



# LINGUAGEM DE PROGRAMAÇÃO ORIENTADA A OBJETOS

**ETEC DE HORTOLÂNDIA**

CURSO TÉCNICO EM INFORMÁTICA INTEGRADO AO ENSINO MÉDIO

PROF. RALFE DELLA CROCE FILHO

# CONTEÚDO

- Histórico
- Características técnicas
- A linguagem Java
  - Tipos primitivos
  - Classes empacotadoras
  - Desvios condicionais
  - Laços de repetição
  - Operadores
- Conversões
- Eclipse
  - Instalação
  - Workspace
  - Criar e importar projeto
- Classe JOptionPane
- Revisão de lógica
- Métodos e assinaturas de métodos

# HISTÓRICO

- Em 1991 teve início o Green Project, na Sun Microsystems, coordenado por Patrick Naughton, Mike Sheridan, e James Gosling.
- Este projeto tinha a intenção de se antecipar a “próxima onda” que aconteceria na área da informática e programação onde aparelhos domésticos e computadores se comunicariam e para isso criaram o \*7 (StarSeven).
- Visando a comunicação e entre softwares desenvolvidos com tecnologias diferentes sendo executados em plataformas (Sistemas Operacionais e Hardware) diferentes essa equipe criou uma nova linguagem de programação.



# HISTÓRICO

- Como as demais tecnologias envolvidas e o mercado ainda não estavam preparados para o tipo de interação abordada o projeto não obteve êxito.
- Aproximadamente em 1993, Tim Bernes-Lee concluiu e disponibilizou seu sistema de informação em hipertexto que ele chamou de *World Wide Web* que ganhou grande popularidade e os idealizadores do projeto Green viram nela uma oportunidade.
- Em 1995 a Sun lançou oficialmente a primeira versão da Linguagem Java.
- Em 2009 a Oracle comprou a Sun Microsystems (por US\$ 7,4 bilhões).



# JAVA HOJE

- O Java é a base para praticamente todos os tipos de aplicações em rede e é o padrão global para o desenvolvimento e distribuição de aplicações móveis e incorporadas, jogos, conteúdo baseado na Web e softwares corporativos.
- De laptops a datacenters, consoles de games a supercomputadores científicos, telefones celulares à Internet, o Java está em todos os lugares!

fonte: [https://www.java.com/pt\\_BR/about/](https://www.java.com/pt_BR/about/) (em 08/2017)

# JAVA HOJE

- 97% dos Desktops Corporativos executam o Java
- 89% dos Desktops (ou Computadores) nos EUA Executam Java
- 9 Milhões de Desenvolvedores de Java em Todo o Mundo
- Plataforma de Desenvolvimento N° 1
- 3 Bilhões de Telefones Celulares Executam o Java
- 100% dos Blu-ray Disc Players Vêm Equipados com o Java
- 5 bilhões de Placas Java em uso
- 125 milhões de aparelhos de TV executam o Java
- 5 dos 5 Principais Fabricantes de Equipamento Original Utilizam o Java ME

fonte: [https://www.java.com/pt\\_BR/about/](https://www.java.com/pt_BR/about/) (em 08/2017)

## MERCADO

- O Índice TIOBE é um indicador de popularidade das linguagens de programação atualizado mensalmente. As avaliações são baseadas no número de engenheiros, cursos e vendedores em várias partes do mundo.

Jul 2017	Jul 2016	Change	Programming Language	Ratings	Change
1	1		Java	13.774%	-6.03%
2	2		C	7.321%	-4.92%
3	3		C++	5.576%	-0.73%
4	4		Python	3.543%	-0.62%
5	5		C#	3.518%	-0.40%
6	6		PHP	3.093%	-0.18%
7	8	⬆	Visual Basic .NET	3.050%	+0.53%
8	7	⬇	JavaScript	2.606%	-0.04%
9	12	⬆	Delphi/Object Pascal	2.490%	+0.45%
10	55	⬆	Go	2.363%	+2.20%



fonte: <https://www.tiobe.com/tiobe-index/> (em 08/2017)

# MERCADO

- Linguagens de Programação mais usadas em 2017 segundo a IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos)



## Language Types (click to hide)



Web



Mobile



Enterprise



Embedded

### Language Rank

### Types

### Spectrum Ranking

1. Python		100.0
2. C		100.0
3. Java		99.4
4. C++		96.9
5. C#		88.6
6. R		88.1
7. JavaScript		85.3
8. PHP		81.1
9. Go		75.7
10. Swift		74.3

fonte: <http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2017> (em 08/2017)



# MERCADO



[Buscar vagas](#) [Guia de profissões](#) [Por dentro das empresas](#) [Soluções para candidatos](#)

[Candidatos](#) > [Buscar vagas](#) > [Desenvolvedor Java](#) > [SP](#)

729 vagas de emprego de desenvolvedor java em São Paulo

**InfoJobs**

**Candidatos**

[Empresas](#)

[Cursos](#)

[Blog](#)

[Infojobs](#) > [Vagas de emprego](#) > [São Paulo](#) > [desenvolvedor java](#)

**905** vagas de emprego de **DESENVOLVEDOR JAVA** em **São Paulo, SP** de 635 anúncios

- Obs.: Busca realizada por “desenvolvedor java” na cidade de São Paulo em 08/2017.

# AMBIENTES DE DESENVOLVIMENTO

- JSE (Java Standard Edition)
- JEE (Java Enterprise Edition)
- JME (Java Micro Edition)

## Technologies

Java SE

Java SE Support

Java SE Advanced & Suite

Java Embedded

Java EE

Java ME

JavaFX

Java Card

Java TV

Java DB

Developer Tools

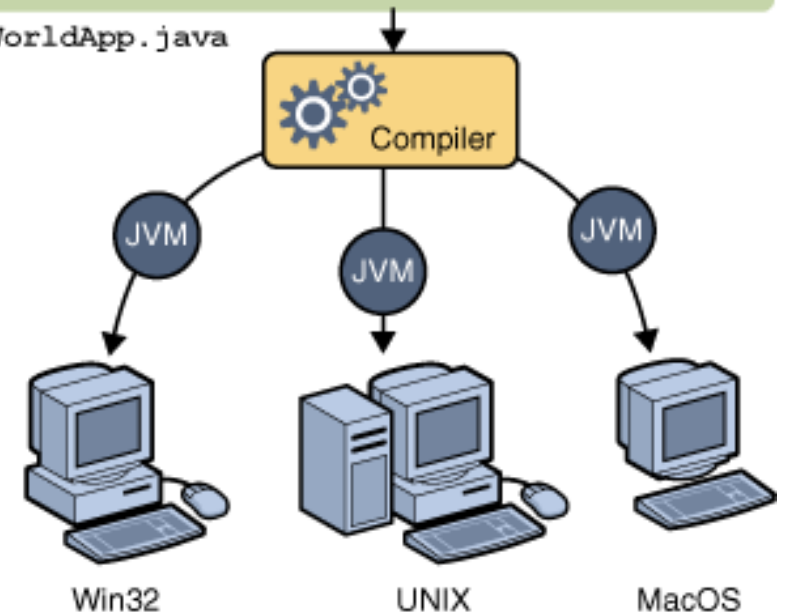
# COMPONENTES BÁSICOS DO JAVA

- JRE (Java Runtime Environment)
- API (Application Programming Interface)
- JVM (Java Virtual Machine)

Java Program

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java



# DOWNLOAD JAVA

Java SE
Java EE
Java ME
Java SE Support
Java SE Advanced & Suite
Java Embedded
Java DB
Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine

[Overview](#) [Downloads](#) [Documentation](#) [Community](#) [Technologies](#) [Training](#)

## Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- [Java Developer Newsletter](#): From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- [Java Developer Day hands-on workshops \(free\)](#) and other events
- [Java Magazine](#)

JDK 8u144 checksum

## Java SE Development Kit 8u144

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.89 MB	<a href="#">jdk-8u144-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.83 MB	<a href="#">jdk-8u144-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	164.65 MB	<a href="#">jdk-8u144-linux-i586.rpm</a>
Linux x86	179.44 MB	<a href="#">jdk-8u144-linux-i586.tar.gz</a>
Linux x64	162.1 MB	<a href="#">jdk-8u144-linux-x64.rpm</a>
Linux x64	176.92 MB	<a href="#">jdk-8u144-linux-x64.tar.gz</a>
Mac OS X	226.6 MB	<a href="#">jdk-8u144-macosx-x64.dmg</a>
Solaris SPARC 64-bit	139.87 MB	<a href="#">jdk-8u144-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	99.18 MB	<a href="#">jdk-8u144-solaris-sparcv9.tar.gz</a>
Solaris x64	140.51 MB	<a href="#">jdk-8u144-solaris-x64.tar.Z</a>
Solaris x64	96.99 MB	<a href="#">jdk-8u144-solaris-x64.tar.gz</a>
Windows x86	190.94 MB	<a href="#">jdk-8u144-windows-i586.exe</a>
Windows x64	197.78 MB	<a href="#">jdk-8u144-windows-x64.exe</a>

### Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

### Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Java.net](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)

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

# DOWNLOAD ECLIPSE



GETTING STARTED

MEMBERS

PROJECTS

MORE ▾

Download Eclipse Technology  
that is right for you



Get Eclipse **OXYGEN**

Install your favorite Eclipse packages.

**DOWNLOAD 64 BIT**

Download Packages

<http://www.eclipse.org/downloads/>

# INSTALAÇÃO ECLIPSE



# A LINGUAGEM JAVA

**PALAVRAS RESERVADAS** dessa linguagem são nomes utilizados pelo compilador para representar comandos de controle do programa, operadores e diretivas.

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

# A LINGUAGEM JAVA

- Tipos primitivos
- O Java possui classes “empacotadoras” (Boolean, Byte, Character, Double, Float, Integer, Long, Short e String) que permitem manipular valores de tipo primitivo como objetos.

Tipo	Faixa de valores	Tamanho
boolean	true ou false	8 bits
char	0 a 65.535	16 bits
byte	-128 a 127	8 bits
short	-32.768 a 32.767	16 bits
int	-2.147.483.648 a 2.147.483.647	32 bits
long	-9.223.372.036.854.808 a 9.223.372.036.854.775.807	64 bits
float	$-3,4 * 10^{38}$ a $3,4 * 10^{38}$	32 bits
double	$-1,7 * 10^{308}$ a $1,7 * 10^{308}$	64 bits



# A LINGUAGEM JAVA

## ■ Operadores aritméticos

Operador	Exemplo	Comentário
=	$x = y$	O conteúdo de y é atribuído à x
+	$x + y$	Soma o conteúdo de x ao conteúdo de y
-	$x - y$	Subtrai o conteúdo de y do conteúdo de x
*	$x * y$	Multiplica o conteúdo de x pelo conteúdo de y
/	$x / y$	Divide o conteúdo de x pelo conteúdo de y
%	$x \% y$	Obtém o resto da divisão do conteúdo de x pelo conteúdo de y

# A LINGUAGEM JAVA

## ■ Operadores aritméticos

Operador	Exemplo	Comentário
<code>+=</code>	<code>x += y</code>	Equivale a <code>x = x + y</code>
<code>-=</code>	<code>x -= y</code>	Equivale a <code>x = x - y</code>
<code>*=</code>	<code>x *= y</code>	Equivale a <code>x = x * y</code>
<code>/=</code>	<code>x /= y</code>	Equivale a <code>x = x / y</code>
<code>++</code>	<code>x++</code>	Considera o valor de x depois soma 1
<code>++</code>	<code>++x</code>	Soma 1 depois considera o valor de x
<code>--</code>	<code>x--</code>	Considera o valor de x depois subtrai 1
<code>--</code>	<code>--x</code>	Subtrai 1 depois considera o valor de x

# A LINGUAGEM JAVA

## ■ Operadores relacionais

Operador	Exemplo	Comentário
==	x == y	O conteúdo de x é igual ao conteúdo de y
!=	x != y	O conteúdo de x é diferente do conteúdo de y
<=	x <= y	O conteúdo de x é menor ou igual ao conteúdo de y
>=	x >= y	O conteúdo de x é maior ou igual ao conteúdo de y
<	x < y	O conteúdo de x é menor que o conteúdo de y
>	x > y	O conteúdo de x é maior que o conteúdo de y

# A LINGUAGEM JAVA

## ■ Operadores lógicos

Operador	Exemplo	Comentário
!	!(condição)	NÃO – Inverte o resultado lógico
&&	(condição1 && condição2)	E – Resultado lógico V somente se as duas condições forem V
	(condição1    condição2)	OU – Resultado lógico V se uma das duas, ou as duas, condições forem V
	(condição1   condição2)	OU exclusivo – Resultado lógico V somente se uma das duas condições forem V

# A LINGUAGEM JAVA

- Identificadores: nomes dados as variáveis, métodos, classes e projetos.
- Regras:
  - Não devem conter caracteres especiais (\*, \$, %, etc.), exceto, \_ (underline).
  - Não devem ser acentuados.
  - Não podem conter espaço.
  - Não podem começar com um número.
  - Não podem ser palavras reservadas.
  - Não podem ser duplicados.

Obs.: Devem ser claros e objetivos não deixando dúvidas sobre o que armazenam.

# A LINGUAGEM JAVA

## ■ Declaração de variáveis

```
// Declaração de variáveis  
int numero = 10;  
double precoCusto, precoVenda;  
boolean ativo = true;  
String nome;
```

# A LINGUAGEM JAVA

## ■ Desvios condicionais

```
if(numero > 0){  
    // Instruções  
}else {  
    // Instruções  
}
```

```
switch (numero) {  
case 5:  
    // Instruções  
    break;  
  
case 10:  
    // Instruções  
    break;  
  
case 15:  
    // Instruções  
    break;  
  
default:  
    // Instruções  
    break;  
}
```

# A LINGUAGEM JAVA

## ■ Laços de repetição

```
while(numero <= 10) {  
    // Instruções  
}
```

```
do {  
    // Instruções  
}while(numero <= 10);
```

```
for (int i = 0; i < 10; i++) {  
    // Instruções  
}
```



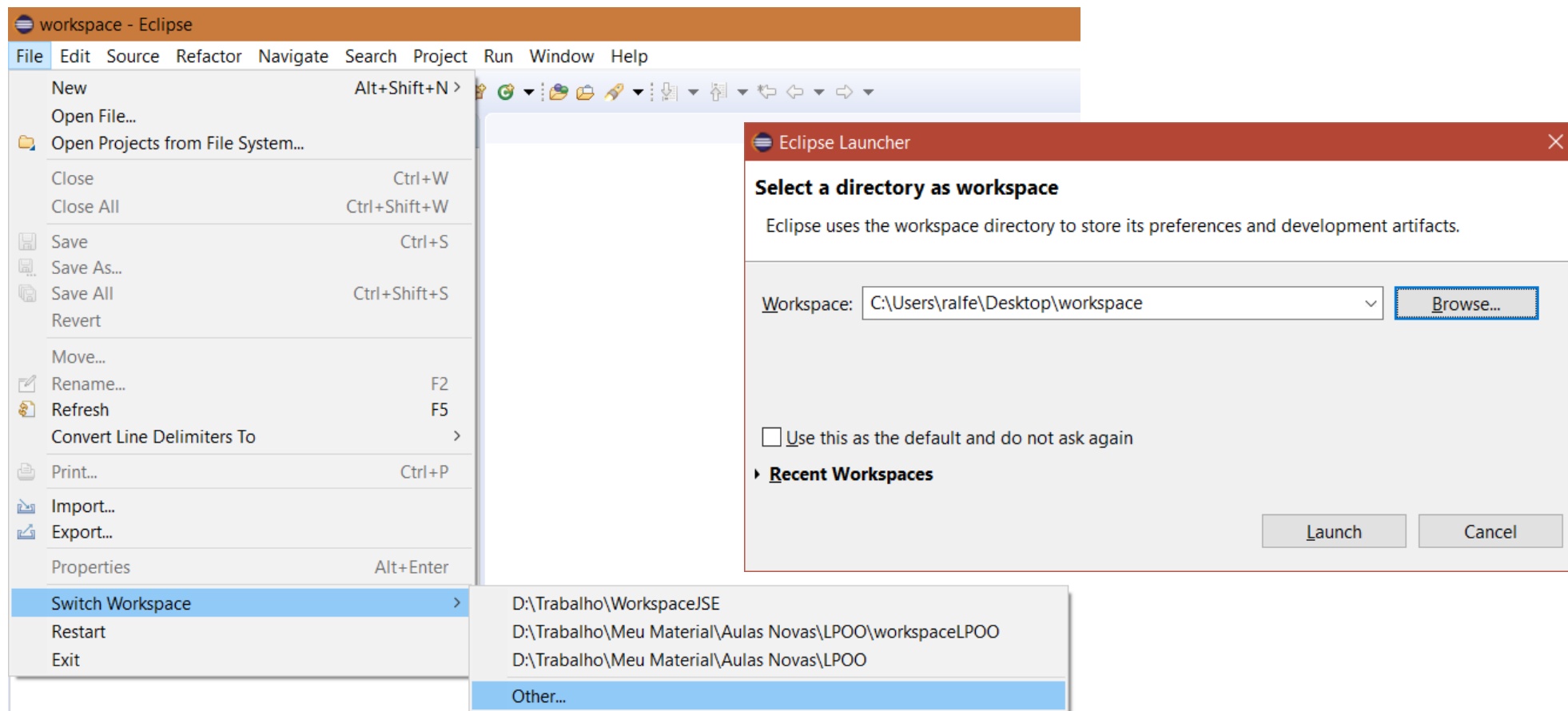
# A LINGUAGEM JAVA

## ■ Conversões de tipos

```
// Declaração de variáveis inteiras.  
short varShort = 0; // Menor faixa de abrangência.  
int varInt = 0;     // Maior faixa de abrangência.  
  
// Declaração de variáveis reais.  
float varFloat = 0; // Menor faixa de abrangência.  
double varDouble = 0; // Maior faixa de abrangência.  
  
// Declaração de variáveis texto.  
String varString = "";
```

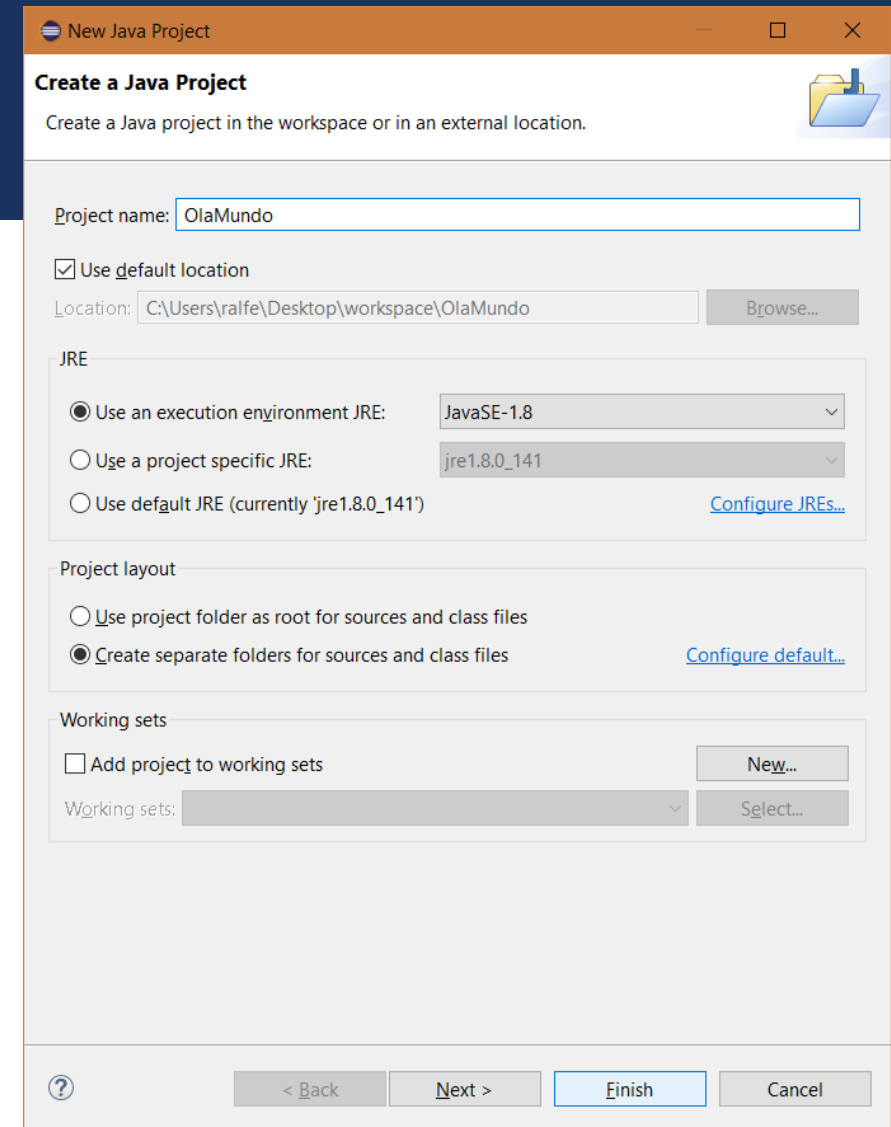
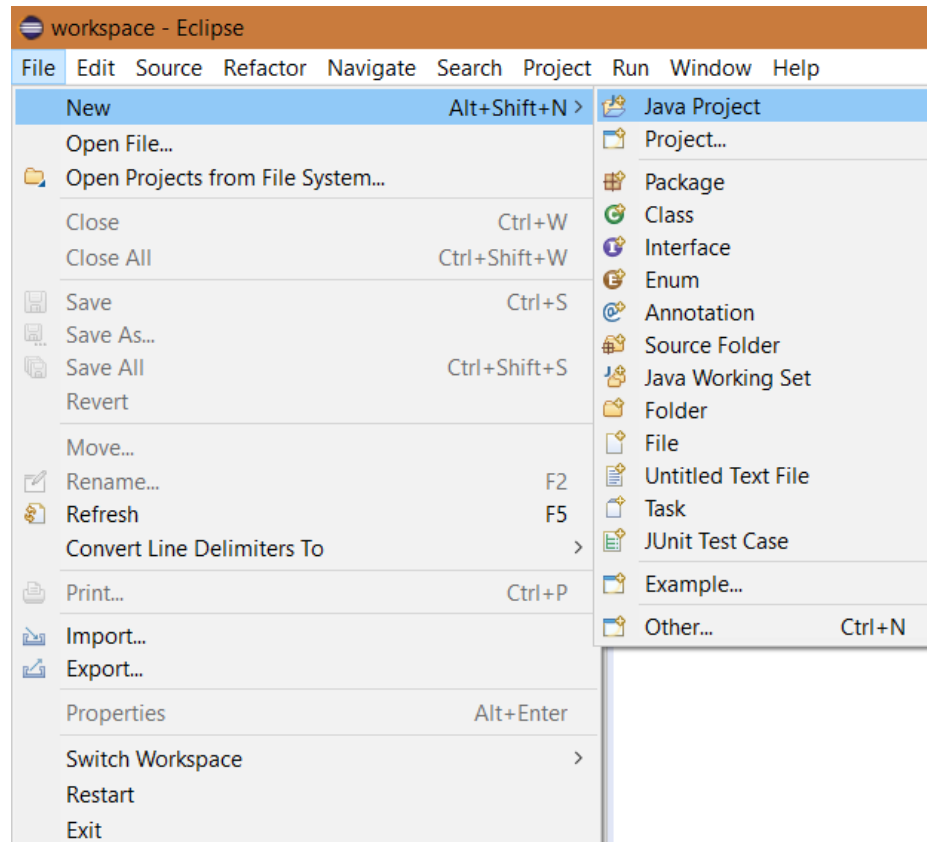
```
// Conversões entre números  
varShort = (short) varInt; // Cast do maior para o menor.  
varInt = varShort;  
  
varFloat = (float) varDouble; // Cast do maior para o menor.  
varDouble = varFloat;  
  
varShort = (short) varFloat; // Cast do maior para o menor.  
varFloat = varShort;  
  
varInt = (int) varDouble; // Cast do maior para o menor.  
varDouble = varInt;  
  
// Conversões de String para número  
varShort = Short.parseShort(varString);  
varInt = Integer.parseInt(varString);  
varFloat = Float.parseFloat(varString);  
varDouble = Double.parseDouble(varString);  
  
// Conversões de número para String  
varString = String.valueOf(varShort);  
varString = String.valueOf(varInt);  
varString = String.valueOf(varFloat);  
varString = String.valueOf(varDouble);
```

# WORKSPACE



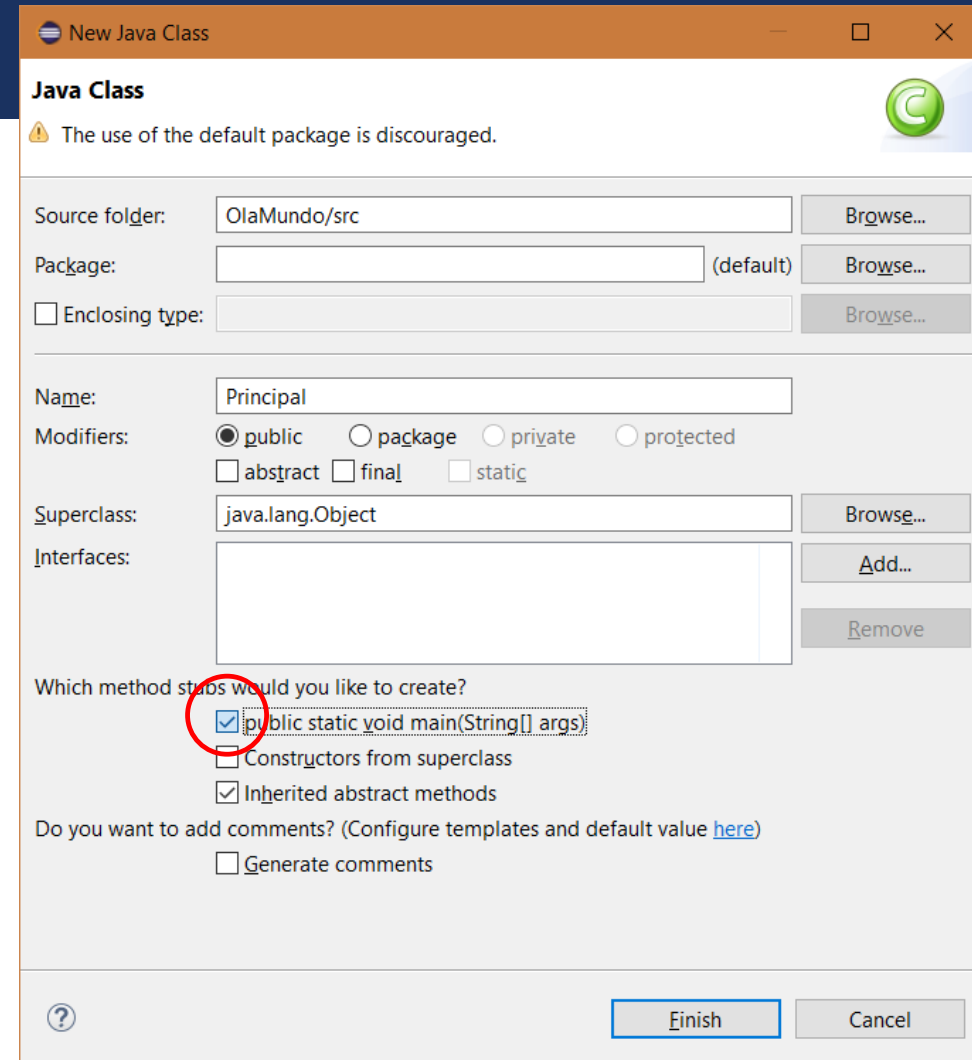
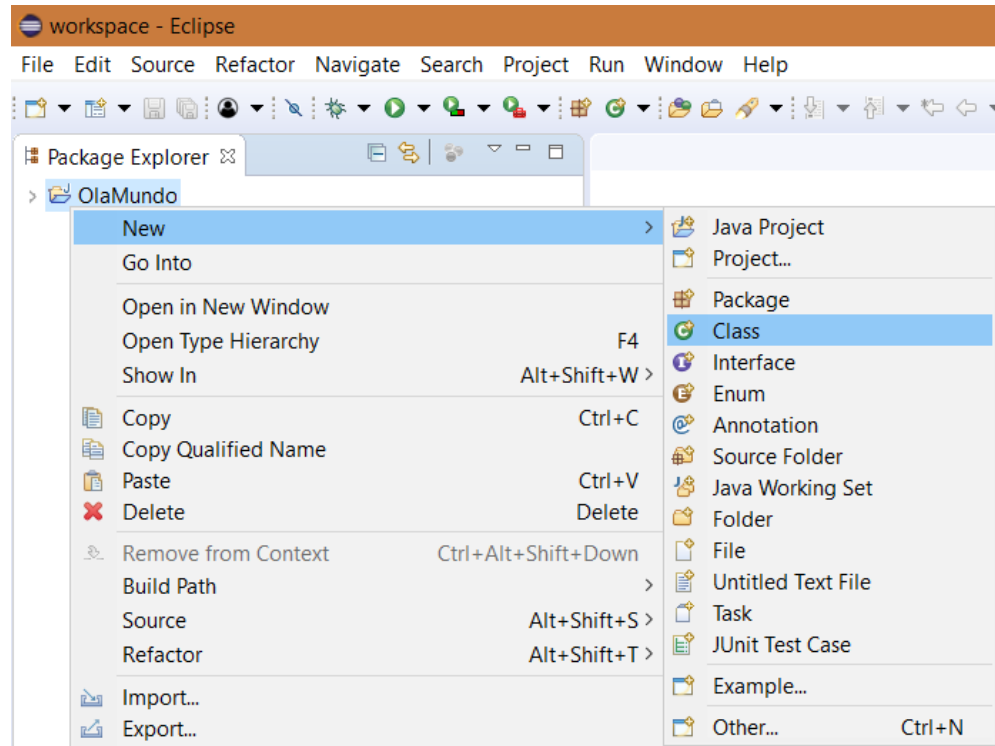
# PROJETO

## ■ OlaMundo

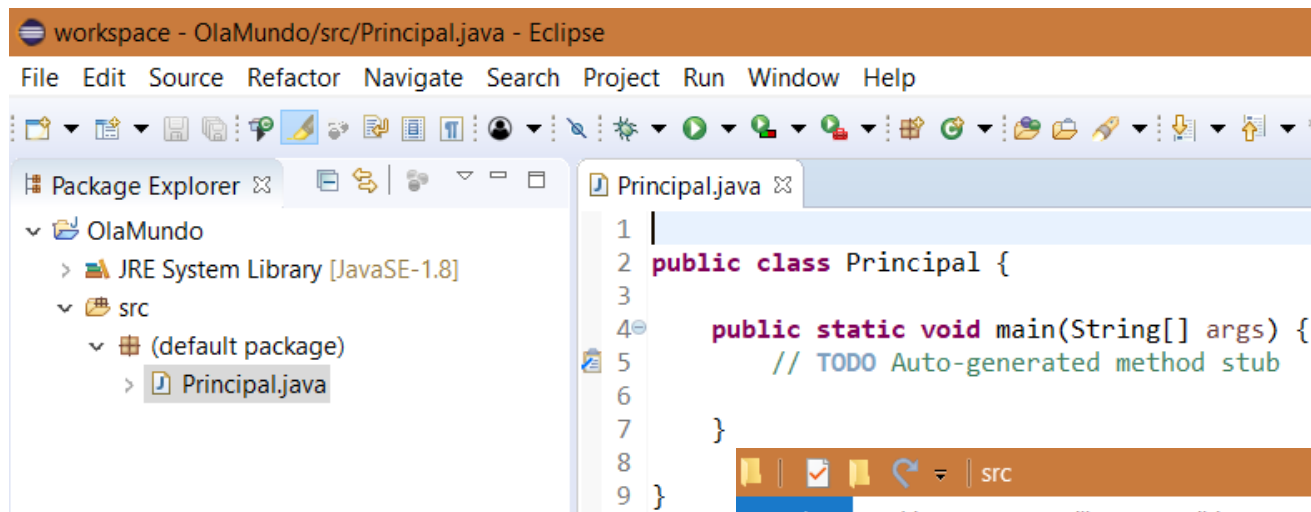


# CLASSE

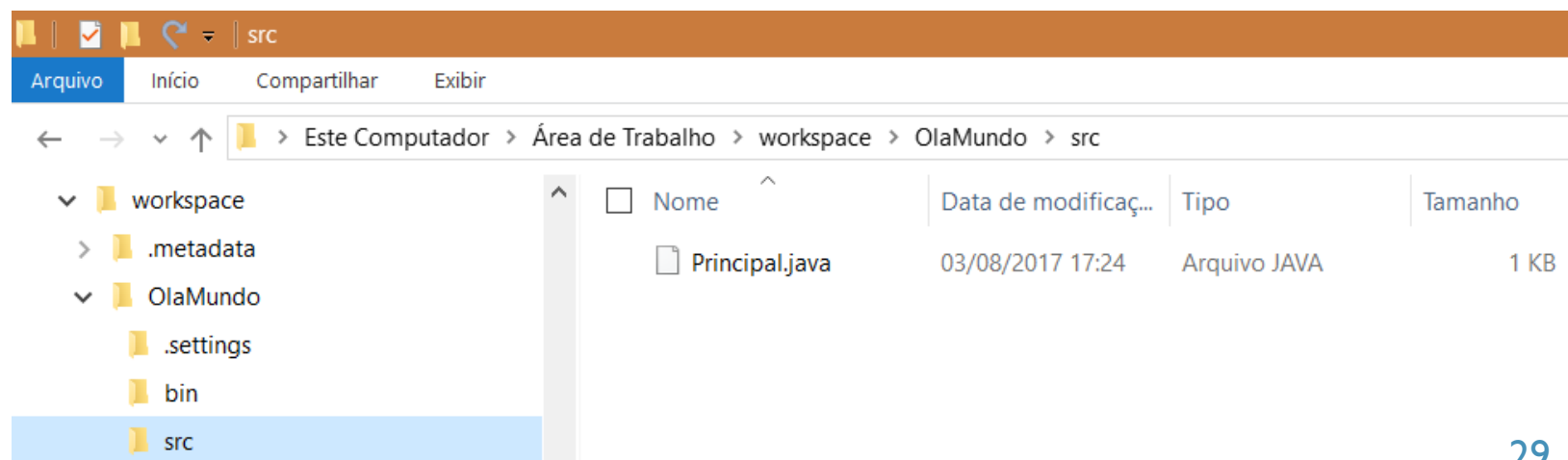
## ■ Classe com o método main



# WORKSPACE



## ■ Área de código



## ■ workspace

# CONCATENAÇÃO

Entrada

ⓧ

ⓧ Digite seu nome:

Thomas

OK Cancelar

Entrada

ⓧ

ⓧ Digite seu sobrenome:

A. Anderson

OK Cancelar

Mensagem

ⓧ

ⓘ Olá Thomas A. Anderson, seja bem-vindo!

OK

```
Principal.java
1 import javax.swing.JOptionPane;
2 /**
3  * @author Prof. Ralfe
4  * @version Última atualização: 04/08/2017
5  */
6 public class Principal {
7
8     public static void main(String[] args) {
9
10         // Declaração de variáveis
11         String nome, sobrenome, mensagem;
12
13         // Entrada
14         nome = JOptionPane.showInputDialog("Digite seu nome:");
15         sobrenome = JOptionPane.showInputDialog("Digite seu sobrenome:");
16
17         // Processamento
18         mensagem = "Olá " + nome + " " + sobrenome + ", seja bem-vindo!";
19
20         // Saída
21         JOptionPane.showMessageDialog(null, mensagem);
22     }
23 }
```

## ■ Projeto: OlaMundo

# CONVERSÕES

## ■ Projeto: Calculadora

```
Principal.java ✕
1 import javax.swing.JOptionPane;
2 /**
3  * @author Prof. Ralfe
4  * @version Última atualização: 04/08/2017
5  */
6 public class Principal {
7
8     public static void main(String[] args) {
9
10         // Declaração de variáveis
11         double n1, n2, resultado;
12
13         // Leitura e conversão dos tipos de dados
14         n1 = Double.parseDouble( JOptionPane.showInputDialog("Digite o primeiro valor: "));
15         n2 = Double.parseDouble( JOptionPane.showInputDialog("Digite o segundo valor: "));
16
17         //Cálculo
18         resultado = n1 + n2;
19
20         // Apresentação do resultado
21         JOptionPane.showMessageDialog(null, "Resultado: " + resultado);
22     }
23 }
```

```
6 public class Principal {
7     public static void main(String[] args) {
8         // Declaração de variáveis
9         double n1, n2, resultado = 0;
10        int operacao;
11
12        // Menu de operações
13        operacao = Integer.parseInt(JOptionPane.showInputDialog("1 - Somar \n2 - Subtrair \n3 - Multiplicar \n4 - Dividir"));
14
15        if(operacao >= 1 && operacao <= 4) {
16            // Leitura e conversão dos tipos de dados
17            n1 = Double.parseDouble( JOptionPane.showInputDialog("Digite o primeiro valor: "));
18            n2 = Double.parseDouble( JOptionPane.showInputDialog("Digite o segundo valor: "));
19
20            // Verifica a operação escolhida e efetua o calculo
21            if(operacao == 1) { // Soma
22                resultado = n1 + n2;
23            }else {
24                if(operacao == 2) { // Subtração
25                    resultado = n1 - n2;
26                }else {
27                    if(operacao == 3) { // Multiplicação
28                        resultado = n1 * n2;
29                    }else { // Divisão
30                        resultado = n1 / n2;
31                    }
32                }
33            }
34            // Apresentação do resultado
35            JOptionPane.showMessageDialog(null, "Resultado: " + resultado);
36        }else {
37            JOptionPane.showMessageDialog(null, "Operação inválida!");
38        }
39    }
40 }
```

## ■ Projeto: Calculadora



## ■ Projeto: Calculadora

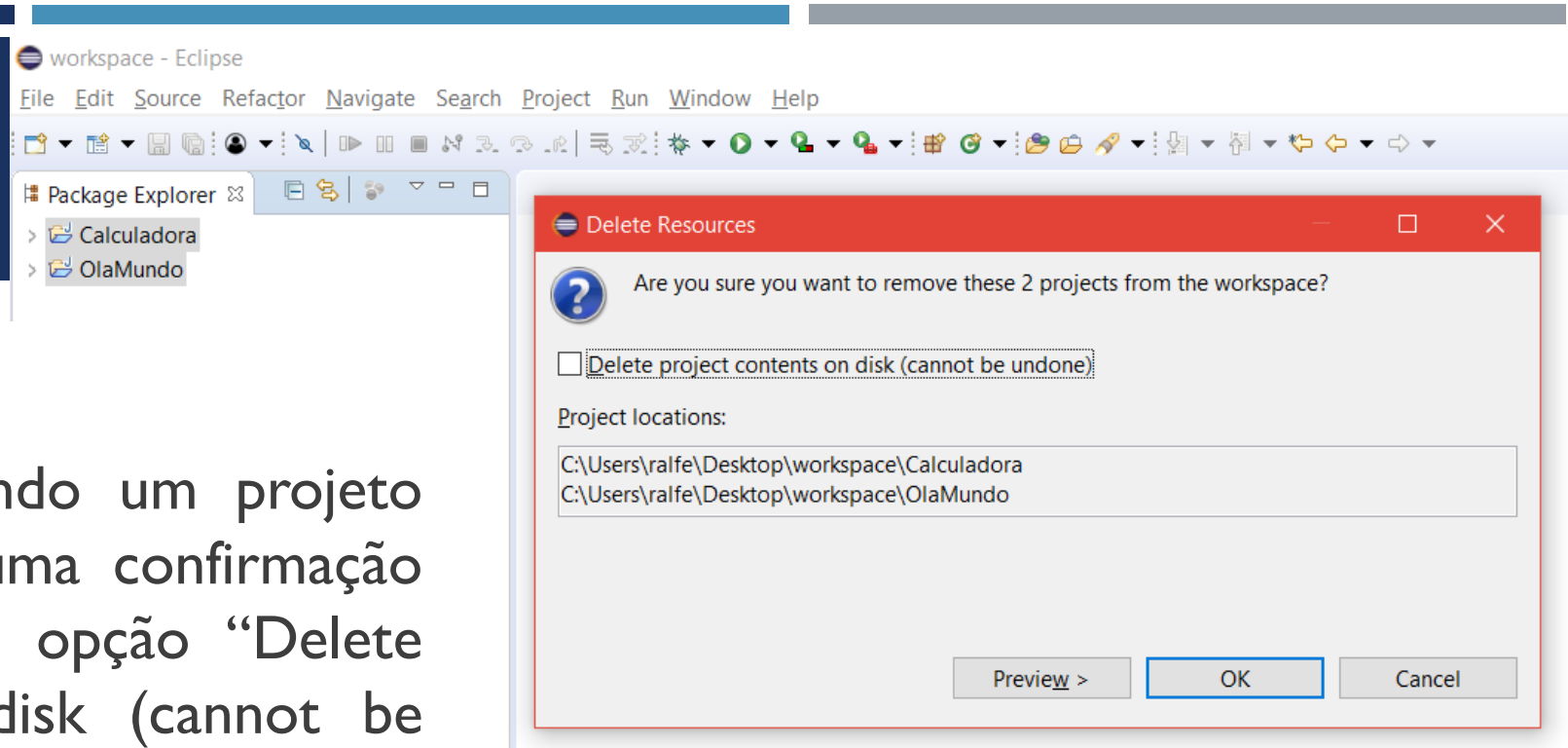
```
8 public static void main(String[] args) {
9
10     // Declaração de variáveis
11     double n1, n2, resultado = 0;
12     int operacao;
13
14     // Menu de operações
15     operacao = Integer.parseInt(JOptionPane.showInputDialog("1 - Somar \n2 - Subtrair \n3 - Multiplicar \n4 - Dividir"));
16
17     // Leitura e conversão dos tipos de dados
18     n1 = Double.parseDouble(JOptionPane.showInputDialog("Digite o primeiro valor: "));
19     n2 = Double.parseDouble(JOptionPane.showInputDialog("Digite o segundo valor: "));
20
21     // Verifica a operação escolhida e efetua o calculo
22     switch (operacao) {
23     case 1: // Soma
24         resultado = n1 + n2;
25         break;
26     case 2: // Subtração
27         resultado = n1 - n2;
28         break;
29     case 3: // Multiplicação
30         resultado = n1 * n2;
31         break;
32     case 4: // Divisão
33         resultado = n1 / n2;
34         break;
35     default:
36         JOptionPane.showMessageDialog(null, "Operação inválida!");
37         break;
38     }
39
40     // Apresentação do resultado
41     JOptionPane.showMessageDialog(null, "Resultado: " + resultado);
42 }
43 }
```

## ■ Projeto: Calculadora

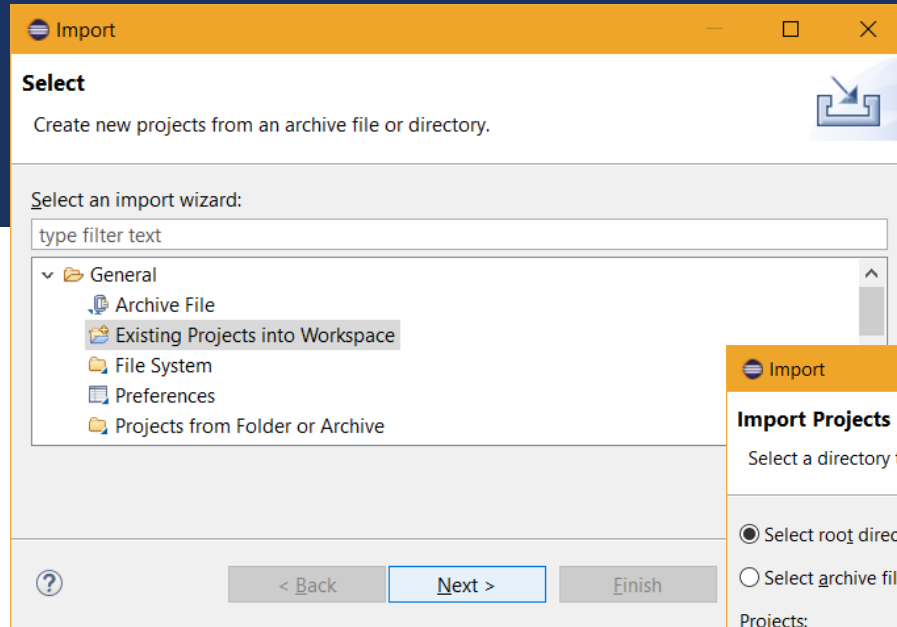
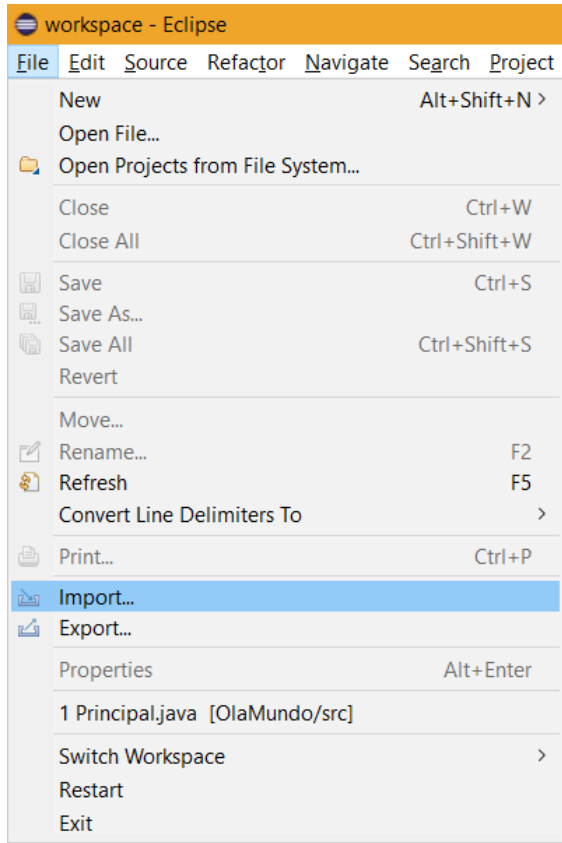
```
6 public class Principal {
7     public static void main(String[] args) {
8         // Declaração de variáveis
9         double n1, n2, resultado = 0;
10        int operacao, continua;
11
12        do {
13            // Menu de operações
14            operacao = Integer.parseInt(JOptionPane.showInputDialog("1 - Somar \n2 - Subtrair \n3 - Multiplicar \n4 - Dividir"));
15            if(operacao < 1 || operacao > 4) {
16                JOptionPane.showMessageDialog(null, "Operação inválida!");
17            }else {
18                // Leitura e conversão dos tipos de dados
19                n1 = Double.parseDouble( JOptionPane.showInputDialog("Digite o primeiro valor: "));
20                n2 = Double.parseDouble( JOptionPane.showInputDialog("Digite o segundo valor: "));
21                // Verifica a operação escolhida e efetua o calculo
22                switch (operacao) {
23                    case 1: // Soma
24                        resultado = n1 + n2;
25                        break;
26                    case 2: // Subtração
27                        resultado = n1 - n2;
28                        break;
29                    case 3: // Multiplicação
30                        resultado = n1 * n2;
31                        break;
32                    case 4: // Divisão
33                        resultado = n1 / n2;
34                        break;
35                }
36                // Apresentação do resultado
37                JOptionPane.showMessageDialog(null, "Resultado: " + resultado);
38            }
39            // O método showConfirmDialog disponibiliza combinações de botões, nesse caso, Sim e Não
40            continua = JOptionPane.showConfirmDialog(null, "Deseja realizar mais um calculo?", "Calculadora", JOptionPane.YES_NO_OPTION);
41        }while (continua == 0); // Essa é uma das formas de testar o retorno (botão clicado) do método showConfirmDialog considerando 0 Sim e 1 Não
42    }
43 }
```

# EXCLUSÃO

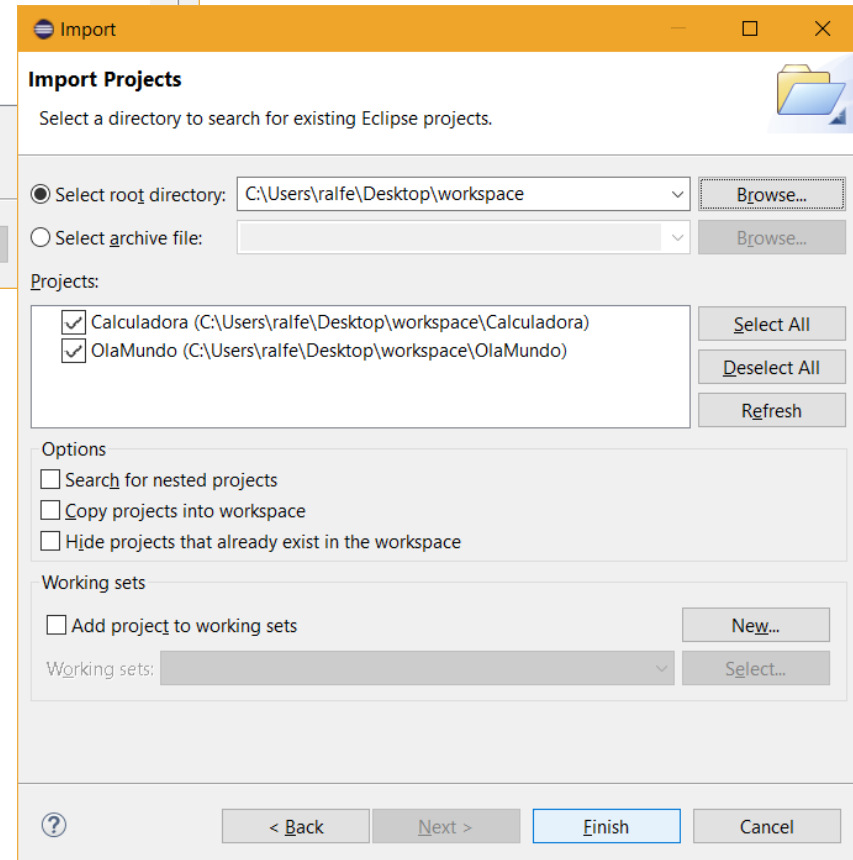
- Excluir projeto
  - Selecionando e deletando um projeto no Package Explorer uma confirmação será apresentada. Se a opção “Delete project contents on disk (cannot be undone)” não for selecionada o projeto será excluído somente da Package Explorer (permanecendo na workspace), se for selecionada, será excluído inclusive da workspace.



# IMPORTAÇÃO



## ■ Importar projeto



# EXERCÍCIOS

## ■ Projeto ControleEscolar I

Em um curso, a situação final do aluno pode ser “Aprovado” ou “Reprovado” e é baseada na média aritmética de duas notas.

- A média mínima para aprovação é 6.0

# EXERCÍCIOS

## ■ Projeto ControleEscolar2

Em um curso, a situação final do aluno pode ser “Aprovado”, “Aprovado na recuperação”, “Reprovado” ou “Reprovado por frequência” e é baseada na média aritmética de duas notas e na frequência do aluno.

- A frequência é obrigatória e excludente e deverá ser igual ou superior a 75.
- A média mínima para aprovação é 6.0
- Caso o aluno não obtenha a média mínima, uma nova nota será lida e também deverá ser maior ou igual a 6.0.

O usuário poderá verificar quantos alunos ele desejar.

# EXERCÍCIOS

## ■ Projeto Imc

O IMC (índice de massa corporal) é um critério da Organização Mundial da Saúde para dar uma indicação sobre a condição de peso de uma pessoa adulta. A fórmula é  $IMC = \text{peso} / (\text{altura})^2$ . Faça um algoritmo que leia o peso e a altura de um adulto e mostre sua condição.

- a. Abaixo de 18.5 é considerado: Abaixo do peso.
- b. Entre 18.5 até 25 é considerado: Peso normal.
- c. Entre 25 até 30 é considerado: Acima do peso.
- d. Acima de 30 é considerado: Obeso.

O programa avaliará os dados de 50 pessoas.

# EXERCÍCIOS

## ■ Projeto Numeros

Leia dois números e a opção escolhida pelo usuário, sendo:

- 1 – Apresente a média aritmética dos números.
- 2 – Apresenta o menor número.
- 3 – Apresenta a diferença entre o maior e o menor número.
- 4 – Apresenta os números em ordem crescente.

O usuário poderá executar as funcionalidades do programa quantas vezes ele desejar.

Obs.: Considere que os dois números são diferentes.



# SUB-ROTINAS

- Subdivisão/organização de um código extenso em blocos de menor complexidade.
- Em outras linguagens dividem-se em Funções (sub-rotina que ao ser executada retorna um valor) e Procedimentos (sub-rotina que ao ser executada não retorna um valor).
- Para a Programação Orientada a Objetos, sub-rotinas (com qualquer formato) são chamadas de métodos.

# ASSINATURAS DE MÉTODOS

- A primeira linha (cabeçalho) de um método define:
  - Visibilidade
  - Retorno de valor
  - O nome do método
  - Passagem de parâmetros
- A combinação do nome e da passagem de parâmetros compõe a assinatura do método que para as linguagens de programação orientadas a objetos é a identificação do método.

# UML (UNIFIED MODELING LANGUAGE)

- É uma linguagem ou notação de diagramas para especificar, visualizar e documentar modelos de softwares orientados a objetos por meio de diagramas:
  - Diagrama de Caso de Uso: mostra atores (pessoas ou outros usuários do sistema), casos de uso (os cenários onde eles usam o sistema), e seus relacionamentos.
  - **Diagrama de Classe: mostra classes e os relacionamentos entre elas.**
  - Diagrama de Sequência: mostra objetos e uma sequência das chamadas do método feitas para outros objetos.
  - Diagrama de Colaboração: mostra objetos e seus relacionamentos, colocando ênfase nos objetos que participam na troca de mensagens.
  - Diagrama de Estado: mostra estados, mudanças de estado e eventos num objeto ou uma parte do sistema.
  - Diagrama de Atividade: mostra atividades e as mudanças de uma atividade para outra com os eventos ocorridos em alguma parte do sistema.
  - Diagrama de Componente: mostra os componentes de programação de alto nível.
  - Diagrama de Distribuição: mostra as instâncias dos componentes e seus relacionamentos.
  - Diagramas de Entidade-Associação: mostram os dados e as relações e as restrições entre os dados.

# SEM PARÂMETRO E SEM RETORNO

```
public static void somar(){  
    // Entrada  
    n1 = Double.parseDouble( JOptionPane.showInputDialog("Digite o primeiro número:") );  
    n2 = Double.parseDouble( JOptionPane.showInputDialog("Digite o segundo número:") );  
    // Processamento  
    r = n1 + n2;  
    // Saida  
    JOptionPane.showMessageDialog(null, "Resultado: " + r);  
}
```

```
switch (operacao) {  
    case 1:  
        somar();  
        break;  
  
    case 2:  
        subtrair();  
        break;  
  
    case 3:  
        multiplicar();  
        break;  
  
    case 4:  
        dividir();  
        break;  
}
```

Principal
<u>+ somar() : void</u> <u>+ subtrair() : void</u> <u>+ multiplicar() : void</u> <u>+ dividir() : void</u> <u>+ main(args[ ] : String) : void</u>

```

public static double somar(){
    // Entrada
    n1 = Double.parseDouble( JOptionPane.showInputDialog("Digite o primeiro número:") );
    n2 = Double.parseDouble( JOptionPane.showInputDialog("Digite o segundo número:") );
    // Processamento
    r = n1 + n2;
    return r;
}

```

## SEM PARÂMETRO E COM RETORNO

Principal
<u>+ somar() : double</u> <u>+ subtrair() : double</u> <u>+ multiplicar() : double</u> <u>+ dividir() : double</u> <u>+ main(args[ ] : String) : void</u>

```

switch (operacao) {
    case 1:
        // Saida
        JOptionPane.showMessageDialog(null, "Resultado: " + somar());
        break;

    case 2:
        // Saida
        JOptionPane.showMessageDialog(null, "Resultado: " + subtrair());
        break;

    case 3:
        // Saida
        JOptionPane.showMessageDialog(null, "Resultado: " + multiplicar());
        break;

    case 4:
        // Saida
        JOptionPane.showMessageDialog(null, "Resultado: " + dividir());
        break;
}

```

# COM PARÂMETRO E SEM RETORNO

```
// Entrada
n1 = Double.parseDouble( JOptionPane.showInputDialog("Digite o primeiro número:") );
n2 = Double.parseDouble( JOptionPane.showInputDialog("Digite o segundo número:") );

// Verifica a operação escolhida e invoca o respectivo método
switch (operacao) {
case 1:
    somar(n1, n2);
    break;

case 2:
    subtrair(n1, n2);
    break;

case 3:
    multiplicar(n1, n2);
    break;

case 4:
    dividir(n1, n2);
    break;
}
```

PROF. RALFE DELLA CROCE FILHO

Principal
<u>+ somar(valor1 : double, valor2 : double) : void</u> <u>+ subtrair(valor1 : double, valor2 : double) : void</u> <u>+ multiplicar(valor1 : double, valor2 : double) : void</u> <u>+ dividir(valor1 : double, valor2 : double) : void</u> <u>+ main(args[ ] : String) : void</u>

```
public static void somar(double valor1, double valor2){
    // Processamento
    r = valor1 + valor2;
    // Saida
    JOptionPane.showMessageDialog(null, "Resultado: " + r);
}
```

```

public static double somar(double valor1, double valor2){
    // Processamento
    r = valor1 + valor2;
    return r;
}

```

### Principal

```

+ somar(valor1 : double, valor2 : double) : double
+ subtrair(valor1 : double, valor2 : double) : double
+ multiplicar(valor1 : double, valor2 : double) : double
+ dividir(valor1 : double, valor2 : double) : double
+ main(args[ ] : String) : void

```

## COM PARÂMETRO E COM RETORNO

```

// Entrada
n1 = Double.parseDouble( JOptionPane.showInputDialog("Digite o primeiro número:") );
n2 = Double.parseDouble( JOptionPane.showInputDialog("Digite o segundo número:") );

// Verifica a operação escolhida e invoca o respectivo método
switch (operacao) {
case 1:
    // Saida
    JOptionPane.showMessageDialog(null, "Resultado: " + somar(n1, n2));
    break;

case 2:
    // Saida
    JOptionPane.showMessageDialog(null, "Resultado: " + subtrair(n1, n2));
    break;

case 3:
    // Saida
    JOptionPane.showMessageDialog(null, "Resultado: " + multiplicar(n1, n2));
    break;

case 4:
    // Saida
    JOptionPane.showMessageDialog(null, "Resultado: " + dividir(n1, n2));
    break;
}

```

# EXERCÍCIOS

- Desenvolva versões dos projetos ControleEscolar2, Imc e Numeros organizando suas funcionalidade em métodos de acordo com os novos enunciados (e seus diagramas).



# EXERCÍCIOS

## ■ Projeto ControleEscolar2Metodo

Em um curso, a situação final do aluno pode ser “Aprovado”, “Aprovado na recuperação”, “Reprovado” ou “Reprovado por frequência” na recuperação e é baseada na média aritmética de duas notas e na frequência do aluno.

1. A frequência é obrigatória e excludente e deverá ser igual ou superior a 75.
2. A média mínima para aprovação é 6.0
3. Caso o aluno não obtenha a média mínima, uma nota maior ou igual a 6.0.

O usuário poderá verificar quantos alunos ele desejar.

## ■ A recuperação (item 3) será tratada em um método.

### Principal

```
+ recuperacao() : void  
+ main(args[ ] : String) : void
```

# EXERCÍCIOS

## ■ Projeto Imc

O IMC (índice de massa corporal) é um critério da Organização Mundial da Saúde para dar uma indicação sobre a condição de peso de uma pessoa adulta. A fórmula é  $IMC = \text{peso} / (\text{altura})^2$ . Faça um algoritmo que leia o peso e a altura de um adulto e mostre sua condição.

- a. Abaixo de 18.5 é considerado: Abaixo do peso.
- b. Entre 18.5 até 25 é considerado: Peso normal.
- c. Entre 25 até 30 é considerado: Acima do peso.
- d. Acima de 30 é considerado: Obeso.

O programa avaliará os dados de 50 pessoas.

### Principal

```
+ verificarImc(imc : double) : void  
+ main(args[ ] : String) : void
```

- ## ■ A verificação e apresentação das mensagens será executado pelo método verificarImc

# EXERCÍCIOS

## ■ Projeto Numeros

Leia dois números e a opção escolhida pelo usuário, sendo:

- 1 – Apresente a média aritmética dos números.
- 2 – Apresenta o menor número.
- 3 – Apresenta a diferença entre o maior e o menor número.
- 4 – Apresenta os números em ordem crescente.

O usuário poderá executar as funcionalidades do programa quantas vezes ele desejar.

Obs.: Considere que os dois números são diferentes.

## ■ Cada opção será implementada em uma método.

### Principal

```
+ media(valor1 : double, valor2 : double) : double  
+ menor(valor1 : double, valor2 : double) : double  
+ diferenca(valor1 : double, valor2 : double) : double  
+ crescente(valor1 : double, valor2 : double) : void  
+ main(args[ ] : String) : void
```