

# Curso Java COMPLETO

Capítulo: Introdução sobre programação

<http://educandoweb.com>

Prof. Dr. Nélcio Alves

---

---

---

---

---

---

---

## Algoritmo, Automação, Programa de Computador

<http://educandoweb.com.br>

Prof. Dr. Nélcio Alves

---

---

---

---

---

---

---

### Algoritmo

Sequência finita de instruções para se resolver um problema.

\* aplica-se a diversas áreas de conhecimento

---

---

---

---

---

---

---

Exemplo:

**Problema:** lavar roupa suja

**Algoritmo:**

- 1) Colocar a roupa em um recipiente
- 2) Colocar um pouco de sabão e amaciante
- 3) Encher de água
- 4) Mexer tudo até dissolver todo o sabão
- 5) Deixar de molho por vinte minutos
- 6) Esfregar a roupa
- 7) Enxaguar
- 8) Torcer

---

---

---

---

---

---

---

## Automação

Consiste em utilizar máquina(s) para executar o procedimento desejado de forma automática ou semiautomática.

---

---

---

---

---

---

---

**Algoritmo:**

- 1) Colocar a roupa em um recipiente
- 2) Colocar um pouco de sabão e amaciante
- 3) Encher de água
- 4) Mexer tudo até dissolver todo o sabão
- 5) Deixar de molho por vinte minutos
- 6) Esfregar a roupa
- 7) Enxaguar
- 8) Torcer



---

---

---

---

---

---

---

Mas o que algoritmo e automação tem a ver com programação de computadores?

---

---

---

---

---

---

---

## Computador

- Hardware - parte física (a máquina em si)
- Software - parte lógica (programas)
  - Sistema operacional (Windows, Linux, iOS)
  - Aplicativos (apps de escritório, app de câmera, navegador web)
  - Jogos
  - Utilitários (Antivírus, compactador de arquivos)
  - Outros



---

---

---

---

---

---

---

## Programa ~ Algoritmo

Programas de computador **são algoritmos** executados pelo computador (em linhas gerais).

Conclusão: o computador é uma máquina que **automatiza** a execução de **algoritmos**.

Qualquer algoritmo? Não. Apenas algoritmos computacionais:

- Processamento de dados
- Cálculos

---

---

---

---

---

---

---

## Resumo da aula

- Algoritmo: sequência finita de instruções para se resolver um problema
- Automação: quando uma máquina realiza o algoritmo
- Computador:
  - hardware / software
  - máquina que automatiza algoritmos (de cálculo)
- Programa de computador: algoritmo executado pelo computador

---

---

---

---

---

---

---

## O que é preciso para se fazer um programa de computador?

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Vamos precisar de:

- Uma **linguagem de programação**: regras **léxicas** e **sintáticas** para se escrever o programa
- Uma **IDE**: software para editar e testar o programa
- Um **compilador**: software para transformar o **código fonte** em **código objeto**
- Um **gerador de código** ou **máquina virtual**: software que permite que o programa seja executado

---

---

---

---

---

---

---

## Linguagem de programação, léxica, sintática

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

Vamos precisar de:

- Uma **linguagem de programação**: regras **léxicas** e **sintáticas** para se escrever o programa
- Uma **IDE**: software para editar e testar o programa
- Um **compilador**: software para transformar o **código fonte** em **código objeto**
- Um **gerador de código** ou **máquina virtual**: software que permite que o programa seja executado

---

---

---

---

---

---

---

## Linguagem de programação

É um conjunto de regras **léxicas** (ortografia) e **sintáticas** (gramática) para se escrever programas.

---

---

---

---

---

---

---

## Léxica

Diz respeito à correção das **palavras** "isoladas" (ortografia).

### Exemplo (Português):

cachorro

caxorro



### Linguagem de programação:

main

maim



---

---

---

---

---

---

---

## Sintática

Diz respeito à correção das **sentenças** (gramática).

### Exemplo (Português):

O cachorro está com fome.


A cachorro está com fome.



### Linguagem de programação:

x = 2 + y;

x = + 2 y;



---

---

---

---

---

---

---

## Linguagem de programação

### Exemplos de linguagens de programação:

C, Pascal, C++, Java, C#, Python, Ruby, PHP, JavaScript, etc.

---

---

---

---

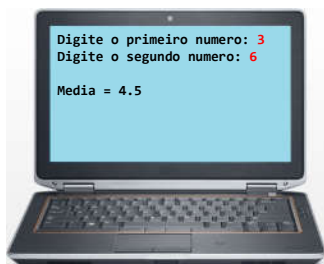
---

---

---

### Exemplo de um programa:

Suponha um programa que solicita do usuário dois números e depois mostra a média aritmética deles:



---

---

---

---

---

---

---

### Solução em linguagem C

```
#include <stdio.h>

int main() {
    double x, y, media;

    printf("Digite o primeiro numero: ");
    scanf("%lf", &x);
    printf("Digite o segundo numero: ");
    scanf("%lf", &y);
    media = (x + y) / 2.0;
    printf("Media = %.1f\n", media);
    return 0;
}
```

---

---

---

---

---

---

---

### Solução em linguagem C++

```
#include <iostream>

using namespace std;

int main() {
    double x, y, media;

    cout << "Digite o primeiro numero: ";
    cin >> x;
    cout << "Digite o segundo numero: ";
    cin >> y;
    media = (x + y) / 2.0;
    cout << "Media = " << media << endl;
    return 0;
}
```

---

---

---

---

---

---

---

## Solução em linguagem C#

```
using System;

namespace programa {
    class Program {
        static void Main(string[] args) {
            double x, y, media;

            Console.Write("Digite o primeiro numero: ");
            x = double.Parse(Console.ReadLine());
            Console.Write("Digite o segundo numero: ");
            y = double.Parse(Console.ReadLine());
            media = (x + y) / 2.0;
            Console.WriteLine("Media = " + media);
        }
    }
}
```

---

---

---

---

---

---

---

---

## Solução em linguagem Java

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double x, y, media;

        System.out.print("Digite o primeiro numero: ");
        x = sc.nextDouble();
        System.out.print("Digite o segundo numero: ");
        y = sc.nextDouble();
        media = (x + y) / 2.0;
        System.out.println("Media = " + media);
        sc.close();
    }
}
```

---

---

---

---

---

---

---

---

## Resumo da aula

- Linguagem: conjunto de regras léxicas e sintáticas para se escrever um programa
  - Léxica = ortografia. Palavras isoladas.
  - Sintática = gramática. Sentença como um todo.
- Exemplos de linguagens: C, Pascal, C++, Java, C#, Python, Ruby, PHP, JavaScript, etc.
- Exemplo de códigos feitos em linguagem C, C++, C# e Java

---

---

---

---

---

---

---

---



## IDE: Ambiente de Desenvolvimento Integrado

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

### IDE – Ambiente Integrado de Desenvolvimento

É um conjunto de softwares utilizado para a construção de programas.

Exemplos:

C/C++ : **Code Blocks**

Java : **Eclipse, NetBeans**

C# : **Microsoft Visual Studio**

---

---

---

---

---

---

---

### Funcionalidades de uma IDE

- Edição de código fonte (indentação, autocompletar, destaque de palavras, etc.)
- Depuração e testes
- Construção do produto final (build)
- Sugestão de modelos (templates)
- Auxiliar em várias tarefas do seu projeto
- Etc.

---

---

---

---

---

---

---

## Resumo da aula

- IDE: é um conjunto de softwares utilizado para a construção de programas
  - C/C++ : **Code Blocks**
  - Java : **Eclipse, NetBeans**
  - C# : **Microsoft Visual Studio**
- Uma IDE oferece várias funcionalidades para facilitar a construção dos programas

---

---

---

---

---

---

---

---

## Compilação e interpretação Código fonte e objeto Máquina virtual

<http://educandoweb.com.br>

Prof. Dr. Nélio Alves

---

---

---

---

---

---

---

---

**Código fonte:** é aquele escrito pelo programador em linguagem de programação

```
#include <stdio.h>

int main() {
    double x, y, media;

    printf("Digite o primeiro numero: ");
    scanf("%lf", &x);
    printf("Digite o segundo numero: ");
    scanf("%lf", &y);
    media = (x + y) / 2.0;
    printf("Media = %.1f\n", media);
    return 0;
}
```

```
using System;

namespace programa {
    class Program {
        static void Main(string[] args) {
            double x, y, media;

            Console.WriteLine("Digite o primeiro numero: ");
            x = double.Parse(Console.ReadLine());
            Console.WriteLine("Digite o segundo numero: ");
            y = double.Parse(Console.ReadLine());
            media = (x + y) / 2.0;
            Console.WriteLine("Media = " + media);
        }
    }
}
```

```
#include <iostream>
using namespace std;

int main() {
    double x, y, media;

    cout << "Digite o primeiro numero: ";
    cin >> x;
    cout << "Digite o segundo numero: ";
    cin >> y;
    media = (x + y) / 2.0;
    cout << "Media = " << media << endl;
    return 0;
}
```

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double x, y, media;

        System.out.print("Digite o primeiro numero: ");
        x = sc.nextDouble();
        System.out.print("Digite o segundo numero: ");
        y = sc.nextDouble();
        media = (x + y) / 2.0;
        System.out.println("Media = " + media);
        sc.close();
    }
}
```

---

---

---

---

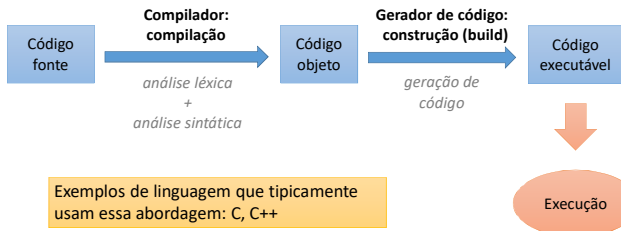
---

---

---

---

## Compilação



---

---

---

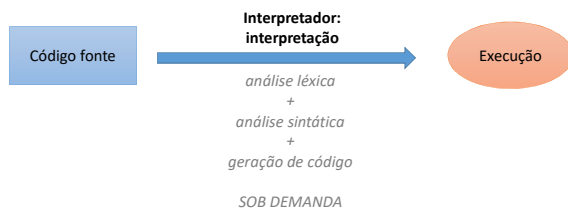
---

---

---

---

## Interpretação



---

---

---

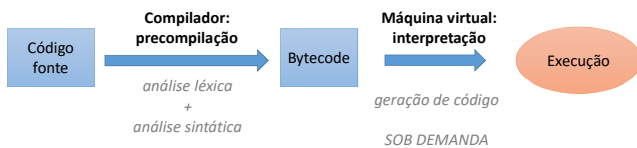
---

---

---

---

## Abordagem híbrida



---

---

---

---

---

---

---

## Vantagens

**Compilação:**

- velocidade do programa
- auxílio do compilador antes da execução

**Interpretação:**

- flexibilidade de manutenção do aplicativo em produção
- expressividade da linguagem
- código fonte não precisa ser recompilado para rodar em plataformas diferentes

**Abordagem híbrida**

---

---

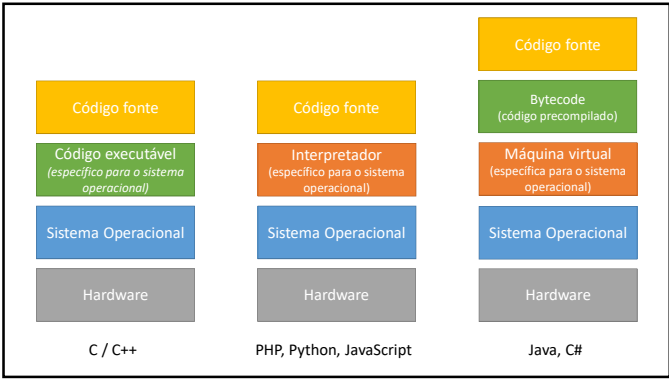
---

---

---

---

---



---

---

---

---

---

---

---

## Resumo da aula

- Tipos de código
  - Código fonte
  - Código objeto / bytecode
- Modelos de execução:
  - Compilação
    - Gerador de código
  - Interpretação
  - Abordagem híbrida
    - Máquina virtual

---

---

---

---

---

---

---

# Curso Java COMPLETO

Capítulo: Introdução à linguagem Java

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

Entendendo as versões do Java

<http://www.oracle.com/technetwork/java/javase>

**LTS** - Long Term Support

---

---

---

---

---

---

---

Java - contextualização

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## O que é Java?

- Linguagem de programação (regras sintáticas)
- Plataforma de desenvolvimento e execução
  - Bibliotecas (API)
  - Ambientes de execução

---

---

---

---

---

---

---

## Histórico

- Problemas resolvidos e motivo de seu sucesso:
  - Ponteiros / gerenciamento de memória
  - Portabilidade falha: reescrever parte do código ao mudar de SO
  - Utilização em dispositivos diversos
  - Custo
- Criada pela Sun Microsystems no meio da década de 1990
- Adquirida pela Oracle Corporation em 2010



---

---

---

---

---

---

---

## Aspectos notáveis

- Código compilado para bytecode e executado em máquina virtual (JVM)
- Portável, segura, robusta
- Roda em vários tipos de dispositivos
- Domina o mercado corporativo desde o fim do século 20
- Padrão Android por muitos anos



---

---

---

---

---

---

---

## Edições

- Java ME - Java Micro Edition - dispositivos embarcados e móveis - IoT
  - <http://www.oracle.com/technetwork/java/javame>
- Java SE - Java Standard Edition - core - desktop e servidores
  - <http://www.oracle.com/technetwork/java/javase>
- Java EE - Java Enterprise Edition - aplicações corporativas
  - <http://www.oracle.com/technetwork/java/javaee>

---

---

---

---

---

---

---

## Plataforma Java SE

- Documentação
  - <https://docs.oracle.com/en/java/javase/11/>

---

---

---

---

---

---

---

## Plataforma Java SE

- JVM - Java Virtual Machine
  - Máquina virtual do Java - necessário para executar sistemas Java

---

---

---

---

---

---

---

## Compilação e interpretação

- Linguagens **compiladas**: C, C++
- Linguagens **interpretadas**: PHP, JavaScript
- Linguagens **pré-compiladas + máquina virtual**: Java, C#

---

---

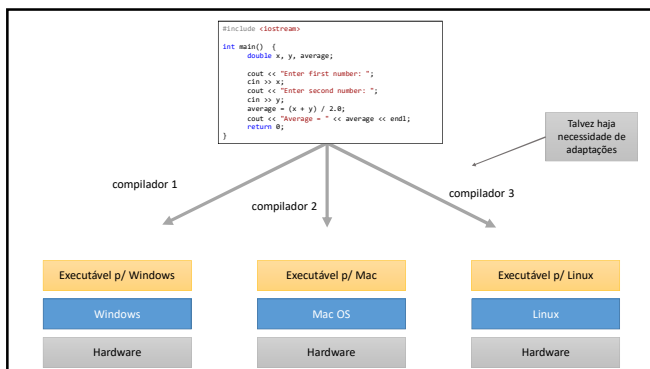
---

---

---

---

---



---

---

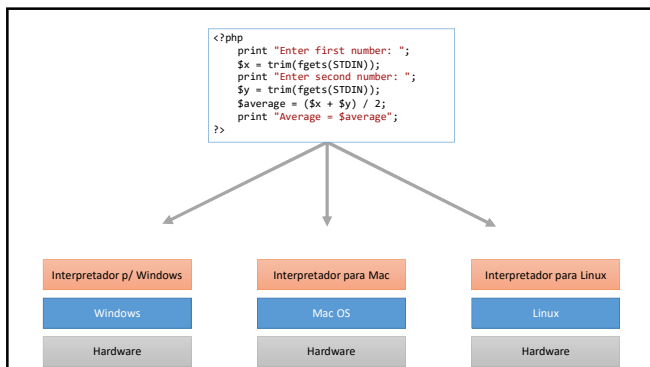
---

---

---

---

---



---

---

---

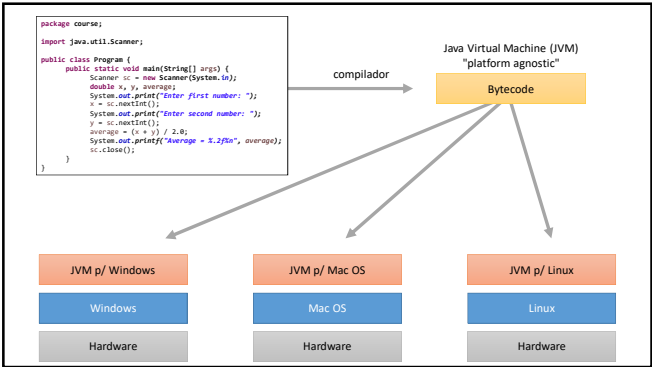
---

---

---

---





---

---

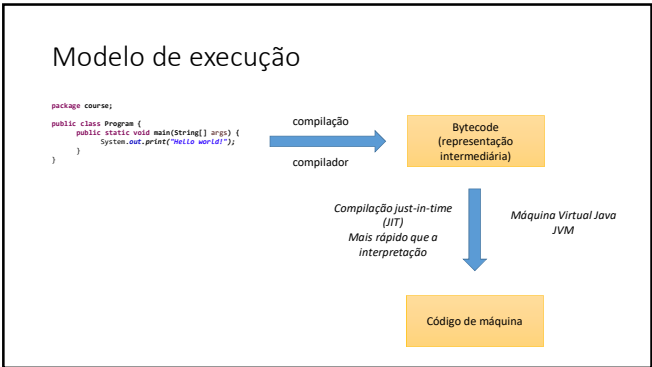
---

---

---

---

---



---

---

---

---

---

---

---

## Estrutura de uma aplicação Java

<http://educandoweb.com.br>  
Prof. Dr. Nelio Alves

---

---

---

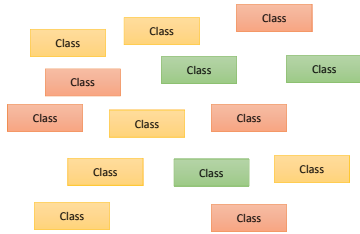
---

---

---

---

Uma aplicação é composta por classes



---

---

---

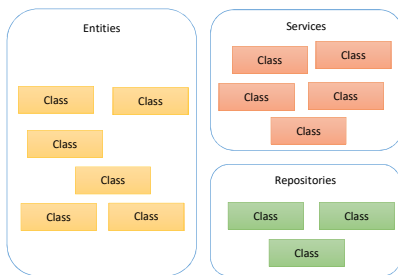
---

---

---

---

package = agrupamento LÓGICO de classes relacionadas



---

---

---

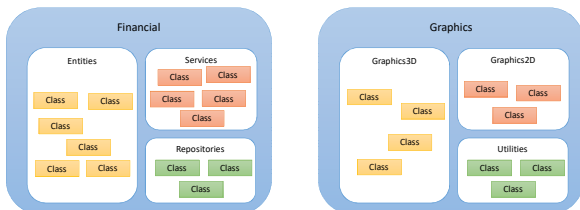
---

---

---

---

Módulo (Java 9+) = Agrupamento lógico de pacotes relacionados  
Runtime = Agrupamento físico



---

---

---

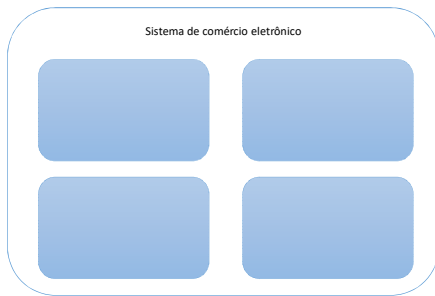
---

---

---

---

Aplicação = Agrupamento de módulo relacionados



---

---

---

---

---

---

---

## Instalação do Java e Eclipse no Windows

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Checklist

- Baixar e instalar o Java JDK
  - <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>
- Configurar variáveis de ambiente do sistema
  - Painel de Controle -> Variáveis de Ambiente
    - JAVA\_HOME:
      - C:\Program Files\Java\jdk-11.0.4
    - Path: **incluir**
      - C:\Program Files\Java\jdk-11.0.4\bin
  - Testar no terminal de comando: `java -version`
- Baixar e descompactar o Eclipse
  - <https://www.eclipse.org/downloads/packages/>
  - Testar: rodar o Eclipse e escolher um "workspace" (pasta onde você vai salvar seus projetos)

---

---

---

---

---

---

---

## Primeiro programa em Java e utilização básica do Eclipse

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

- Workspace (selecione a pasta aonde os projetos serão salvos)
- Mudar o layout: Window -> Perspective -> Open Perspective -> Java
- Zerar o layout: Window -> Perspective -> Reset Perspective
- Mostrar a aba Console: Window -> Show View -> Console
- Criar projeto: File -> New -> Java Project

- Criar classe:
  - Botão direito na pasta "src" -> New -> Class
  - Package: deixe em branco
  - Nome da classe: Main (com M maiúsculo)
  - Marque a opção: `public static void main(String[] args)`
- Mudar o tamanho da fonte:
  - CTRL +
  - CTRL -

# Curso Java COMPLETO

Capítulo: Estrutura sequencial

<http://educandoweb.com.br>

Prof. Dr. Nélcio Alves

---

---

---

---

---

---

---

## Expressões aritméticas

<http://educandoweb.com.br>

Prof. Dr. Nélcio Alves

---

---

---

---

---

---

---

## Expressões aritméticas



$$4 + 5 \xrightarrow{\text{resultado}} 9$$

---

---

---

---

---

---

---

## Operadores aritméticos

C, C++,  
Java, C# →

Operador	Significado
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão ("mod")

**Precedência:** 1º lugar: \* / %  
2º lugar: + -

---

---

---

---

---

---

---

## Exemplos de expressões aritméticas

$2 * 6 / 3$  Resultado = 4  
 $3 + 2 * 4$  Resultado = 11  
 $(3 + 2) * 4$  Resultado = 20  
 $60 / (3 + 2) * 4$  Resultado = 48  
 $60 / ((3 + 2) * 4)$  Resultado = 3

---

---

---

---

---

---

---

## Exemplos com o operador "mod"

$14 \% 3$  Resultado = 2  
 $19 \% 5$  Resultado = 4

Pois:

$$\begin{array}{r|l} 14 & 3 \\ \hline 2 & 4 \end{array}$$

$$\begin{array}{r|l} 19 & 5 \\ \hline 4 & 3 \end{array}$$

---

---

---

---

---

---

---

# Variáveis e tipos primitivos em Java

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Visão geral

- Um programa de computador em execução lida com dados
- Como esses dados são armazenados?
- Em **VARIÁVEIS!**

---

---

---

---

---

---

---

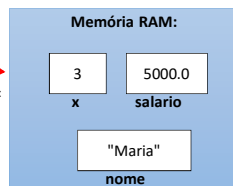
## Variáveis

### Definição informal:

Em programação, uma variável é uma porção de memória (RAM) utilizada para armazenar dados durante a execução dos programas.



Desenho  
esquemático:



---

---

---

---

---

---

---

Declaração de variáveis

Sintaxe:

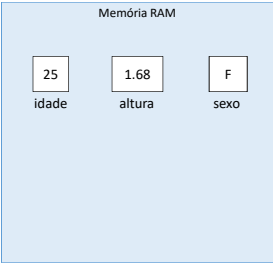
```
<tipo> <nome> = <valor inicial>;
                        (opcional)
```

Exemplos:

```
int idade = 25;
double altura = 1.68;
char sexo = 'F';
```

Uma variável possui:

- Nome (ou identificador)
- Tipo
- Valor
- Endereço



---

---

---

---

---

---

---

Tipos primitivos em Java

Descrição	Tipo	Tamanho	Valores	Valor padrão
tipos numéricos inteiros	byte	8 bits	-128 a 127	0
	short	16 bits	-32768 a 32767	0
	int	32 bits	-2147483648 a 2147483647	0
	long	64 bits	-9223372036854770000 a 9223372036854770000	0L
tipos numéricos com ponto flutuante	float	32 bits	-1,4024E-37 a 3,4028E+38	0.0f
	double	64 bits	-4,94E-307 a 1,79E+308	0.0
um caractere Unicode	char	16 bits	"\u0000" a "\uFFFF"	"\u0000"
valor verdade	boolean	1 bit	{false, true}	false

String - cadeia de caracteres (palavras ou textos)

Veja: [unicode-table.com](https://unicode-table.com)

Exemplo: 'a' = "\u0061"

---

---

---

---

---

---

---

Um bit pode armazenar 2 valores possíveis (0 ou 1)

Cada bit = 2 possibilidades

8 bits:

$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^8 = 256$  possibilidades

---

---

---

---

---

---

---



## Nomes de variáveis

- Não pode começar com dígito: use uma letra ou \_
- Não pode ter espaço em branco
- Não usar acentos ou til
- Sugestão: use o padrão "camel case"



Errado:

```
int 5minutos;  
int salário;  
int salário do funcionario;
```

Correto:

```
int _5minutos;  
int salario;  
int salarioDoFuncionario;
```

---

---

---

---

---

---

---

---

## Resumo da aula

- Conceito informal
- Declaração de variáveis: <tipo> <nome> = valor;
- Tipos primitivos:
  - **Números inteiros:** byte, short, int, long
  - **Números com ponto flutuante:** float, double
  - **Valor verdade:** boolean
  - **Um caractere Unicode:** char
- Tipo String: cadeia de caracteres (palavras, textos)
- Nomes de variáveis / padrão camel case

---

---

---

---

---

---

---

---

## As três operações básicas de programação

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

---

Um programa de computador é capaz de realizar essencialmente três operações:



---

---

---

---

---

---

---

## Entrada de dados

Usuário → Programa  
(dentro de variáveis)



Dispositivo de ENTRADA



Também chamada de  
LEITURA:

"O programa está lendo dados."

---

---

---

---

---

---

---

## Processamento de dados

É quando o programa realiza os cálculos



O processamento de dados se dá por um comando chamado ATRIBUIÇÃO

$media = (x + y) / 2.0;$

---

---

---

---

---

---

---

## Saída de dados

Programa → Usuário



Dispositivo de SAÍDA



Também chamada de  
ESCRITA:

"O programa está escrevendo dados."

---

---

---

---

---

---

---

## Saída de dados em Java

<http://educandoweb.com.br>

Prof. Dr. Nélio Alves

---

---

---

---

---

---

---

## Saída de dados

Programa → Usuário



Dispositivo de SAÍDA



Também chamada de  
ESCRITA:

"O programa está escrevendo dados."

---

---

---

---

---

---

---

Para escrever na tela um texto qualquer

**Sem quebra de linha ao final:**

```
System.out.print("Bom dia!");
```

**Com quebra de linha ao final:**

```
System.out.println("Bom dia!");
```

---

---

---

---

---

---

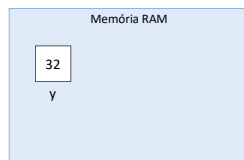
---

Para escrever o conteúdo de uma variável de algum tipo básico

Suponha uma variável tipo **int** declarada e iniciada:

```
int y = 32;
```

```
System.out.println(y);
```



---

---

---

---

---

---

---

Para escrever o conteúdo de uma variável com ponto flutuante

Suponha uma variável tipo **double** declarada e iniciada:

```
double x = 10.35784;
```

**%n** = quebra de linha  
(independente de  
plataforma)

```
System.out.println(x);
```

```
System.out.printf("%.2f%n", x);
```

```
System.out.printf("%.4f%n", x);
```

Localidade do  
sistema

**ATENÇÃO:**

Para considerar o separador de decimais como ponto, **ANTES** da declaração do Scanner, faça:

```
Locale.setDefault(Locale.US);
```

---

---

---

---

---

---

---

Para concatenar vários elementos em um mesmo comando de escrita

Regra geral para **print** e **println**:

elemento1 + elemento2 + elemento3 + ... + elementoN

```
System.out.println("RESULTADO = " + x + " METROS");
```

---

---

---

---

---

---

---

Para concatenar vários elementos em um mesmo comando de escrita

Regra geral para **printf**:

"TEXT01 %f TEXT02 %f TEXT03", variavel1, variavel2

%f = ponto flutuante

%n = quebra de linha

```
System.out.printf("RESULTADO = %.2f metros%n", x);
```

MAIS INFORMAÇÕES: <https://docs.oracle.com/javase/tutorial/java/data/numberformat.html>

---

---

---

---

---

---

---

Para concatenar vários elementos em um mesmo comando de escrita

Regra geral para **printf**:

"TEXT01 %f TEXT02 %f TEXT03", variavel1, variavel2

%f = ponto flutuante

%d = inteiro

%s = texto

%n = quebra de linha

```
String nome = "Maria";  
int idade = 31;  
double renda = 4000.0;  
System.out.printf("%s tem %d anos e ganha R$ %.2f reais%n", nome, idade, renda);
```

MAIS INFORMAÇÕES: <https://docs.oracle.com/javase/tutorial/java/data/numberformat.html>

---

---

---

---

---

---

---

## Resumo da aula

- `System.out.print`
- `System.out.println`
- `System.out.printf`
  - `%d`
  - `%f`
  - `%s`
  - `%n`
- `Locale`
- Como concatenar vários elementos em um mesmo comando de escrita
- Exemplos

---

---

---

---

---

---

---

---

## Exercício de fixação

Em um novo programa, inicie as seguintes variáveis:

```
String product1 = "Computer";
String product2 = "Office desk";

int age = 30;
int code = 5290;
char gender = 'F';

double price1 = 2100.0;
double price2 = 650.50;
double measure = 53.234567;
```

Em seguida, usando os valores das variáveis, produza a seguinte saída na tela do console:

```
Products:
Computer, which price is $ 2100,00
Office desk, which price is $ 650,50

Record: 30 years old, code 5290 and gender: F

Measure with eight decimal places: 53,23456700
Rounded (three decimal places): 53,235
US decimal point: 53.235
```

(correção na próxima página)

---

---

---

---

---

---

---

---

```
import java.util.Locale;

public class Main {

    public static void main(String[] args) {

        String product1 = "Computer";
        String product2 = "Office desk";

        byte age = 30;
        int code = 5290;
        char gender = 'F';

        double price1 = 2100.0;
        double price2 = 650.50;
        double measure = 53.234567;

        System.out.println("Products:");
        System.out.printf("%s, which price is $ %.2f\n", product1, price1);
        System.out.printf("%s, which price is $ %.2f\n", product2, price2);
        System.out.println();
        System.out.printf("Record: %d years old, code %d and gender: %c\n", age, code, gender);
        System.out.println();
        System.out.printf("Measure with eight decimal places: %.8f\n", measure);
        System.out.printf("Rounded (three decimal places): %.3f\n", measure);
        Locale.setDefault(Locale.US);
        System.out.printf("US decimal point: %.3f\n", measure);

    }
}
```

---

---

---

---

---

---

---

---

• Comentários de linha:

- Começam com //

• Atalhos:

- Importar classes: CTRL + SHIFT + O
- Autoindentação: CTRL + SHIFT + F
- sysout CTRL + espaço

---

---

---

---

---

---

---

## Processamento de dados em Java, Casting

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Processamento de dados

Comando de atribuição.

**Sintaxe:**

**<variável> = <expressão>;**

Lê-se "recebe"

**REGRA:**

- 1) A expressão é calculada
- 2) O resultado da expressão é armazenado na variável

---

---

---

---

---

---

---

### Exemplo 1

```
int x, y;  
  
x = 5;  
  
y = 2 * x;  
  
System.out.println(x);  
System.out.println(y);
```

---

---

---

---

---

---

---

### Exemplo 2

```
int x;  
double y;  
  
x = 5;  
  
y = 2 * x;  
  
System.out.println(x);  
System.out.println(y);
```

---

---

---

---

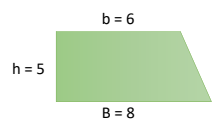
---

---

---

### Exemplo 3

```
double b, B, h, area;  
  
b = 6.0;  
B = 8.0;  
h = 5.0;  
  
area = (b + B) / 2.0 * h;  
  
System.out.println(area);
```



$$area = \frac{(b + B)}{2} \times h$$

No exemplo:

$$area = \frac{(6 + 8)}{2} \times 5 \\ = \frac{14}{2} \times 5 = 7 \times 5 = 35$$

---

---

---

---

---

---

---



```
double b, B, h, area;

b = 6.0;
B = 8.0;
h = 5.0;

area = (b + B) / 2.0 * h;

System.out.println(area);
```

Boa prática:

Sempre indique o tipo do número, se a expressão for de ponto flutuante (não inteira).

Para **double** use:  
.0

Para **float** use:  
f

---

---

---

---

---

---

---

```
float b, B, h, area;

b = 6f;
B = 8f;
h = 5f;

area = (b + B) / 2f * h;

System.out.println(area);
```

Boa prática:

Sempre indique o tipo do número, se a expressão for de ponto flutuante (não inteira).

Para **double** use:  
.0

Para **float** use:  
f

---

---

---

---

---

---

---

## Exemplo 4

```
int a, b;
double resultado;

a = 5;
b = 2;

resultado = a / b;

System.out.println(resultado);
```

---

---

---

---

---

---

---

## Casting

É a conversão explícita de um tipo para outro.

É necessário quando o compilador não é capaz de “adivinhar” que o resultado de uma expressão deve ser de outro tipo.

---

---

---

---

---

---

---

## Exemplo 4

```
int a, b;  
double resultado;  
  
a = 5;  
b = 2;  
  
resultado = a / b;  
  
System.out.println(resultado);
```

---

---

---

---

---

---

---

## Exemplo 4

```
int a, b;  
double resultado;  
  
a = 5;  
b = 2;  
  
resultado = (double) a / b;  
  
System.out.println(resultado);
```

---

---

---

---

---

---

---

### Exemplo 5

```
double a;  
int b;  
  
a = 5.0;  
b = a;  
  
System.out.println(b);
```

---

---

---

---

---

---

---

## Entrada de dados em Java

<http://educandoweb.com.br>

Prof. Dr. Nélio Alves

---

---

---

---

---

---

---

## Entrada de dados

Usuário → Programa  
(dentro de variáveis)



Dispositivo de ENTRADA



Também chamada de  
LEITURA:

"O programa está lendo dados."

---

---

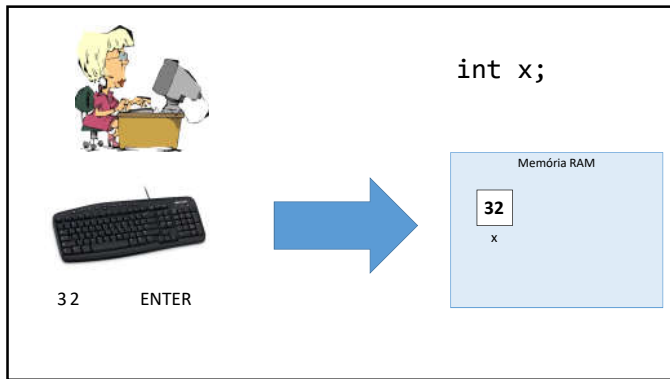
---

---

---

---

---



---

---

---

---

---

---

---

### Scanner

Para fazer entrada de dados, nós vamos criar um objeto do tipo "Scanner" da seguinte forma:

```
Scanner sc = new Scanner(System.in);
```

import java.util.Scanner;

faça sc.close() quando não precisar mais do objeto sc

---

---

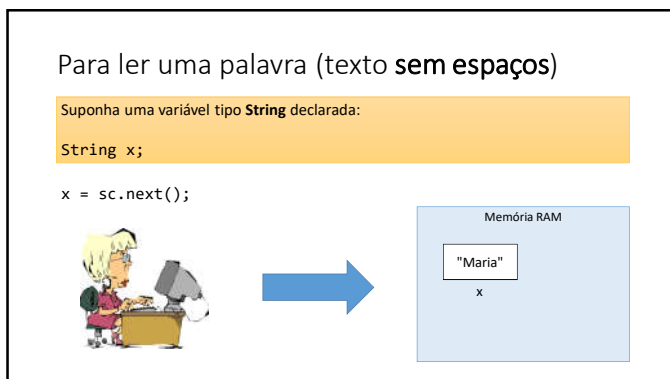
---

---

---

---

---



---

---

---

---

---

---

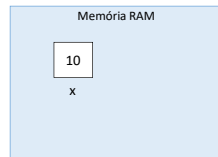
---

## Para ler um número inteiro

Suponha uma variável tipo **int** declarada:

```
int x;
```

```
x = sc.nextInt();
```



---

---

---

---

---

---

---

## Para ler um número com ponto flutuante

Suponha uma variável tipo **double** declarada:

```
double x;
```

```
x = sc.nextDouble();
```

← Localidade do sistema

### ATENÇÃO:

Para considerar o separador de decimais como ponto, **ANTES** da declaração do Scanner, faça:

```
Locale.setDefault(Locale.US);
```

---

---

---

---

---

---

---

## Para ler um caractere

Suponha uma variável tipo **char** declarada:

```
char x;
```

```
x = sc.next().charAt(0);
```

---

---

---

---

---

---

---

Para ler vários dados na mesma linha

```
string x;  
int y;  
double z;  
  
x = sc.next();  
y = sc.nextInt();  
z = sc.nextDouble();
```

---

---

---

---

---

---

---

Para ler um texto ATÉ A QUEBRA DE LINHA

```
import java.util.Scanner;  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        String s1, s2, s3;  
  
        s1 = sc.nextLine();  
        s2 = sc.nextLine();  
        s3 = sc.nextLine();  
  
        System.out.println("DADOS DIGITADOS:");  
        System.out.println(s1);  
        System.out.println(s2);  
        System.out.println(s3);  
  
        sc.close();  
    }  
}
```

---

---

---

---

---

---

---

ATENÇÃO: quebra de linha pendente

Quando você usa um comando de leitura diferente do `nextLine()` e dá alguma quebra de linha, essa quebra de linha fica "pendente" na entrada padrão.

Se você então fizer um `nextLine()`, aquela quebra de linha pendente será absorvida pelo `nextLine()`.

Solução:

Faça um `nextLine()` extra antes de fazer o `nextLine()` de seu interesse.

```
int x;  
String s1, s2, s3;  
  
x = sc.nextInt();  
s1 = sc.nextLine();  
s2 = sc.nextLine();  
s3 = sc.nextLine();  
  
System.out.println("DADOS DIGITADOS:");  
System.out.println(x);  
System.out.println(s1);  
System.out.println(s2);  
System.out.println(s3);
```

---

---

---

---

---

---

---

## Resumo da aula

- Scanner
  - next()
  - nextInt()
  - nextDouble()
  - next().charAt(0)
- Locale
- Como ler até a quebra de linha
  - nextLine()
  - como limpar o buffer de leitura

---

---

---

---

---

---

---

## Funções matemáticas em Java

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Algumas funções matemáticas em Java

Exemplo	Significado
<code>A = Math.sqrt(x);</code>	Variável A recebe a raiz quadrada de x
<code>A = Math.pow(x, y);</code>	Variável A recebe o resultado de x elevado a y
<code>A = Math.abs(x);</code>	Variável A recebe o valor absoluto de x

---

---

---

---

---

---

---

```

public class Main {
    public static void main(String[] args) {

        double x = 3.0;
        double y = 4.0;
        double z = -5.0;
        double A, B, C;

        A = Math.sqrt(x);
        B = Math.sqrt(y);
        C = Math.sqrt(25.0);
        System.out.println("Raiz quadrada de " + x + " = " + A);
        System.out.println("Raiz quadrada de " + y + " = " + B);
        System.out.println("Raiz quadrada de 25 = " + C);

        A = Math.pow(x, y);
        B = Math.pow(x, 2.0);
        C = Math.pow(5.0, 2.0);
        System.out.println(x + " elevado a " + y + " = " + A);
        System.out.println(x + " elevado ao quadrado = " + B);
        System.out.println("5 elevado ao quadrado = " + C);

        A = Math.abs(y);
        B = Math.abs(z);
        System.out.println("Valor absoluto de " + y + " = " + A);
        System.out.println("Valor absoluto de " + z + " = " + B);
    }
}

```

---

---

---

---

---

---

---

---

Incluindo funções em expressões maiores

$$x = \frac{-b \pm \sqrt{\Delta}}{2.a}$$

$$\Delta = b^2 - 4ac$$

```
delta = Math.pow(b, 2.0) - 4*a*c;
```

```
x1 = (-b + Math.sqrt(delta)) / (2.0 * a);
x2 = (-b - Math.sqrt(delta)) / (2.0 * a);
```

---

---

---

---

---

---

---

---

## Funções matemáticas

- sqrt – raiz quadrada
- pow – potenciação
- abs – valor absoluto
- Exemplos

Maiores informações: java.lang.Math

---

---

---

---

---

---

---

---



# Curso Java COMPLETO

Capítulo: Estrutura condicional

<http://educandoweb.com.br>

Prof. Dr. Nélcio Alves

---

---

---

---

---

---

---

## Expressões comparativas

<http://educandoweb.com.br>

Prof. Dr. Nélcio Alves

---

---

---

---

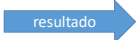
---

---

---

## Expressões comparativas



5 > 10  Falso

---

---

---

---

---

---

---

## Operadores comparativos

C, C++,  
Java, C# →

Operador	Significado
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual
==	igual
!=	diferente

---

---

---

---

---

---

---

## Exemplos de expressões comparativas

(suponha x igual a 5)

X > 0	Resultado: V
X == 3	Resultado: F
10 <= 30	Resultado: V
X != 2	Resultado: V

---

---

---

---

---

---

---

## Expressões lógicas

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Expressões lógicas



---

---

---

---

---

---

---

## Operadores lógicos

C, C++,  
Java, C# →

Operador	Significado
&&	E
	OU
!	NÃO

---

---

---

---

---

---

---

## Ideia por trás do operador "E"

Você pode obter uma habilitação de motorista se:

- For aprovado no exame psicotécnico,  
E
- For aprovado no exame de legislação,  
E
- For aprovado no exame de direção

**Todas condições  
devem ser  
verdadeiras!**

---

---

---

---

---

---

---

## Exemplos de expressões lógicas

(suponha x igual a 5)

$X \leq 20 \ \&\& \ X == 10$       Resultado: F  
V      F

$X > 0 \ \&\& \ X != 3$       Resultado: V  
V      V

$X \leq 20 \ \&\& \ X == 10 \ \&\& \ X != 3$       Resultado: F  
V      F      V

---

---

---

---

---

---

---

## Tabela verdade do operador "E"

A	B	A && B
F	F	F
F	V	F
V	F	F
V	V	V

---

---

---

---

---

---

---

## Ideia por trás do operador "OU"

Você pode estacionar na vaga especial se:

- For idoso(a),
- OU**
- For uma pessoa com deficiência,
- OU**
- For uma gestante

**Pelo menos uma  
condição deve  
ser verdadeira!**

---

---

---

---

---

---

---

## Exemplos de expressões lógicas

(suponha x igual a 5)

$X == 10 \ || \ X \leq 20$  Resultado: V

F

V

$X > 0 \ || \ X \neq 3$  Resultado: V

V

V

$X \leq 0 \ || \ X \neq 3 \ || \ X \neq 5$  Resultado: V

F

V

F

## Tabela verdade do operador "OU"

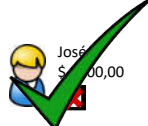
A	B	A    B
F	F	F
F	V	V
V	F	V
V	V	V

## Ideia por trás do operador "NÃO"

Você tem direito a receber uma bolsa de estudos se você:

### NÃO

- Possuir renda maior que \$ 3000,00



O operador  
"NÃO" inverte a  
condição

### Exemplos de expressões lógicas

(suponha x igual a 5)

!(X == 10)  
F

Resultado: V

!(X >= 2)  
V

Resultado: F

---

---

---

---

---

---

---

### Exemplos de expressões lógicas

(suponha x igual a 5)

!(X <= 20 && X == 10)  
V F  
F  
V

Resultado: V

---

---

---

---

---

---

---

### Tabela verdade do operador "NÃO"

A	!A
F	V
V	F

---

---

---

---

---

---

---

# Estrutura condicional

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

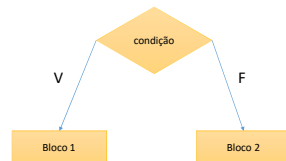
---

---

## Conceito

Estrutura condicional:

É uma **estrutura de controle** que permite definir que um certo **bloco de comandos** somente será executado dependendo de uma **condição**



---

---

---

---

---

---

---

## Sintaxe da estrutura condicional

### Simples:

```
if ( <condição> ) {  
  <comando 1>  
  <comando 2>  
}
```

### REGRA:

V: executa o bloco de comandos  
F: pula o bloco de comandos

Importante:  
Repare na indentação!

---

---

---

---

---

---

---

## Sintaxe da estrutura condicional

### Composta:

```
if ( <condição> ) {  
  <comando 1>  
  <comando 2>  
}  
else {  
  <comando 3>  
  <comando 4>  
}
```

### REGRA:

V: executa somente o bloco do if  
F: executa somente o bloco do else

Importante:  
Repare na indentação!

---

---

---

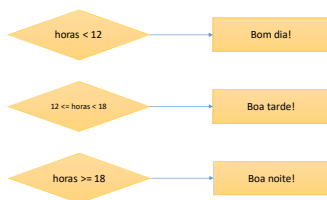
---

---

---

---

E se eu tiver mais de duas possibilidades?



---

---

---

---

---

---

---

## Encadeamento de estruturas condicionais

```
if ( condição 1 ) {  
  comando 1  
  comando 2  
}  
else {  
  if ( condição 2 ) {  
    comando 3  
    comando 4  
  }  
  else {  
    comando 5  
    comando 6  
  }  
}
```

Importante:  
Repare na indentação!

---

---

---

---

---

---

---



## Encadeamento de estruturas condicionais

```
if ( condição 1 ) {  
    comando 1  
    comando 2  
}  
else if ( condição 2 ) {  
    comando 3  
    comando 4  
}  
else if ( condição 3 ) {  
    comando 5  
    comando 6  
}  
else {  
    comando 7  
    comando 8  
}
```

Importante:  
Repare na indentação!

---

---

---

---

---

---

---

## Sintaxe opcional: operadores de atribuição cumulativa

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Problema exemplo

Uma operadora de telefonia cobra R\$ 50.00 por um plano básico que dá direito a 100 minutos de telefone. Cada minuto que exceder a franquia de 100 minutos custa R\$ 2.00. Fazer um programa para ler a quantidade de minutos que uma pessoa consumiu, daí mostrar o valor a ser pago.

Entrada	Saída
22	Valor a pagar: R\$ 50.00

Entrada	Saída
103	Valor a pagar: R\$ 56.00

---

---

---

---

---

---

---

## Operadores de atribuição cumulativa

a += b;	a = a + b;
a -= b;	a = a - b;
a *= b;	a = a * b;
a /= b;	a = a / b;
a %= b;	a = a % b;

---

---

---

---

---

---

---

---

```
import java.util.Locale;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        int minutos = sc.nextInt();

        double conta = 50.0;
        if (minutos > 100) {
            conta += (minutos - 100) * 2.0;
        }

        System.out.printf("Valor da conta = R$ %.2f\n", conta);

        sc.close();
    }
}
```

---

---

---

---

---

---

---

---

## Sintaxe opcional: estrutura switch-case

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

---

Estrutura switch-case

Quando se tem várias opções de fluxo a serem tratadas com base no valor de uma variável, ao invés de várias estruturas if-else encadeadas, alguns preferem utilizar a estrutura switch-case.

---

---

---

---

---

---

---

Problema exemplo

Fazer um programa para ler um valor inteiro de 1 a 7 representando um dia da semana (sendo 1=domingo, 2=segunda, e assim por diante). Escrever na tela o dia da semana correspondente, conforme exemplos.

Entrada	Saída
1	Dia da semana: domingo
Entrada	Saída
4	Dia da semana: quarta
Entrada	Saída
9	Dia da semana: valor inválido

---

---

---

---

---

---

---

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int x = sc.nextInt();
        String dia;

        if (x == 1) {
            dia = "domingo";
        }
        else if (x == 2) {
            dia = "segunda";
        }
        else if (x == 3) {
            dia = "terça";
        }
        else if (x == 4) {
            dia = "quarta";
        }
        else if (x == 5) {
            dia = "quinta";
        }
        else if (x == 6) {
            dia = "sexta";
        }
        else if (x == 7) {
            dia = "sabado";
        }
        else {
            dia = "valor invalido";
        }

        System.out.println("Dia da semana: " + dia);
        sc.close();
    }
}
```

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int x = sc.nextInt();
        String dia;

        switch (x) {
            case 1:
                dia = "domingo";
                break;
            case 2:
                dia = "segunda";
                break;
            case 3:
                dia = "terça";
                break;
            case 4:
                dia = "quarta";
                break;
            case 5:
                dia = "quinta";
                break;
            case 6:
                dia = "sexta";
                break;
            case 7:
                dia = "sabado";
                break;
            default:
                dia = "valor invalido";
                break;
        }

        System.out.println("Dia da semana: " + dia);
        sc.close();
    }
}
```

---

---

---

---

---

---

---

## Sintaxe do switch-case

```
switch ( expressão ) {  
  case valor1:  
    comando1  
    comando2  
    break;  
  case valor2:  
    comando3  
    comando4  
    break;  
  
  default:  
    comando5  
    comando6  
    break;  
}
```

---

---

---

---

---

---

---

## Expressão condicional ternária

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Expressão condicional ternária

Estrutura opcional ao if-else quando se deseja decidir um **VALOR** com base em uma condição.

### Sintaxe:

```
( condição ) ? valor_se_verdadeiro : valor_se_falso
```

### Exemplos:

```
( 2 > 4 ) ? 50 : 80      ➡ 80
```

```
( 10 != 3 ) ? "Maria" : "Alex"      ➡ "Maria"
```

---

---

---

---

---

---

---

## Demo

```
double preco = 34.5;
double desconto;
if (preco < 20.0) {
    desconto = preco * 0.1;
}
else {
    desconto = preco * 0.05;
}
```

```
double preco = 34.5;
double desconto = (preco < 20.0) ? preco * 0.1 : preco * 0.05;
```

---

---

---

---

---

---

---

---

## Escopo e inicialização

<http://educandoweb.com.br>

Prof. Dr. Nélio Alves

---

---

---

---

---

---

---

---

## Checklist

- Escopo de uma variável: é a região do programa onde a variável é válida, ou seja, onde ela pode ser referenciada.
- Uma variável não pode ser usada se não for iniciada.
- Falaremos de escopo de métodos no Capítulo 5

---

---

---

---

---

---

---

---

## Demo

```
double price = sc.nextDouble();  
  
if (price > 100.0) {  
    double discount = price * 0.1;  
}  
  
System.out.println(discount);
```

---

---

---

---

---

---

---

# Curso Java COMPLETO

Capítulo: Estruturas repetitivas

<http://educandoweb.com.br>

Prof. Dr. Nélcio Alves

---

---

---

---

---

---

---

## Como utilizar o Debug no Eclipse (execução passo a passo)

<http://educandoweb.com.br>

Prof. Dr. Nélcio Alves

---

---

---

---

---

---

---

## Como executar o debug do Eclipse

- Para marcar uma linha de breakpoint:
  - Run -> Toggle Breakpoint
- Para iniciar o debug:
  - Botão direito na classe -> Debug as -> Java Application
- Para executar uma linha:
  - F6
- Para interromper o debug:



---

---

---

---

---

---

---

```
import java.util.Locale;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        double largura = sc.nextDouble();
        double comprimento = sc.nextDouble();
        double metroQuadrado = sc.nextDouble();

        double area = largura * comprimento;
        double preco = area * metroQuadrado;

        System.out.printf("AREA = %.2f\n", area);
        System.out.printf("PRECO = %.2f\n", preco);

        sc.close();
    }
}
```

---

---

---

---

---

---

---

---

## Estrutura repetitiva "enquanto" (while)

<http://educandoweb.com.br>

Prof. Dr. Nélio Alves

---

---

---

---

---

---

---

---

## Estrutura repetitiva "enquanto"

É uma **estrutura de controle** que **repete** um bloco de comandos **enquanto** uma **condição** for verdadeira.

**Quando usar:** quando **não** se sabe previamente a quantidade de repetições que será realizada.

### Problema exemplo:

Fazer um programa que lê números inteiros até que um zero seja lido. Ao final mostra a soma dos números lidos.

Entrada	Saída
5	11
2	
4	
0	

---

---

---

---

---

---

---

---



## Sintaxe / regra

```
while ( condição ) {  
    comando 1  
    comando 2  
}
```

### Regra:

**V: executa e volta**  
**F: pula fora**

---

---

---

---

---

---

---

## Resumo da aula

- Estrutura repetitiva "enquanto"
- Recomendada quando não se sabe previamente a quantidade de repetições
- Regra:
  - V: executa e volta
  - F: pula fora

---

---

---

---

---

---

---

## Exercício de testes de mesa com while

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

<pre>x = 5; y = 0; while (x &gt; 2) {     System.out.print(x);     y = y + x;     x = x - 1; }</pre>	<pre>x = 2; y = 0; while (x &lt; 60) {     System.out.println(x);     x = x * 2;     y = y + 10; }</pre>	<pre>x = 100; y = 100; while (x != y) {     System.out.print("olha");     x = Math.sqrt(y); }</pre>
<div><div></div><div></div><div></div></div> <div><b>x</b><b>y</b><b>i</b></div> <div>Tela:</div> <div></div>	<div><div></div><div></div><div></div></div> <div><b>x</b><b>y</b><b>i</b></div> <div>Tela:</div> <div></div>	<div><div></div><div></div><div></div></div> <div><b>x</b><b>y</b><b>i</b></div> <div>Tela:</div> <div></div>

---

---

---

---

---

---

---

---

<pre>x = 0; while (x &lt; 5) {     y = x + 3;     System.out.print(y);     x = x + 1; } System.out.println("Fim");</pre>	<pre>x = 2; y = 10; System.out.println("Olá"); while (x &lt; y) {     System.out.println(x + "-" + y);     x = x * 2;     y = y + 1; }</pre>	<pre>x = 4; y = 0; i = 0; while (i &lt; x) {     i = i + 1;     y = y + i;     System.out.print(i);     System.out.println(y); }</pre>
<div><div></div><div></div><div></div></div> <div><b>x</b><b>y</b><b>i</b></div> <div>Tela:</div> <div></div>	<div><div></div><div></div><div></div></div> <div><b>x</b><b>y</b><b>i</b></div> <div>Tela:</div> <div></div>	<div><div></div><div></div><div></div></div> <div><b>x</b><b>y</b><b>i</b></div> <div>Tela:</div> <div></div>

---

---

---

---

---

---

---

---

<pre>x = 5; y = 0; while (x &gt; 2) {     System.out.print(x);     y = y + x;     x = x - 1; }</pre>	<pre>x = 2; y = 0; while (x &lt; 60) {     System.out.println(x);     x = x * 2;     y = y + 10; }</pre>	<pre>x = 100; y = 100; while (x != y) {     System.out.print("olha");     x = Math.sqrt(y); }</pre>
<div><div>5432</div><div>05812</div><div></div></div> <div><b>x</b><b>y</b><b>i</b></div> <div>Tela:</div> <div>543</div>	<div><div>248</div><div>102020</div><div></div></div> <div><b>x</b><b>y</b><b>i</b></div> <div>Tela:</div> <div>2481632</div>	<div><div>100</div><div>100</div><div></div></div> <div><b>x</b><b>y</b><b>i</b></div> <div>Tela:</div> <div></div>

---

---

---

---

---

---

---

---

```

x = 0;
while (x < 5) {
    y = x * 3;
    System.out.println(y);
    x = x + 1;
}
System.out.println("Fim");

```

0 3 6	3 6 9	
x	y	i

Tela:

0 3 6 9 12 Fim

```

x = 2;
y = 10;
System.out.println("Olá");
while (x < y) {
    System.out.println(x + "-" + y);
    x = x * 2;
    y = y + 1;
}

```

2 4 8	10 15 20	
x	y	i

Tela:

0 1 2  
2 - 10  
4 - 15  
8 - 20

```

x = 4;
y = 0;
i = 0;
while (i < x) {
    i = i + 1;
    y = y + 1;
    System.out.print(i);
    System.out.println(y);
}

```

4	0 1 2 3	0 1 2 3
x	y	i

Tela:

1 1  
2 3  
3 6  
4 10

---

---

---

---

---

---

---

---

# Estrutura repetitiva "para" (for)

<http://educandoweb.com.br>

Prof. Dr. Nélio Alves

---

---

---

---

---

---

---

---

# Estrutura repetitiva "para"

É uma **estrutura de controle** que **repete** um bloco de comandos **para** um certo **intervalo de valores**.

Quando usar: quando se sabe previamente a quantidade de repetições, ou o intervalo de valores.

**Por exemplo:**

Fazer um programa que lê um valor inteiro N e depois N números inteiros. Ao final, mostra a soma dos N números lidos

Entrada	Saída
3	11
5	
2	
4	

---

---

---

---

---

---

---

---

## Sintaxe / regra

Executa somente  
na primeira vez

V: executa e volta  
F: pula fora

Executa toda vez depois  
de voltar

```
for ( início ; condição ; incremento ) {  
    comando 1  
    comando 2  
}
```

## Importante

Perceba que a estrutura "para" é ótima para se fazer uma repetição baseada em uma CONTAGEM:

```
for (int i=0; i<5; i++) {  
    System.out.println("Valor de i: " + i);  
}
```

Resultado na tela:

```
Valor de i: 0  
Valor de i: 1  
Valor de i: 2  
Valor de i: 3  
Valor de i: 4
```

## Contagem regressiva

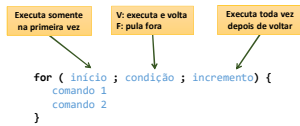
```
for (int i=4; i>=0; i--) {  
    System.out.println("Valor de i: " + i);  
}
```

Resultado na tela:

```
Valor de i: 4  
Valor de i: 3  
Valor de i: 2  
Valor de i: 1  
Valor de i: 0
```

## Resumo da aula

- Estrutura repetitiva "para"
- Usar quando se sabe previamente a quantidade de repetições
- Ótimo para fazer contagens (progressiva ou regressiva)
- Regra:



## Exercício de testes de mesa com for

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

<pre>x = 4; y = x + 2; for (i=0; i&lt;x; i++) {     System.out.print(x+" "+y);     y = y + i; }</pre>	<pre>for (i=1; i&lt;5; i++){     y = i - 1;     x = i * 10;     System.out.print(i); }</pre>	<pre>y = 10; for (i=0; i&lt;4; i++){     System.out.print(i);     y = y + i;     System.out.println(y); }</pre>																		
<table border="1"><tr><td></td><td></td><td></td></tr><tr><td>x</td><td>y</td><td>i</td></tr></table>				x	y	i	<table border="1"><tr><td></td><td></td><td></td></tr><tr><td>x</td><td>y</td><td>i</td></tr></table>				x	y	i	<table border="1"><tr><td></td><td></td><td></td></tr><tr><td>x</td><td>y</td><td>i</td></tr></table>				x	y	i
x	y	i																		
x	y	i																		
x	y	i																		
Tela:	Tela:	Tela:																		
<div></div>	<div></div>	<div></div>																		

```
x = 4;
y = 0;
for (i=0; i<x; i++) {
    System.out.print(i);
    System.out.println(x);
    y = y + 10;
}
```

**x****y****i**

Tela:

```
x = 4;
y = 0;
for (i=0; i<x; i++) {
    y = y + i;
}
System.out.println(y);
```

**x****y****i**

Tela:

```
x = 8;
y = 3;
for (i=0; y<x; i++){
    x = x - 2;
    y = y + 1;
    System.out.println(i);
}
```

**x****y****i**

Tela:

---

---

---

---

---

---

---

---

```
x = 4;
y = x + 2;
for (i=0; i<x; i++) {
    System.out.print(x+" "+y);
    y = y + i;
}
```

4

0 8 2  
3 12 3 4

0 1 2  
3 4

**x****y****i**

Tela:

4 6 4 6 4 7 4 9

```
for (i=1; i<5; i++){
    y = i - 1;
    x = i * 10;
    System.out.print(i);
}
```

10 20 30  
40

0 1 2  
3

1 2 3  
4 5

**x****y****i**

Tela:

1 2 3 4

```
y = 10;
for (i=0; i<4; i++){
    System.out.print(i);
    y = y + i;
    System.out.println(y);
}
```

10 20 30  
34 16 3 4

0 1 2  
3 4

**x****y****i**

Tela:

0 10  
1 11  
2 13  
3 16

---

---

---

---

---

---

---

---

```
x = 4;
y = 0;
for (i=0; i<x; i++) {
    System.out.print(i);
    System.out.println(x);
    y = y + 10;
}
```

4

0 30 20  
30 40 3 4

0 1 2  
3 4

**x****y****i**

Tela:

0 4  
1 4  
2 4  
3 4

```
x = 4;
y = 0;
for (i=0; i<x; i++) {
    y = y + i;
}
System.out.println(y);
```

4

0 0 1  
3 6

0 1 2  
3 4

**x****y****i**

Tela:

6

```
x = 8;
y = 3;
for (i=0; y<x; i++){
    x = x - 2;
    y = y + 1;
    System.out.println(i);
}
```

8 0 4  
3 4 5

0 1 2  
3 4

0 1 2  
3 4

**x****y****i**

Tela:

0  
1

---

---

---

---

---

---

---

---

## Estrutura repetitiva "faça-enquanto"

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Estrutura repetitiva "faça-enquanto"

Menos utilizada, mas em alguns casos se encaixa melhor ao problema.

O bloco de comandos executa pelo menos uma vez, pois a condição é verificada no final.

---

---

---

---

---

---

---

## Sintaxe / regra

```
do {  
  comando 1  
  comando 2  
} while ( condição );
```

**Regra:**

**V:** volta

**F:** pula fora

---

---

---

---

---

---

---

Problema exemplo:

Fazer um programa para ler uma temperatura em Celsius e mostrar o equivalente em Fahrenheit. Perguntar se o usuário deseja repetir (s/n). Caso o usuário digite "s", repetir o programa.

Fórmula:  $F = \frac{9C}{5} + 32$

Exemplo:

Digite a temperatura em Celsius: 30.0  
Equivalente em Fahrenheit: 86.0  
Deseja repetir (s/n)? s  
Digite a temperatura em Celsius: 21.0  
Equivalente em Fahrenheit: 69.8  
Deseja repetir (s/n)? s  
Digite a temperatura em Celsius: -10.5  
Equivalente em Fahrenheit: 13.1  
Deseja repetir (s/n)? n

---

---

---

---

---

---

---

---

Resumo da aula

- Estrutura repetitiva "faça-enquanto"
  - O bloco de comandos executa pelo menos uma vez, pois a condição é verificada no final.
  - Regra:
    - V: volta
    - F: pula fora
- ```
do {  
    comando 1  
    comando 2  
} while ( condição );
```

---

---

---

---

---

---

---

---

```
import java.util.Locale;  
import java.util.Scanner;  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        Locale.setDefault(Locale.US);  
        Scanner sc = new Scanner(System.in);  
  
        char resp;  
        do {  
            System.out.print("Digite a temperatura em Celsius: ");  
            double C = sc.nextDouble();  
            double F = 9.0 * C / 5.0 + 32.0;  
            System.out.printf("Equivalente em Fahrenheit: %.1f\n", F);  
            System.out.print("Deseja repetir (s/n)? ");  
            resp = sc.next().charAt(0);  
        } while (resp != 'n');  
  
        sc.close();  
    }  
}
```

---

---

---

---

---

---

---

---



# Curso Java COMPLETO

Capítulo: Outros tópicos básicos sobre Java

<http://educandoweb.com.br>

Prof. Dr. Nélcio Alves

---

---

---

---

---

---

---

## Restrições e convenções para nomes

<http://educandoweb.com.br>

Prof. Dr. Nélcio Alves

---

---

---

---

---

---

---

## Restrições para nomes de variáveis

- Não pode começar com dígito: use uma letra ou \_
- Não usar acentos ou til
- Não pode ter espaço em branco
- Sugestão: use nomes que tenham um significado

Errado:

```
int 5minutes;  
int salário;  
int salario do funcionario;
```

Correto:

```
int _5minutes;  
int salario;  
int salarioDoFuncionario;
```

---

---

---

---

---

---

---

## Convenções

- Camel Case: lastName
  - pacotes
  - atributos
  - métodos
  - variáveis e parâmetros
- Pascal Case: ProductService
  - classes

---

---

---

---

---

---

---

```
package entities;

public class Account {

    private String holder;
    private Double balance;

    public Account(String holder, Double balance) {
        this.holder = holder;
        this.balance = balance;
    }

    public String getHolder() {
        return holder;
    }

    public void deposit(double amount) {
        balance += amount;
    }

    public void withdraw(double amount) {
        balance -= amount;
    }
}
```

---

---

---

---

---

---

---

## Operadores bitwise

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Operadores bitwise

| Operador | Significado                       |
|----------|-----------------------------------|
| &        | Operação "E" bit a bit            |
|          | Operação "OU" bit a bit           |
| ^        | Operação "OU-exclusivo" bit a bit |

| C1 | C2 | C1 E C2 |
|----|----|---------|
| F  | F  | F       |
| F  | V  | F       |
| V  | F  | F       |
| V  | V  | V       |

| C1 | C2 | C1 OU C2 |
|----|----|----------|
| F  | F  | F        |
| F  | V  | V        |
| V  | F  | V        |
| V  | V  | V        |

| C1 | C2 | C1 XOR C2 |
|----|----|-----------|
| F  | F  | F         |
| F  | V  | V         |
| V  | F  | V         |
| V  | V  | F         |

## Demo

(89) 0101 1001 }  
(60) 0011 1100 }  
                  {&: 0001 1000 (24)  
                  {|: 0111 1101 (125)  
                  {^: 0110 0101 (101)

```
int n1 = 89;  
int n2 = 60;  
System.out.println(n1 & n2);  
System.out.println(n1 | n2);  
System.out.println(n1 ^ n2);
```

## Aplicação comum: verificar bit

(89) 0101 1001 }  
(32) 0010 0000 } &: 0000 0000 (0)  
(113) 0111 0001 }  
(32) 0010 0000 } &: 0010 0000 (32)

```
package course;

import java.util.Scanner;

public class Program {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int mask = 0b100000;
        int n = sc.nextInt();

        if ((n & mask) != 0) {
            System.out.println("6th bit is true!");
        }
        else {
            System.out.println("6th bit is false");
        }

        sc.close();
    }
}
```

---

---

---

---

---

---

---

## Funções interessantes para String

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Checklist

- Formatar: toLowerCase(), toUpperCase(), trim()
- Recortar: substring(inicio), substring(inicio, fim)
- Substituir: Replace(char, char), Replace(string, string)
- Buscar: IndexOf, LastIndexOf
- str.Split(" ")

---

---

---

---

---

---

---

```
String original = "abcde FGHIJ ABC abc DEFG ";

String s01 = original.toLowerCase();
String s02 = original.toUpperCase();
String s03 = original.trim();
String s04 = original.substring(2);
String s05 = original.substring(2, 9);
String s06 = original.replace('a', 'x');
String s07 = original.replace("abc", "xy");
int i = original.indexOf("bc");
int j = original.lastIndexOf("bc");

System.out.println("Original: " + original + "\n");
System.out.println("toLowerCase: " + s01 + "\n");
System.out.println("toUpperCase: " + s02 + "\n");
System.out.println("trim: " + s03 + "\n");
System.out.println("substring(2): " + s04 + "\n");
System.out.println("substring(2, 9): " + s05 + "\n");
System.out.println("replace('a', 'x'): " + s06 + "\n");
System.out.println("replace('abc', 'xy'): " + s07 + "\n");
System.out.println("Index of 'bc': " + i);
System.out.println("Last index of 'bc': " + j);
```

---

---

---

---

---

---

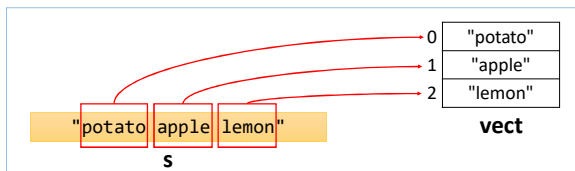
---

---

## Operação split

```
String s = "potato apple lemon";

String[] vect = s.split(" ");
String word1 = vect[0];
String word2 = vect[1];
String word3 = vect[2];
```




---

---

---

---

---

---

---

---

## Comentários em Java (básico)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

---

```
package course;

import java.util.Locale;
import java.util.Scanner;

/*
Este programa calcula as raízes de uma equação do segundo grau
Os valores dos coeficientes devem ser digitados um por linha
*/
public class Program {
    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        double a, b, c, delta;

        System.out.println("Digite os valores dos coeficientes:");
        a = sc.nextDouble();
        b = sc.nextDouble();
        c = sc.nextDouble();

        delta = b * b - 4 * a * c; // cálculo do valor de delta
    }
}
```

---

---

---

---

---

---

---

## Funções (sintaxe)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

---

---

---

---

---

---

---

## Funções

- Representam um processamento que possui um significado
  - `Math.sqrt(double)`
  - `System.out.println(string)`
- Principais vantagens: modularização, delegação e reaproveitamento
- Dados de entrada e saída
  - Funções podem receber dados de entrada (parâmetros ou argumentos)
  - Funções podem ou não retornar uma saída
- Em orientação a objetos, funções em classes recebem o nome de "métodos"

---

---

---

---

---

---

---

## Problema exemplo

Fazer um programa para ler três números inteiros e mostrar na tela o maior deles.

### Exemplo:

```
Enter three numbers:
5
8
3
Higher = 8
```

---

---

---

---

---

---

---

---

```
package course;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter three numbers:");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();

        if (a > b && a > c) {
            System.out.println("Higher = " + a);
        } else if (b > c) {
            System.out.println("Higher = " + b);
        } else {
            System.out.println("Higher = " + c);
        }

        sc.close();
    }
}
```

---

---

---

---

---

---

---

---

```
package course;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter three numbers:");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();

        int higher = max(a, b, c);

        showResult(higher);

        sc.close();
    }

    public static int max(int x, int y, int z) {
        int aux;
        if (x > y && x > z) {
            aux = x;
        } else if (y > z) {
            aux = y;
        } else {
            aux = z;
        }
        return aux;
    }

    public static void showResult(int value) {
        System.out.println("Higher = " + value);
    }
}
```

---

---

---

---

---

---

---

---