



Programação Orientada a Objetos

Introdução à Linguagem JAVA

Material produzido pelo Prof. Flávio Treib e adaptado pelo Prof. Adalto Selau Sparremberger

JAVA - Histórico

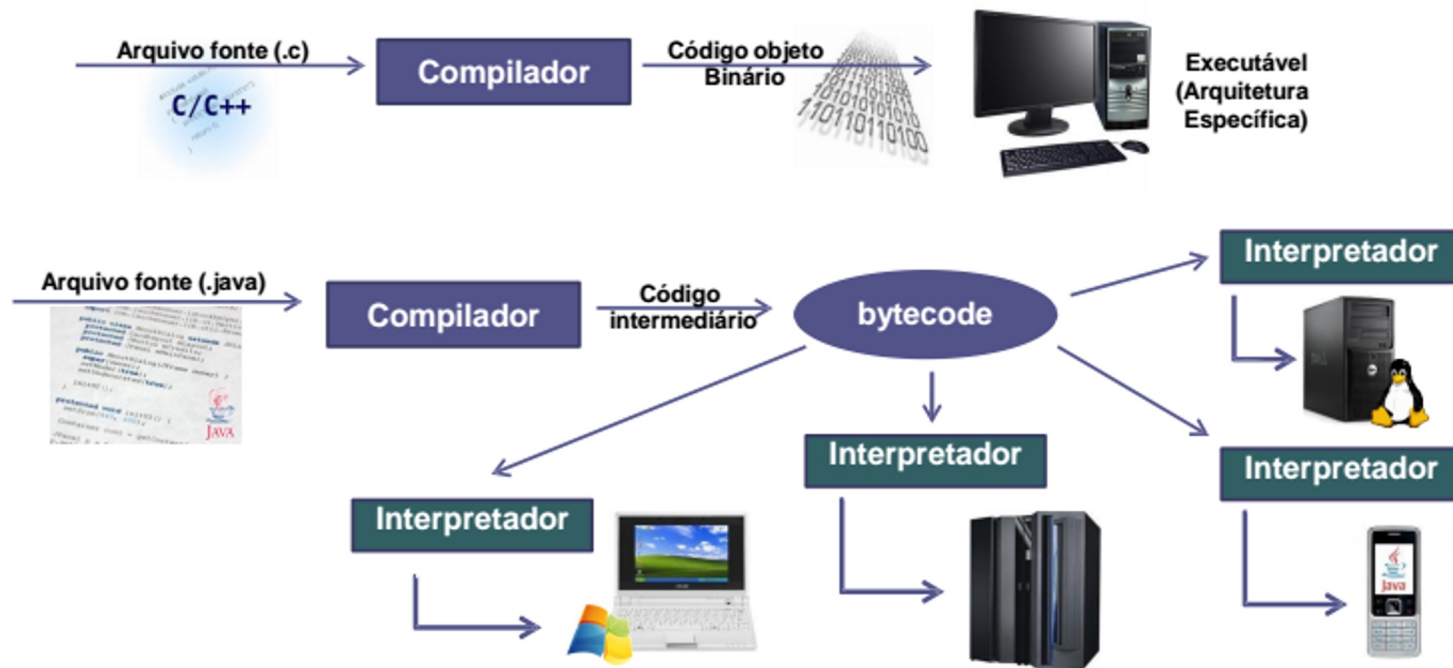
- 1991:
 - Com o crescente mercado de dispositivos inteligentes a Sun Microsystems decide financiar o Green Project:
 - linguagem simples;
 - pouca memória;
 - arquiteturas diferentes;
 - Linguagem devia se chamar Oak;
 - Java, por causa da origem do café da equipe;
- 1995: Apresentada formalmente;
- 2009: Sun comprada pela Oracle



JAVA

- linguagem orientada a objetos;
- simplificada em relação às anteriores;
- fortemente tipada;
- sem ponteiros;
- gerenciamento automático de memória;
- Independente;
- Interpretada;
- Portável.

JAVA



JAVA



PROGRAMAÇÃO JAVA

- Escrever classes;
- Representação de objetos reais ou abstratos;
- Aplicação em execução = conjunto de objetos, criados a partir das classes, trocando mensagens entre si;
- Saber como estruturar a aplicação;

JAVA

- Deriva da linguagem C, regras se assemelham
 - Blocos modularizados em métodos
 - Delimitados por chaves {}
 - Variáveis são declaradas antes de serem usadas
- Começa com pacotes (*packages*)
 - Dentro dos pacotes, as classes
 - Dentro das classes , os métodos, variáveis, constantes, etc.

JAVA – palavras reservadas

abstract	continue	para	new	switch
assert	default	goto	package	synchronized
boolean	do	if	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

JAVA – Tipos de dados primitivos

- Numéricos sem casa decimal:
 - byte (8 bits)
 - short (16 bits)
 - int (32 bits)
 - long (64 bits)
- Numéricos com casas decimais:
 - float (32 bits) - impreciso para números muito grandes
 - double (64 bits) - precisão dupla
- char (16 bits) → caractere unicode ('L' - '\u0058')
- boolean (1 bit) → true/false - não pode ser numérico, sempre minúsculo

JAVA – Strings

- Enquanto o tipo de char representa apenas 1 caractere, a representação de textos/palavras deverá ser feita pela classe String.

```
char digito = 'a';  
String nome = "Flavio";
```

- A classe String pode ser utilizada de forma similar aos tipos primitivos, mas os valores literais desse tipo são transcritos entre aspas e não entre apóstrofes.

JAVA – Variáveis

- Java é uma linguagem “fortemente tipada”. Cada variável pode armazenar um único tipo.
- Cada variável possui:
 - Um valor (pode ser null)
 - Um tamanho (definido pelo tipo)
 - Exemplos:
 - **int** numeroDias;
 - **char** opcao;
 - **float** cotacaoDolar;
 - **boolean** status;
 - **double** qtd1, preco1;

JAVA – Variáveis

- Dois detalhes importantes:
 - Usamos vírgula em países da América não-inglesa e na Europa, mas para representar valores decimais, em computação, usamos o ponto '.' como separador, e não vírgula!
 - Por padrão, o Java assume valores decimais como double. Pra especificar que é um 'float', coloque aquele 'f' ao final. Ou 'F'.
Ao final do tipo 'long' coloque 'l' ou 'L'.

JAVA – Constantes

- Constantes são atributos de uma classe que não mudam de valor. O modificador ***final*** indica que o atributo é imutável.

```
public class Matematica {  
    static final double PI = 3.14; //a constante não muda  
    static double areaCircunferencia(double r) {  
        return PI * r * r;  
    }  
    static double perimetroCircunferencia(double r){  
        return PI * r;  
    }  
}
```

JAVA – Operadores Aritméticos Binários

- Requerem dois operandos
- Operações aritméticas básicas
- Funciona com variáveis numéricas e literais
- Exemplos:

```
int x, y, z;
```

```
x = 229 + 23; // adição
```

```
y = 73 - 9; // subtração
```

```
z = 72 * 6; // multiplicação
```

```
x = y / z; // divisão
```

```
y = x % z; // resto da divisão
```

JAVA – Operadores Aritméticos Unários

- O operador ++ incrementa de 1
- O operador -- decrementa de 1
- Duas formas de utilização: pré e pós-fixada
- Exemplos:

`x = 2;`

`y = x++;` // pós-fixado => `x = 3` e `y = 2`

`x = 2;`

`y = ++x;` // pré-fixado => `x = 3` e `y = 3`

JAVA – Operadores Aritméticos

Operador	Significado	Exemplo
+	Adição	$a + b$
-	Subtração	$a - b$
*	Multiplicação	$a * b$
/	Divisão	a / b
%	Resto da divisão inteira	$a \% b$
-	Sinal negativo (- unário)	$-a$
+	Sinal positivo (+ unário)	$+a$
++	Incremento unitário	$++a$ ou $a++$
--	Decremento unitário	$--a$ ou $a--$

JAVA – Operadores Relacionais

Operador	Significado	Exemplo
==	Igual	$a == b$
!=	Diferente	$a != b$
>	Maior que	$a > b$
>=	Maior ou igual a	$a >= b$
<	Menor que	$a < b$
<=	Menor ou igual a	$a <= b$

JAVA – Operadores Lógicos

Operador	Significado	Exemplo
&&	E lógico	a && b
	Ou lógico	a b
!	not Negação	!a
^	Ou exclusivo	a^b

Ou exclusivo - se uma e apenas uma condição for verdadeira. Se duas forem verdadeiras resulta em false.

JAVA – Operador de Atribuição

- Atribuição simples (=)
- A expressão da direita é atribuída à variável que está à esquerda.

`x = 110;`

`y = x + 12;`

- Pode ser utilizado de forma encadeada

`x = y = z = 55;`

JAVA – Operadores Compostos

- O operador de atribuição pode ser combinado com qualquer operador binário.

`x = 110;`

`x += 12; // mesmo que x = x + 12`

`x /= 4; // mesmo que x = x / 4`

JAVA

- **Saída de Dados**

`System.out.println` // nova linha

“\n” – também serve para nova linha

`System.out.print` // mesma linha

- **Concatenação**

`String palavra1 = “Flavio ”;`

`String palavra2 = “Treib”;`

`System.out.println(palavra1 + palavra2);`

`System.out.println(“Nome: ” + palavra1);`

JAVA

- **Comentário de várias linhas:**

```
/**  
 *  
 * @author Flavio  
 */
```

- **Comentário de uma linha:**

```
//comentário de 1 linha
```

JAVA – Estrutura básica

- No java dividimos grupos de códigos em pacotes. Isto facilita a importação e reutilização destes códigos com funções e classes. O comando **package** referencia o pacote onde a classe atual se encontra. É o nome do pacote.

```
package aula1;
```


JAVA

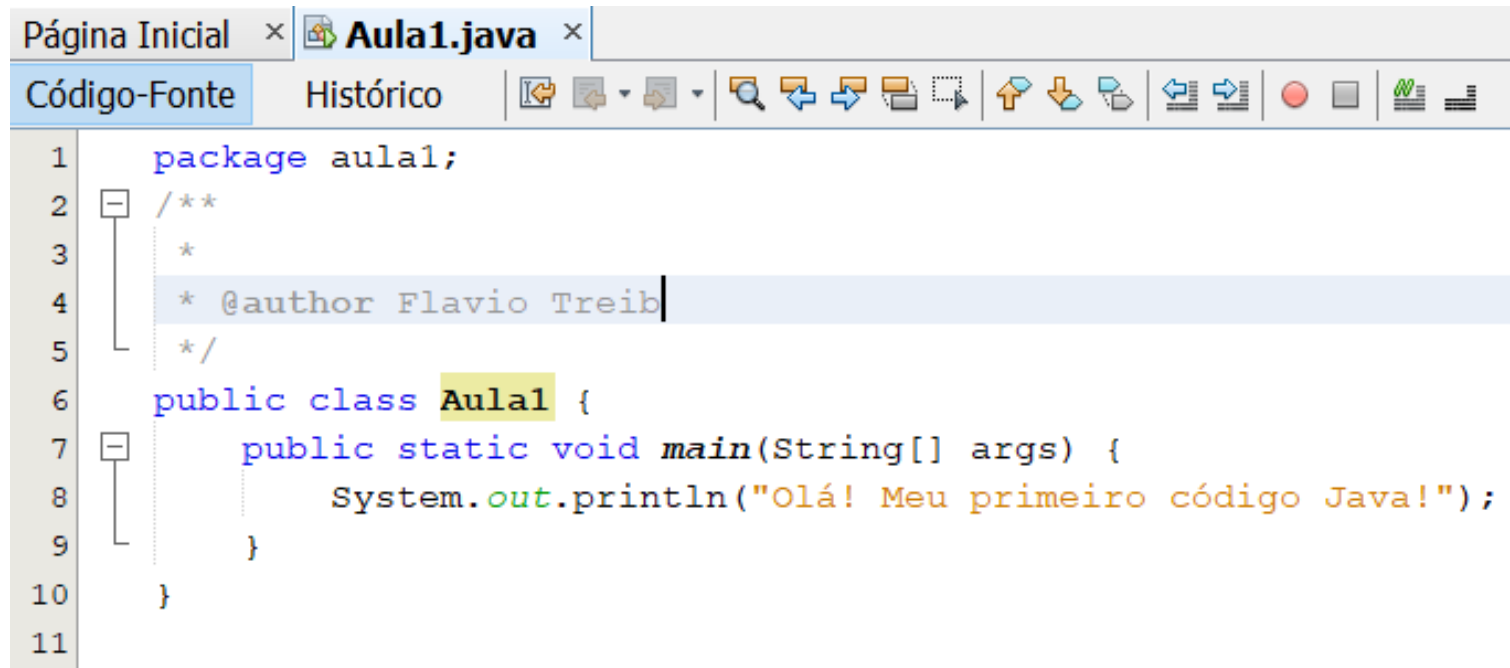
- Uma classe pública, SEMPRE delimitada por chaves.

```
public class Aula1{  
    //comandos  
}
```

- Método principal do código Java. Necessário pois todo código Java é executado por um programa da máquina virtual. Este programa procura na classe que você passou como parâmetro, um método chamado "main" e que seja estático (static) e público (public). SEMPRE delimitada por chaves.

```
public static void main(String[]) args){  
    // comandos  
}
```

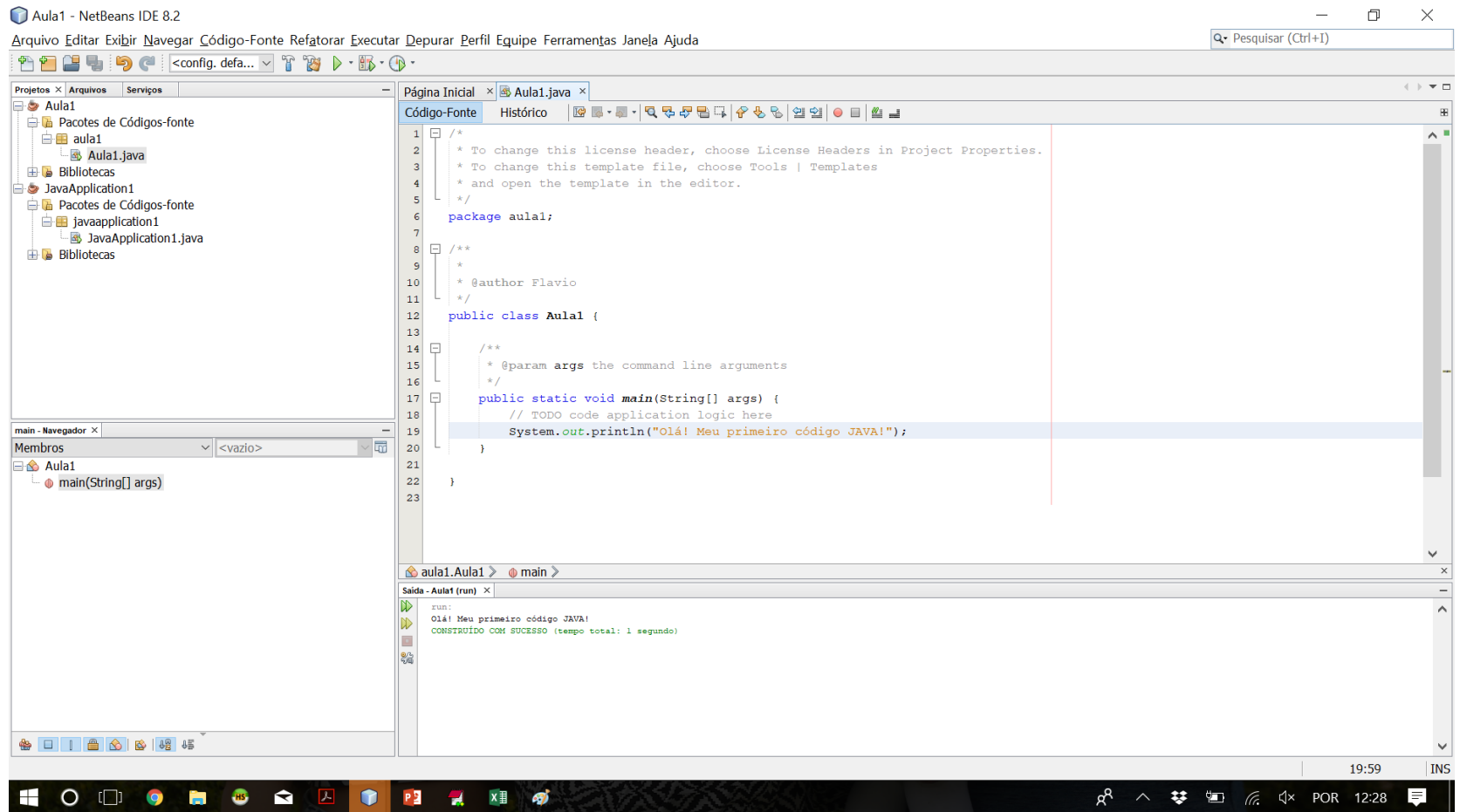
JAVA – Estrutura básica



The image shows a screenshot of an IDE window with the title bar 'Página Inicial x Aula1.java x'. The 'Código-Fonte' tab is active. The code is as follows:

```
1 package aula1;
2 /**
3  *
4  * @author Flavio Treib
5  */
6 public class Aula1 {
7     public static void main(String[] args) {
8         System.out.println("Olá! Meu primeiro código Java!");
9     }
10 }
11
```

NetBeans



NetBeans 13

- Abra seu NetBeans.
 - Disponível em: <https://netbeans.apache.org/download/index.html>
- Vá em Arquivo, depois em Novo Projeto.
- Você verá uma série de opções.
- São as possibilidades de se programar em Java, os tipos de aplicações etc.
- No nosso caso é Java >> Aplicação Java >> Próximo
- Coloque o nome no projeto >> Finalizar

Exercícios

1. Frase na tela - Implemente um programa que escreve na tela a frase "O primeiro programa a gente nunca esquece!".
2. Etiqueta - Elabore um programa que escreve seu nome completo na primeira linha, seu endereço na segunda, e o CEP e telefone na terceira.
3. Implemente um programa que desenhe um Quadrado na tela
4. Implemente um programa que desenhe um "pinheiro" na tela

Exercícios

5. Tabela de notas - Escreva um programa que produza a seguinte saída na tela:

ALUNO(A)	NOTA
=====	=====
ALINE	9.0
MÁRIO	DEZ
SÉRGIO	4.5
SHIRLEY	7.0

6. Elabore um programa que mostre o seguinte menu na tela:

Cadastro de Clientes

0 - Fim

1 - Inclui

2 - Altera

3 - Exclui

4 - Consulta

Opção:

