momento certo (ou *Just-In-Time* — JIT) para uma linguagem que o computador possa entender. Como o programa existe na memória ele pode armazenar valores de dados na memória principal.



Java Application

```
// Welcome1.java - A first program in Java.  
public class Welcome1  
{  
    // main method begins execution of Java application  
    public static void main( String args[] )  
    {  
        System.out.println("Welcome to Java Programming!");  
    } // end method main  
  
} // end class Welcome1
```



Compilação

- Obter e instalar o compilador Java SE gratuitamente na pagina da Oracle.
- Compilar em linha de comando
 - `javac Welcome1.java`
- Executar em linha de comando
 - `java Welcome1`

Compilação



```
C:\examples\ch02\fig02_01>javac Welcome1.java
C:\examples\ch02\fig02_01>java Welcome1
Welcome to Java Programming!
C:\examples\ch02\fig02_01>
```

Você digita esse comando para executar o aplicativo

O programa gera a saída na tela
Welcome to Java Programming!



Links para obter o Java.

- Java SE
- <https://www.oracle.com/java/technologies/javase-downloads.html>
- Netbeans + JDK
- <https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>

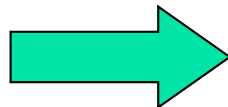
[Java SE](#)[Java EE](#)[Java ME](#)[Java SE Subscription](#)[Java Embedded](#)[Java Card](#)[Java TV](#)[Community](#)[Java Magazine](#)[Overview](#)[Downloads](#)[Documentation](#)[Community](#)[Technologies](#)[Training](#)

JDK 8u111 with NetBeans 8.2

This distribution of the JDK includes the Java SE bundle of [NetBeans IDE](#), which is a powerful integrated development environment for developing applications on the Java platform. [Learn more](#)

You must accept the [JDK 8u111 and NetBeans 8.2 Cobundle License Agreement](#) to download this software.

☐ Accept License Agreement ☒ Decline License Agreement



Java SE and NetBeans Cobundle (JDK 8u111 and NB 8.2)		
Product / File Description	File Size	Download
Linux x86	286.73 MB	jdk-8u111-nb-8_2-linux-i586.sh
Linux x64	282.57 MB	jdk-8u111-nb-8_2-linux-x64.sh
Mac OS X x64	342.99 MB	jdk-8u111-nb-8_2-macosx-x64.dmg
Windows x86	317.21 MB	jdk-8u111-nb-8_2-windows-i586.exe
Windows x64	326.03 MB	jdk-8u111-nb-8_2-windows-x64.exe

- [License](#)
- [Java SE 8 Readme](#)
- [NB 8.2 3rd Party Readme](#)
- [Installation Instructions](#)
- [Java SE Release Notes](#)
- [NetBeans Release Notes](#)

Java SE Development Kit 8u111 和 NetBeans IDE 8.2 复合软件包下载 (简体中文)

- [安装说明](#)
- [Java SE 发行说明](#)
- [NetBeans 发行说明](#)

Java SE Development Kit 8u111 e NetBeans IDE 8.2 Bundle (Português do Brasil)

- [Instruções de instalação](#)
- [Notas da versão do Java SE](#)
- [Notas da versão do NetBeans](#)

Enter first integer: 45
Enter second integer: 72
Sum is 117

Primeiro exemplo

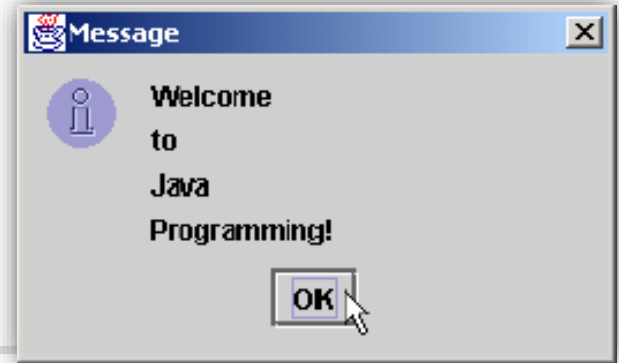
```
// Addition program that displays the sum of two numbers.
import java.util.Scanner; // program uses class Scanner
public class Addition
{
    // main method begins execution of Java application
    public static void main( String args[] )
    { // create Scanner to obtain input from command window
        Scanner input = new Scanner( System.in );
        int number1;        // first number to add
        int number2;        // second number to add
        int sum;            // sum of number1 and number2

        System.out.print( "Enter first integer: " ); // prompt
        number1 = input.nextInt(); // read first number from user

        System.out.print( "Enter second integer: " ); // prompt
        number2 = input.nextInt(); // read second number from user

        sum = number1 + number2; // add numbers
        System.out.printf( "Sum is %d\n", sum ); // display sum
    } // end method main
} // end class Addition
```

Segundo exemplo



```
// Welcome4.java - Printing multiple lines in a dialog box
// import class JOptionPane
import javax.swing.JOptionPane;

public class Welcome4
{ // main method begins execution of Java application
    public static void main( String args[] )
    {
        JOptionPane.showMessageDialog(null,
            "Welcome\nto\nJava\nProgramming!" );
        System.exit( 0 ); // terminate application
    } // end method main
} // end class Welcome4
```

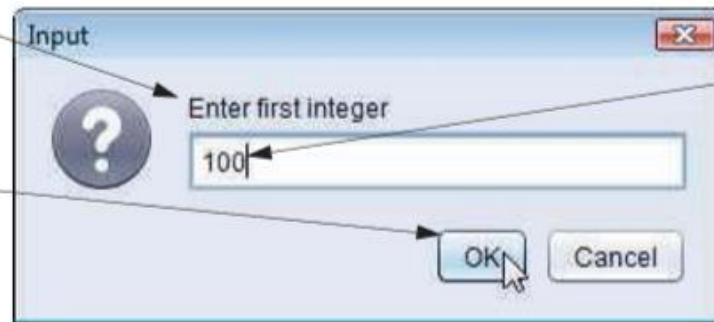

Terceiro Exemplo

```
1 // Figura 12.2: Addition.java
2 // Programa de adição que utiliza JOptionPane para entrada e saída.
3 import javax.swing.JOptionPane;
4
5 public class Addition
6 {
7     public static void main(String[] args)
8     {
9         // obtém a entrada de usuário a partir dos diálogos de entrada JOptionPane
10        String firstNumber =
11            JOptionPane.showInputDialog("Enter first integer");
12        String secondNumber =
13            JOptionPane.showInputDialog("Enter second integer");
14
15        // converte String em valores int para utilização em um cálculo
16        int number1 = Integer.parseInt(firstNumber);
17        int number2 = Integer.parseInt(secondNumber);
18
19        int sum = number1 + number2;
20
21        // exibe o resultado em um diálogo de mensagem JOptionPane
22        JOptionPane.showMessageDialog(null, "The sum is " + sum,
23            "Sum of Two Integers", JOptionPane.PLAIN_MESSAGE);
24    }
25 } // fim da classe Addition
```

Terceiro Exemplo

(a) Diálogo de entrada exibido nas linhas 10 e 11

Solicitação para o usuário
Quando o usuário clica em **OK**,
`showInputDialog` retorna
para o programa o 100 digitado
pelo usuário como uma `String`;
o programa deve converter a
`String` em um `int`

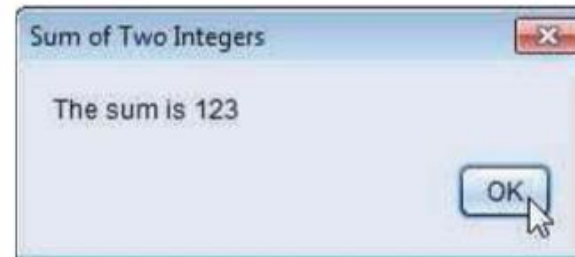


Campo de texto no qual o
usuário digita um valor





(b) Diálogo de entrada exibido nas linhas 12 e 13



(c) O diálogo de mensagem exibido pelas linhas 22 e 23. Quando o
usuário clica em **OK**, o diálogo de mensagem é fechado (removido da tela)



Opções de ícones

Tipo de diálogo de mensagem	Ícone	Descrição
ERROR_MESSAGE		Indica um erro.
INFORMATION_MESSAGE		Indica uma mensagem informativa.
WARNING_MESSAGE		Alerta de um potencial problema.
QUESTION_MESSAGE		Faz uma pergunta. Normalmente, esse diálogo exige uma resposta, como clicar em um botão Yes ou No.
PLAIN_MESSAGE	Sem ícone	Um diálogo que contém uma mensagem, mas nenhum ícone.



Operadores



Operadores Aritméticos

Operação Java	Operador	Expressão algébrica	Expressão Java
Adição	+	$f + 7$	<code>f + 7</code>
Subtração	-	$p - c$	<code>p - c</code>
Multiplicação	*	bm	<code>b * m</code>
Divisão	/	x/y ou $\frac{x}{y}$ ou $x \div y$	<code>x / y</code>
Resto	%	$r \bmod s$	<code>r % s</code>

Operador(es)	Operação(ões)	Ordem de avaliação (precedência)
* / %	Multiplicação Divisão Resto	Avaliado primeiro. Se houver vários operadores desse tipo, eles são avaliados da <i>esquerda para a direita</i> .
+ -	Adição Subtração	Avaliado em seguida. Se houver vários operadores desse tipo, eles são avaliados da <i>esquerda para a direita</i> .
=	Atribuição	Avaliado por último.



Operadores Relacionais

Standard algebraic equality or relational operator	Java equality or relational operator	Example of Java condition	Meaning of Java condition
<i>Equality operators</i>			
=	==	x == y	x is equal to y
≠	!=	x != y	x is not equal to y
<i>Relational operators</i>			
>	>	x > y	x is greater than y
<	<	x < y	x is less than y
≥	>=	x >= y	x is greater than or equal to y
≤	<=	x <= y	x is less than or equal to y

Operadores

Assignment operator	Sample expression	Explanation	Assigns
<i>Assume: int c = 3, d = 5, e = 4, f = 6, g = 12;</i>			
+=	c += 7	c = c + 7	10 to c
-=	d -= 4	d = d - 4	1 to d
*=	e *= 5	e = e * 5	20 to e
/=	f /= 3	f = f / 3	2 to f
%=	g %= 9	g = g % 9	3 to g

Operator	Called	Sample expression	Explanation
++	preincrement	++a	Increment a by 1, then use the new value of a in the expression in which a resides.
++	postincrement	a++	Use the current value of a in the expression in which a resides, then increment a by 1.
--	predecrement	--b	Decrement b by 1, then use the new value of b in the expression in which b resides.
--	postdecrement	b--	Use the current value of b in the expression in which b resides, then decrement b by 1.



Operadores Lógicos

expression1	expression2	expression1 && expression2
false	false	false
false	true	false
true	false	false
true	true	true

expression1	expression2	expression1 expression2
false	false	false
false	true	true
true	false	true
true	true	true

expression	! expression
false	true
true	false

expression1	expression2	expression1 ^ expression2
false	false	false
false	true	true
true	false	true
true	true	false

Regras de precedência

Operators	Associativity	Type
()	left to right	parentheses
++ --	right to left	unary postfix
++ -- + - ! (type)	right to left	unary
* / %	left to right	multiplicative
+ -	left to right	additive
< <= > >=	left to right	relational
== !=	left to right	equality
&	left to right	boolean logical AND
^	left to right	boolean logical exclusive OR
	left to right	boolean logical inclusive OR
&&	left to right	logical AND
	left to right	logical OR
? :	right to left	conditional
= += -= *= /= %=	right to left	assignment



Palavras reservadas

abstract	boolean	break	byte	case
catch	char	class	continue	default
do	double	else	extends	false
final	finally	float	for	if
implements	import	instanceof	int	interface
long	native	new	null	package
private	protected	public	return	short
static	super	switch	synchronized	this
throw	throws	transient	true	try
void	volatile	while		

Keywords that are reserved, but not used, by Java

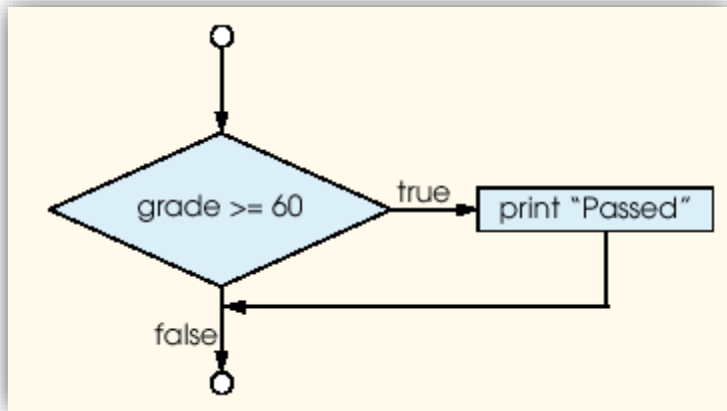
const	goto
--------------	-------------

Type	Size in bits	Values	Standard
boolean	8	true or false	
char	16	'\u0000' to '\uFFFF' (0 to 65535)	(ISO Unicode character set)
byte	8	−128 to +127 (-2^7 to $2^7 - 1$)	
short	16	−32,768 to +32,767 (-2^{15} to $2^{15} - 1$)	
int	32	−2,147,483,648 to +2,147,483,647 (-2^{31} to $2^{31} - 1$)	
long	64	−9,223,372,036,854,775,808 to +9,223,372,036,854,775,807 (-2^{63} to $2^{63} - 1$)	
float	32	<i>Negative range:</i> −3.4028234663852886E+38 to −1.40129846432481707e−45 <i>Positive range:</i> 1.40129846432481707e−45 to 3.4028234663852886E+38	(IEEE 754 floating point)
double	64	<i>Negative range:</i> −1.7976931348623157E+308 to −4.94065645841246544e−324 <i>Positive range:</i> 4.94065645841246544e−324 to 1.7976931348623157E+308	(IEEE 754 floating point)



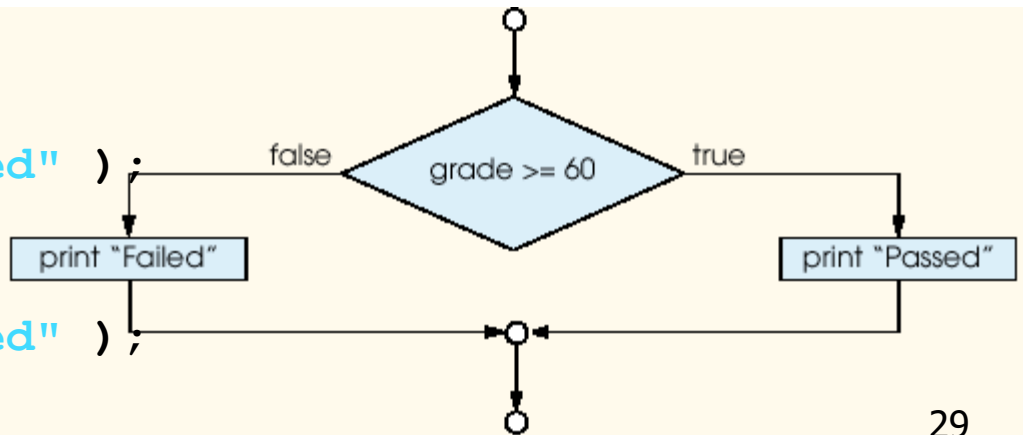
Estruturas de Controle

Estrutura Condicional



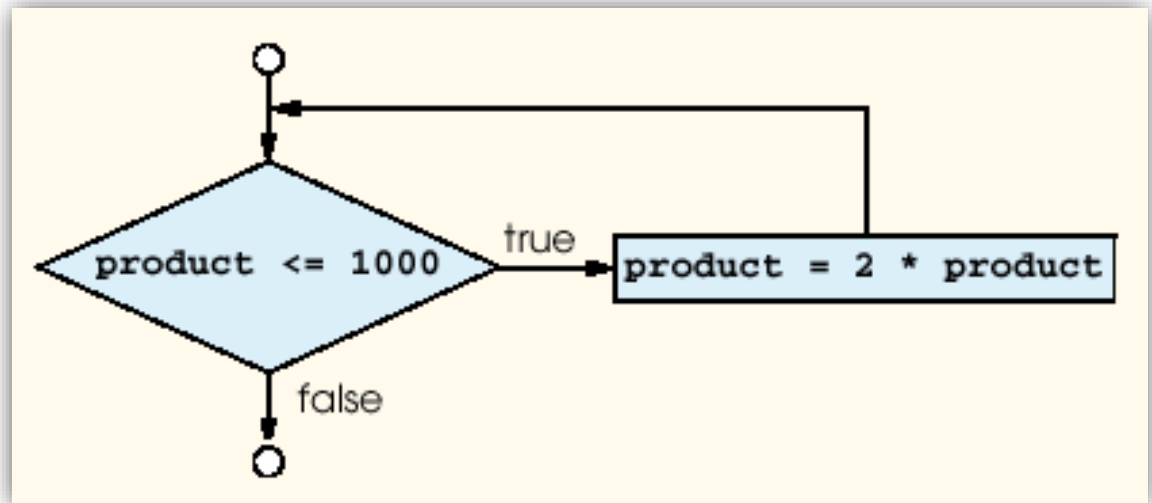
```
if ( studentGrade >= 60 )  
    System.out.println( "Passed" );
```

```
if ( studentGrade >= 60 )  
    System.out.println( "Passed" );  
else  
    System.out.println( "Failed" );
```



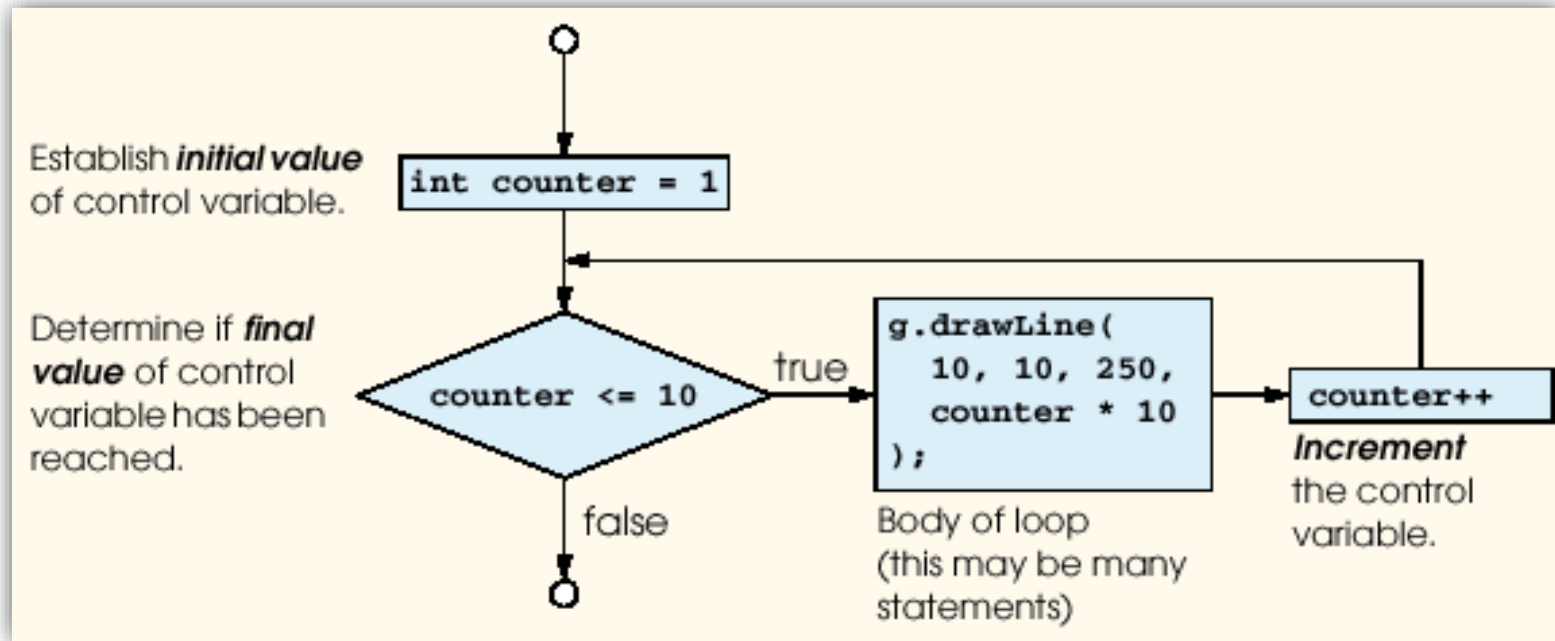
Estrutura while

```
int product = 2;  
while ( product <= 1000 )  
    product = 2 * product;
```

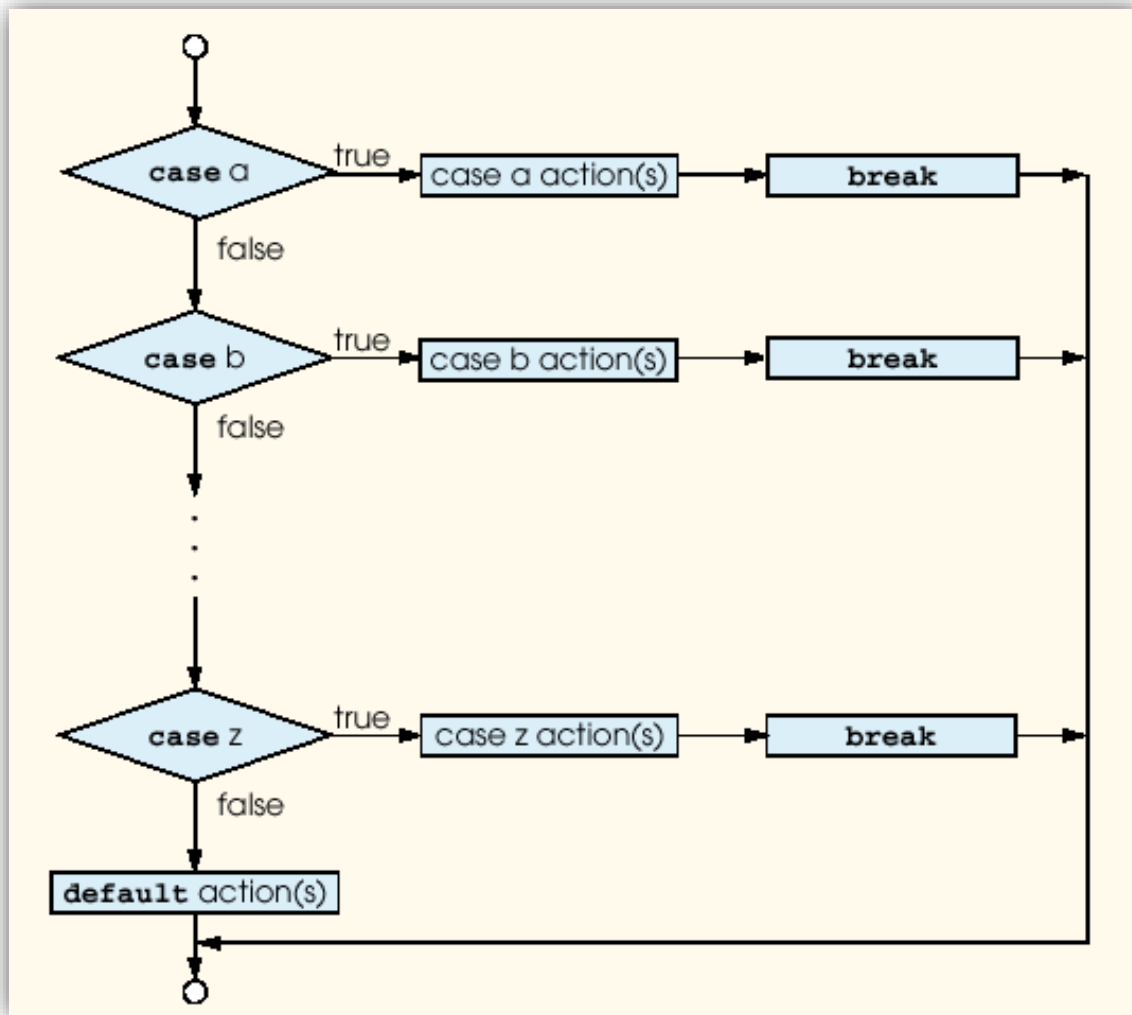


Estrutura for

```
for ( int counter = 1; counter <= 10; counter++ )  
    g.drawLine( 10, 10, 250, counter * 10 );
```



Estrutura switch



Estrutura do/while

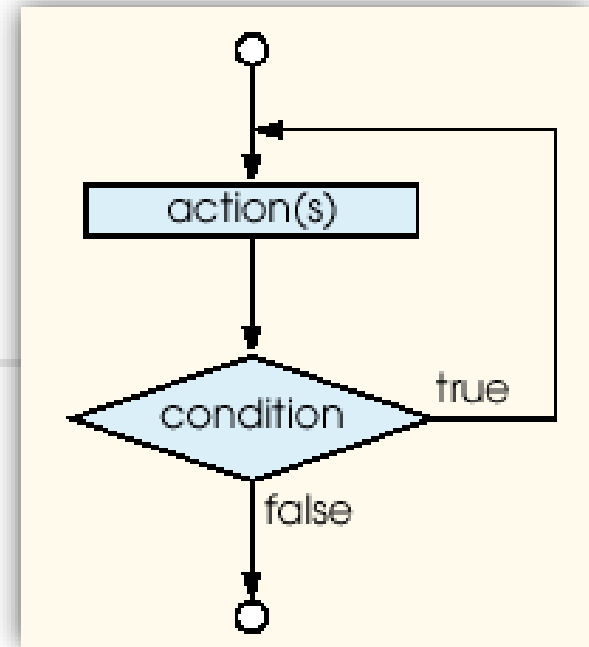
```
int counter = 1;
```

```
do {
```

```
    g.drawOval( 110 - counter * 10, 110 -  
                counter * 10, counter * 20,  
                counter * 20 );
```

```
    ++counter;
```

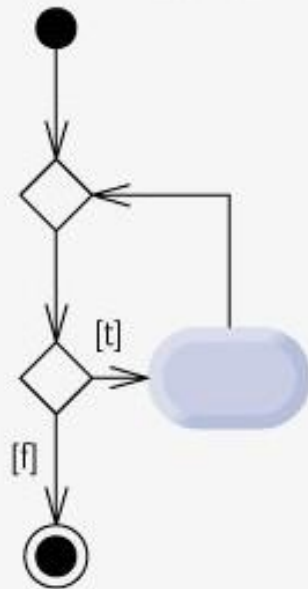
```
} while ( counter <= 10 ); // end do/while
```



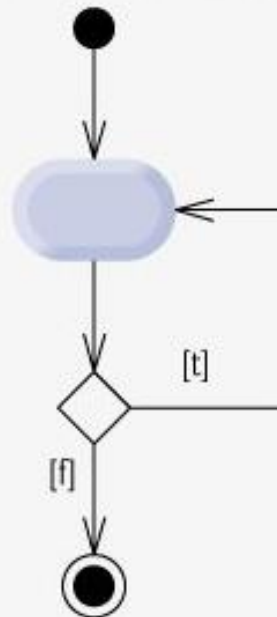
Resumo das estruturas

Repetition

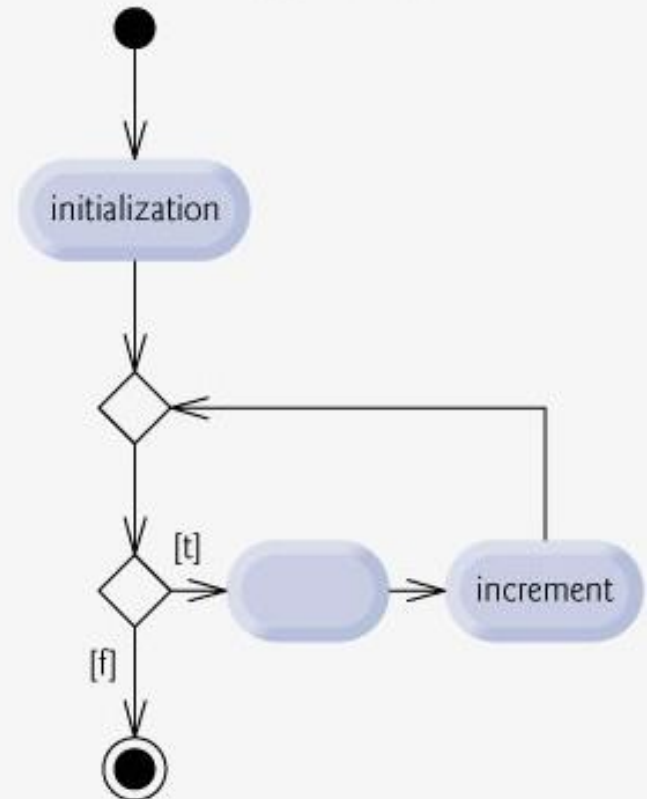
while statement



do...while statement



for statement

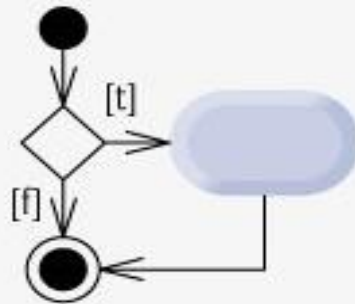


Sequence

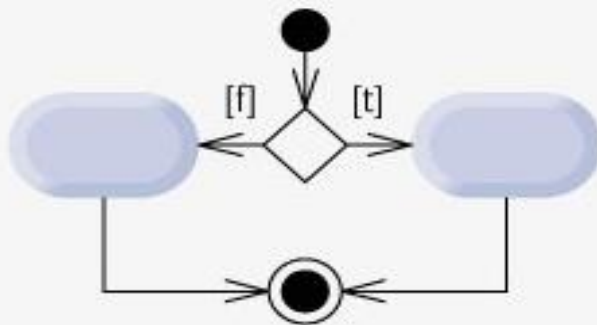


Selection

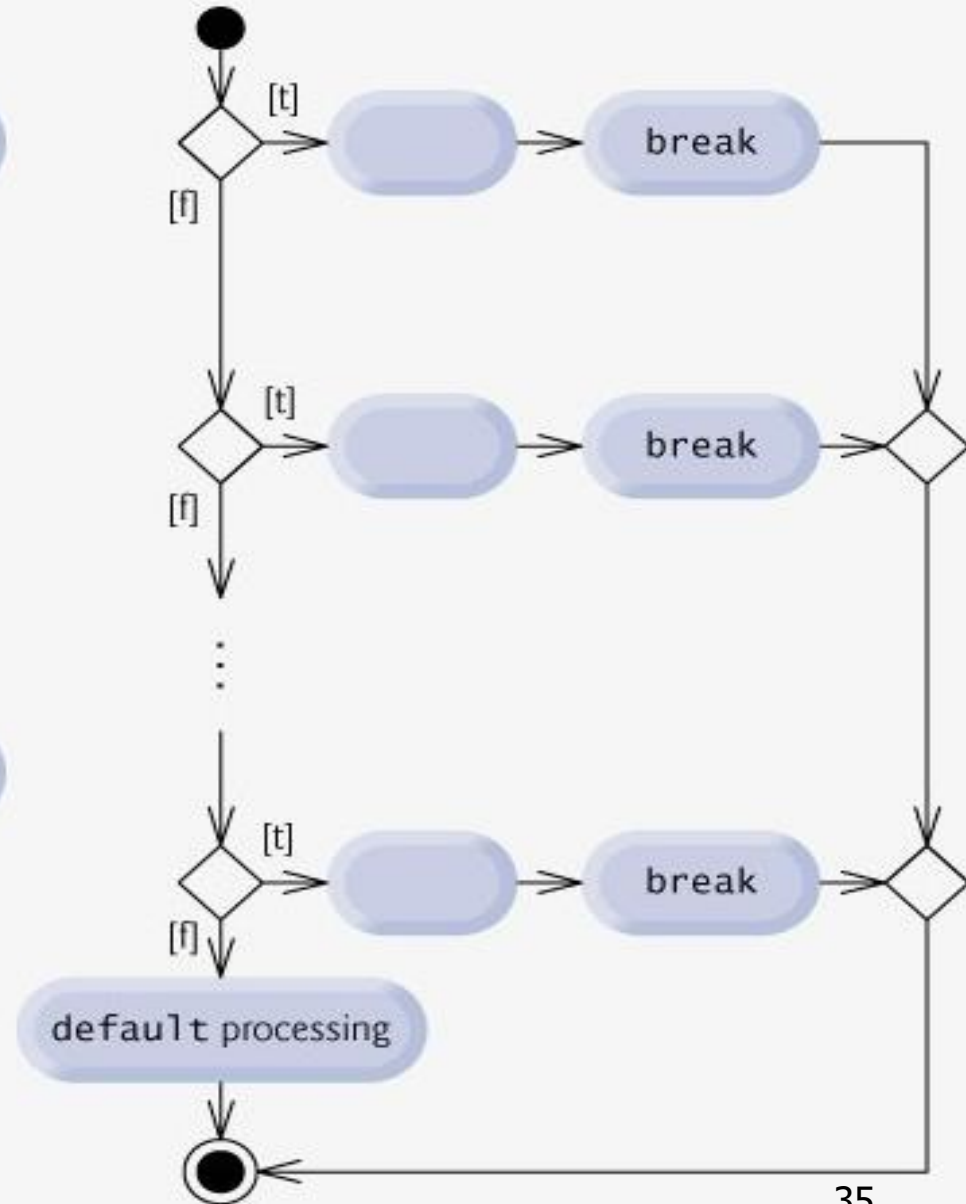
if statement
(single selection)



if...else statement
(double selection)



switch statement with breaks
(multiple selection)





Métodos

Outra denominação para “função
membro”



Métodos da Classe Math

Method	Description	Example
abs(x)	absolute value of x (this method also has versions for float , int and long values)	abs(23.7) is 23.7 abs(0.0) is 0.0 abs(-23.7) is 23.7
ceil(x)	rounds x to the smallest integer not less than x	ceil(9.2) is 10.0 ceil(-9.8) is -9.0
cos(x)	trigonometric cosine of x (x is in radians)	cos(0.0) is 1.0
exp(x)	exponential method e^x	exp(1.0) is 2.71828 exp(2.0) is 7.38906
floor(x)	rounds x to the largest integer not greater than x	floor(9.2) is 9.0 floor(-9.8) is -10.0
log(x)	natural logarithm of x (base e)	log(2.718282) is 1.0 log(7.389056) is 2.0



Métodos da Classe Math

Method	Description	Example
<code>max(x, y)</code>	larger value of x and y (this method also has versions for float , int and long values)	<code>max(2.3, 12.7)</code> is 12.7 <code>max(-2.3, -12.7)</code> is -2.3
<code>min(x, y)</code>	smaller value of x and y (this method also has versions for float , int and long values)	<code>min(2.3, 12.7)</code> is 2.3 <code>min(-2.3, -12.7)</code> is -12.7
<code>pow(x, y)</code>	x raised to power y (x^y)	<code>pow(2.0, 7.0)</code> is 128.0 <code>pow(9.0, .5)</code> is 3.0
<code>sin(x)</code>	trigonometric sine of x (x is in radians)	<code>sin(0.0)</code> is 0.0
<code>sqrt(x)</code>	square root of x	<code>sqrt(900.0)</code> is 30.0 <code>sqrt(9.0)</code> is 3.0
<code>tan(x)</code>	trigonometric tangent of x (x is in radians)	<code>tan(0.0)</code> is 0.0



Geração de números aleatórios

- Método `random()` da classe **Math**
 - Gera sempre valores em `double` na faixa $0.0 \leq x < 1.0$
- Classe **Random** da biblioteca `java.util`
 - Pode gerar várias formas de números aleatórios, através do uso das funções:
`nextInt`, `nextDouble`, `nextFloat`,
`nextGaussian`, `nextLong`, `nextBoolean` ...



Geração de números aleatórios

```
int a = (int) ( deslocamento +  
    Math.random()*escala);
```

- Gerar valores entre 100 e 899

```
int a = (int) ( 100 + Math.random()*800);
```

- Random randomNumbers = new Random();

```
int a = randomNumbers.nextInt();
```

```
int a = 100 + randomNumbers.nextInt(800);
```




Programação orientada a objetos

- Baseada na criação de classes contendo métodos e atributos

```
public class GradeBook
{
    private String atributo;
    public void setAtributo(String a)
    {
        atributo = a;
    }
    public String getAtributo()
    {
        return atributo;
    }
    public void displayMessage()
    {
        System.out.println( "Hello World!! " + getAtributo() );
    } // end method displayMessage
} // end class GradeBook
```



Referências

- Java How to Program 3, 4, 5, 6, 7, 8, 9, 10 ed. - Paul Deitel and Harvey Deitel
- Sun (<http://java.sun.com>)
- Oracle (<http://www.oracle.com/technetwork/java>)