



Programação 2

Introdução a aplicativos Java – entrada/saída e operadores

Prof. Domingo Santos
domingos.santos@upe.br

O paradigma de Orientação a Objeto no desenvolvimento de software

Visão aprofundada dos conceitos de orientação a objetos:

- programação em java
- classes
- mensagem
- herança
- encapsulamento
- abstração
- modularidade
- reutilização
- metodologia
- definição de objetos estruturas
- atributos
- serviços e transformação do modelo analítico para o de especificação

Técnicas de programação e modularização.

Livro(s) texto(s):

- **DEITEL, Paul J; DEITEL, Harvey M. Java: como programar . 8. ed. São Paulo: Pearson Education, 2010**
- SIERRA, Kathy. Java. 2. ed. Rio de Janeiro: Alta Books, 2010.
- DEITEL, Harvey M.; DEITEL, Paul J. C++/ como programar. 5. ed. São Paulo: Pearson
- Prentice Hall, 2006

Livros de referência:

- HORSTMANN, Cay S.; CORNELL, Gary. Core Java. São Paulo: Pearson Prentice Hall, 2010.
 - ASCENCIO, Ana Fernanda Gomes; ARAÚJO, Graziela Santos de. Estrutura de dados:
 - algoritmos, análise da complexidade e implementações em JAVA e C/C++. São Paulo: 2010
 - ASCENCIO, Ana Fernanda Gomes. Fundamentos da programação de computadores/
 - algoritmos, PASCAL, C/C++ (padrão ANSI) e JAVA. 3. ed. São Paulo: Pearson Education do Brasil, 2012
 - SIERRA, Kathy. Java. 2. ed. Rio de Janeiro: Alta Books, 2010.
-

Forma de Avaliação

- Prova 1:
 - Prova Teórica
 - 50% da nota final

 - Prova 2:
 - Projeto sobre o tema
 - Em grupo
 - 50% da nota final
-

- Chamada:
 - 5 minutos após o início da aula
- Qualquer pergunta merece atenção, não guarde a dúvida para você
- Foco na aula!
 - Para avançarmos mais rápido, e largar cedo
- Classroom

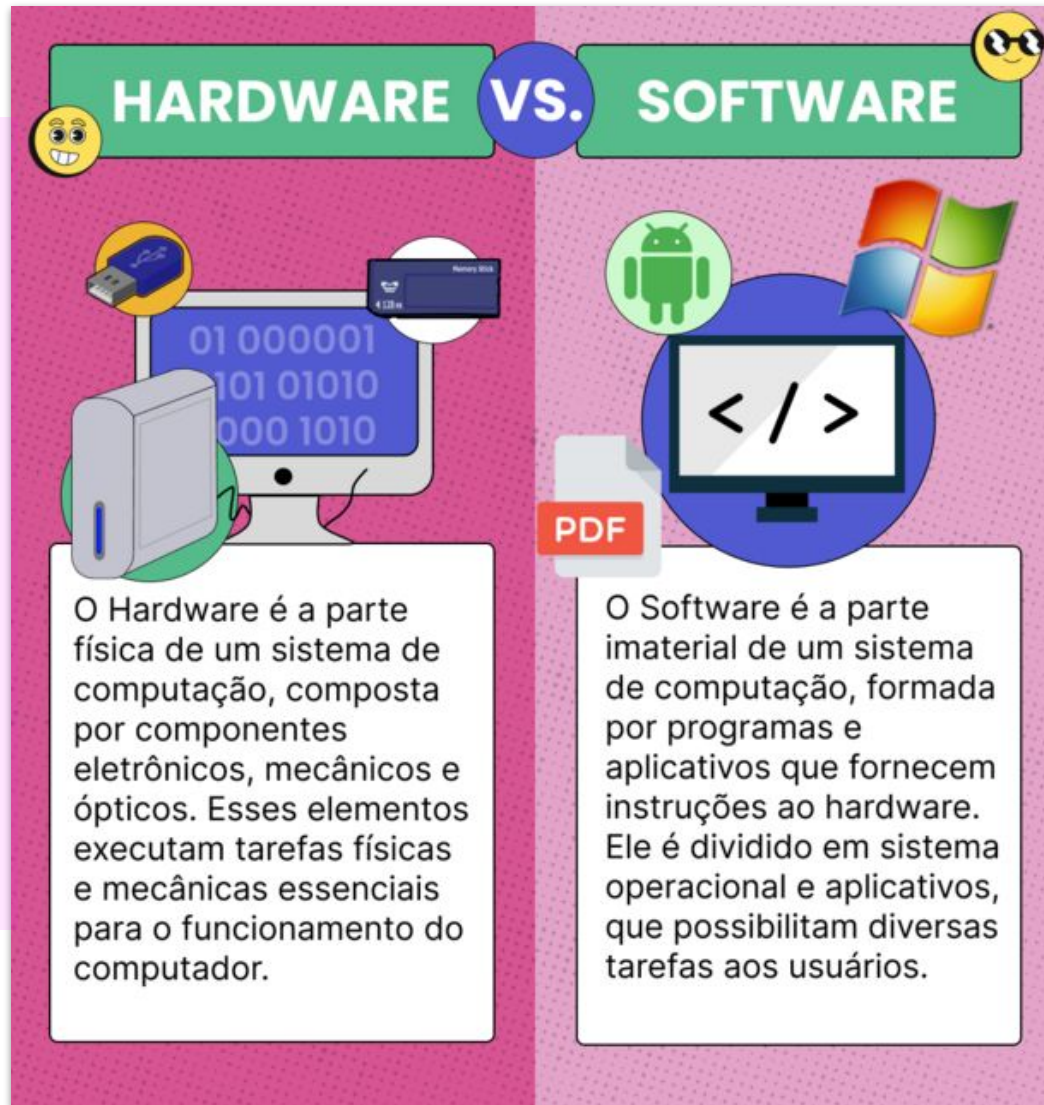


- Considerações iniciais:
 - Hardware x software
 - Hierarquia de dados
 - Organização do computador
 - Instrução para computadores
- Introdução ao Java:
 - Instalação
 - Primeiro programa
 - Conceitos de memória
 - Aritimética
 - Tomada de decisão

Considerações iniciais

Hardware x software

- Mouse
- Teclado
- Monitor
- Unidades flash USB



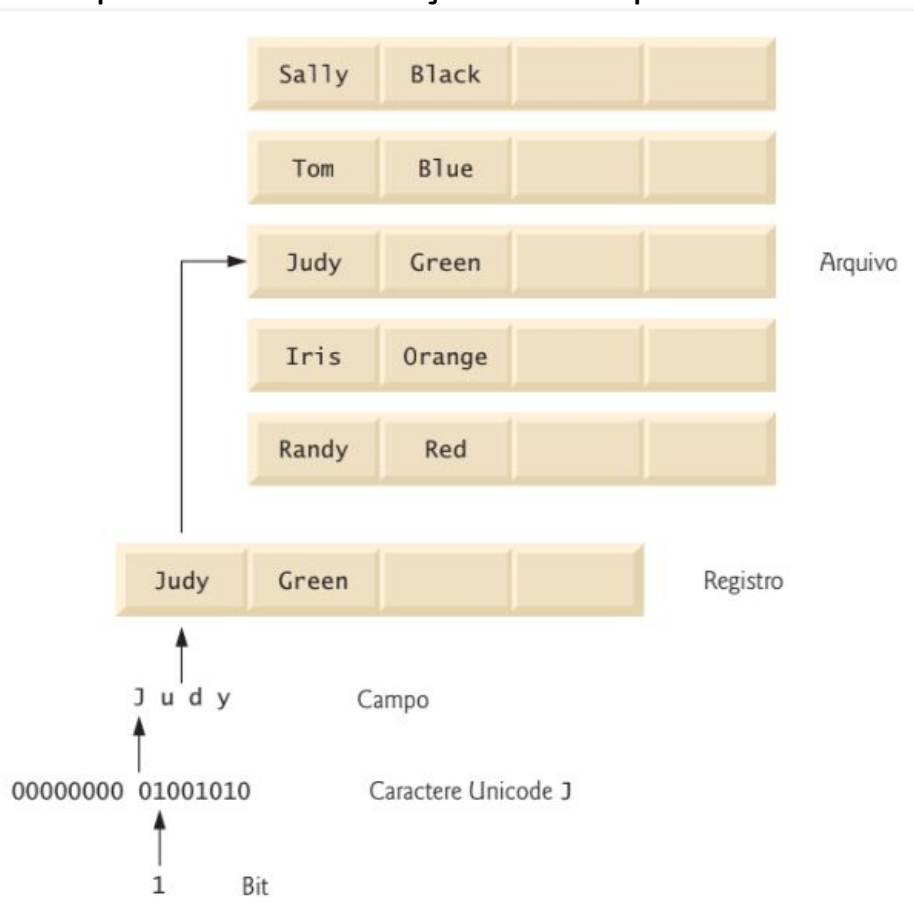
- Windows
- Word
- League of legends
- Antivirus

Organização do computador - Unidades lógicas

Unidade de entrada	Dispositivos de entrada e as coloca à disposição de outras unidades para serem processadas. Exemplo: Teclado e Mouse
Unidade de saída	Pega as informações que o computador processa e as coloca em vários dispositivos de saída. Exemplo: Caixas de som e monitor
Unidade de memória	Essa seção de “armazenamento” de acesso rápido e de relativa baixa capacidade. em geral, são perdidas quando o computador é desligado (Volátil). Memória RAM.
Unidade de aritmética e lógica (ALU)	Executa cálculos como adição, subtração, multiplicação e divisão.
Unidade de processamento central (CPU)	Essa seção “administrativa” coordena e supervisiona a operação das outras seções. Sistemas com multi núcleos.
Unidade de armazenamento secundária	Armazenamento de longo prazo e de alta capacidade. É mantida mesmo após o desligamento (não volátil). Memória HD ou SSD (Disco de Estado Sólido)

Hierarquia de dados

Exemplo de transformação de bits para texto



Bits	O menor item de dados em um computador
Caracteres	Dígitos, letras e símbolos. <ul style="list-style-type: none"> ASCII (American Standard Code for Information Interchange) UTF-8 (8-bit Unicode Transformation Format)

UTF-8 Exemplos

128	Ç	144	É	160	á	176	ð	192	Ł	208	Š	224	α	240	≡
129	Û	145	Ê	161	â	177	é	193	ł	209	ś	225	β	241	±
130	é	146	Æ	162	ó	178	ë	194	Ł	210	ŧ	226	Γ	242	≥
131	â	147	ô	163	û	179	ï	195	ł	211	ŋ	227	π	243	≤
132	ä	148	õ	164	ü	180	ï	196	—	212	Ł	228	Σ	244	ƒ
133	à	149	ò	165	ñ	181	ı	197	†	213	ƒ	229	σ	245	Ƶ
134	á	150	ú	166	ı	182	ı	198	‡	214	ŋ	230	μ	246	÷
135	ç	151	û	167	°	183	ı	199	‡	215	ŋ	231	τ	247	≈
136	ê	152	ÿ	168	ı	184	ı	200	Ł	216	‡	232	Φ	248	°
137	ë	153	Û	169	ı	185	ı	201	ŋ	217	ı	233	Θ	249	.
138	è	154	Ü	170	ı	186	ı	202	Ł	218	ı	234	Ω	250	.
139	ı	155	ı	171	ı	187	ı	203	ŋ	219	ı	235	δ	251	√
140	ı	156	ı	172	ı	188	ı	204	ŋ	220	ı	236	∞	252	∞
141	ı	157	ı	173	ı	189	ı	205	ı	221	ı	237	φ	253	²
142	Ä	158	ı	174	ı	190	ı	206	ŋ	222	ı	238	ε	254	■
143	Ä	159	f	175	ı	191	ı	207	ı	223	ı	239	ı	255	

Source: www.LookupTables.com

Hierarquia de dados

Campos	Um campo é um grupo de caracteres ou bytes que transmitem um significado. Exemplo: nome de uma pessoa
Registros	Vários campos relacionados. Exemplo: <ul style="list-style-type: none">• Número de identificação de funcionário (um número inteiro).• Nome (uma string de caracteres).• Endereço (uma string de caracteres).
Arquivos	<ul style="list-style-type: none">• Um arquivo é um grupo de registros relacionados.• Dados arbitrários em formatos arbitrários.
Banco de dados	Conjuntos de dados organizados para fácil acesso e manipulação. Exemplo: SQL
Big data	Conjuntos de dados extremamente grandes e complexos

Criar instruções para o computador

Linguagens de máquina:

- Linguagem do computador definida pelo seu projeto de hardware.
- Extremamente complicada para humanos
- Encadeamento de números



Linguagens assembly

- Códigos mnemônicos para realizar operações específicas
- Considerada como uma linguagem de baixo nível, dada a sua complexidade para atividades simples
- Foram desenvolvidos para converter os primeiros programas de linguagem assembly em linguagem de máquina

load	basepay
add	overpay
store	grosspay

Linguagens de alto nível

- Acelerar o processo de programação
- Compiladores convertem os programas de linguagem de alto nível em linguagem de máquina. Exemplo C
- Interpretadores, código-fonte é executado linha por linha por um programa chamado interpretador, sem a necessidade de compilação prévia. Exemplo Python

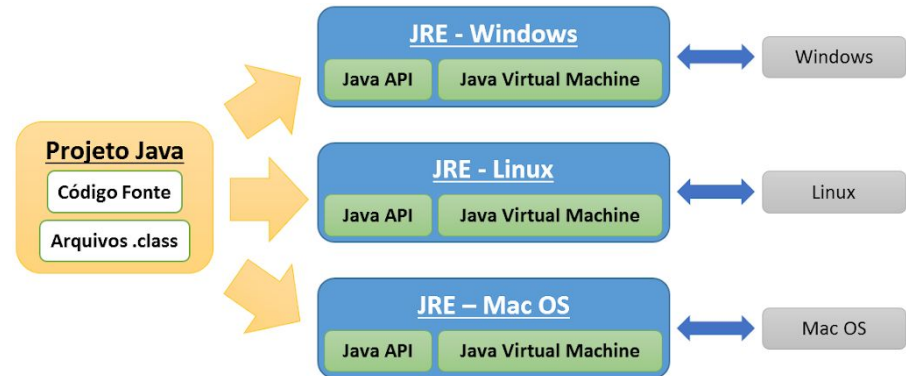
Criar instruções para o computador

E o JAVA?

- Linguagem de programação compilada e interpretada, conhecido como compilação just-in-time (JIT).
- Bytecode em Java é uma forma de código intermediário gerado pelo compilador
- Em seguida, esse bytecode é interpretado pela **máquina virtual Java (JVM)**, que converte o bytecode em código de máquina nativo durante a execução do programa
- É necessário instalar a JVM

Vamos utilizar o Java como linguagem de programação da cadeira

- A estrutura do Java foi feita para Orientação a Objetos (OO)
- Livros importantes em OO usam java
- Ainda é uma das principais linguagens de programação da atualidade
- Possui muitas possibilidades, programação:
 - Desktop
 - Android nativo
 - Web



**He was forced to use
Java**



Introdução ao Java

IDEs:

- **Eclipse** (www.eclipse.org)
- **NetBeans** (www.netbeans.org)
- **IntelliJ IDEA** (www.jetbrains.com)

Instalação Eclipse:

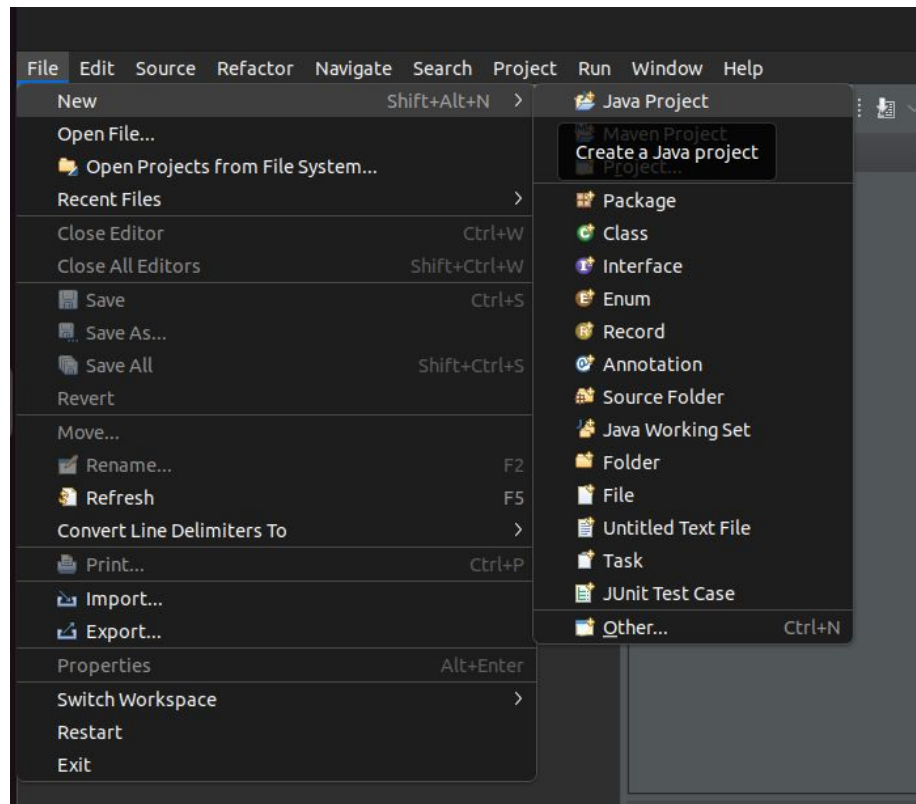
<https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers>

Máquina Virtual Java (JVM)

<https://www.java.com/pt-BR/download/manual.jsp>

Primeiro Programa em Java

- Entre na IDE Eclipse
- Defina a pasta que será seu workspace
- Criar um projeto java novo:
 - File -> new -> Java Project



Primeiro Programa em Java

- Defina o nome do projeto
 - HelloWorld
- Desmarcar o Módulo: module-info.java
- Clique em finish

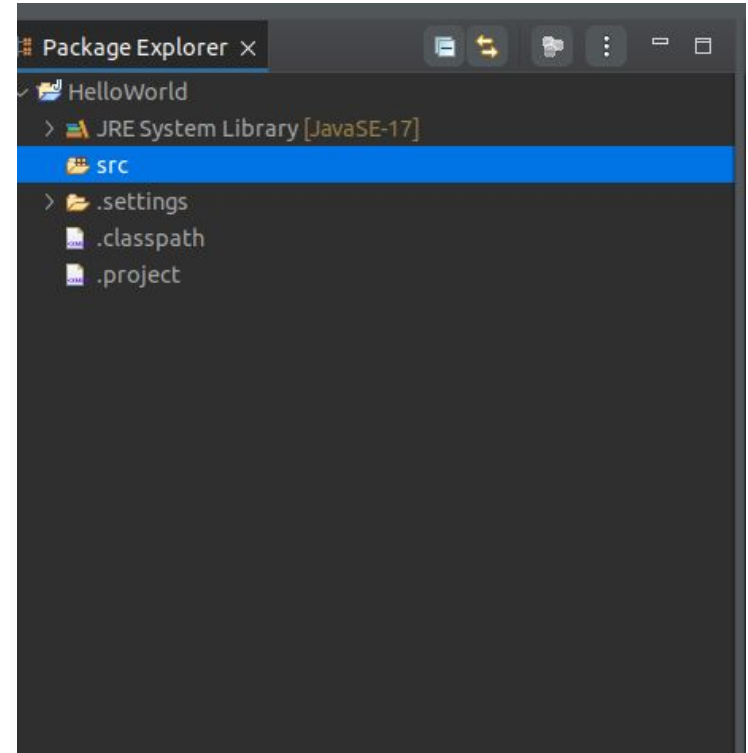
The screenshot shows the 'New Java Project' dialog box in the Eclipse IDE. The dialog is titled 'New Java Project' and has a close button in the top right corner. It contains several sections for configuring a new Java project:

- Create a Java Project:** This section has a subtitle 'Create a Java project in the workspace or in an external location.' and a folder icon. The 'Project name:' field is set to 'HelloWorld' and is highlighted with a red rectangle.
- Location:** The 'Use default location' checkbox is checked. The 'Location:' field shows the path '/home/domingossj/eclipse-workspace/HelloWorld' with a 'Browse...' button to its right.
- JRE:** This section has three radio buttons: 'Use an execution environment JRE:' (selected), 'Use a project specific JRE:', and 'Use default JRE 'jre' and workspace compiler preferences'. The first option is set to 'JavaSE-17'. There is a 'Configure JREs...' link to the right.
- Project layout:** This section has two radio buttons: 'Use project folder as root for sources and class files' and 'Create separate folders for sources and class files' (selected). There is a 'Configure default...' link to the right.
- Working sets:** This section has a checkbox 'Add project to working sets' (unchecked) and a 'New...' button. Below it, the 'Working sets:' field is empty with a dropdown arrow and a 'Select...' button.
- Module:** This section has a checkbox 'Create module-info.java file' (unchecked) which is highlighted with a red rectangle. Below it, the 'Module name:' field is empty and there is a 'Generate comments' checkbox (unchecked).

At the bottom of the dialog, there is a help icon (?) and four buttons: '< Back', 'Next >', 'Cancel', and 'Finish'.

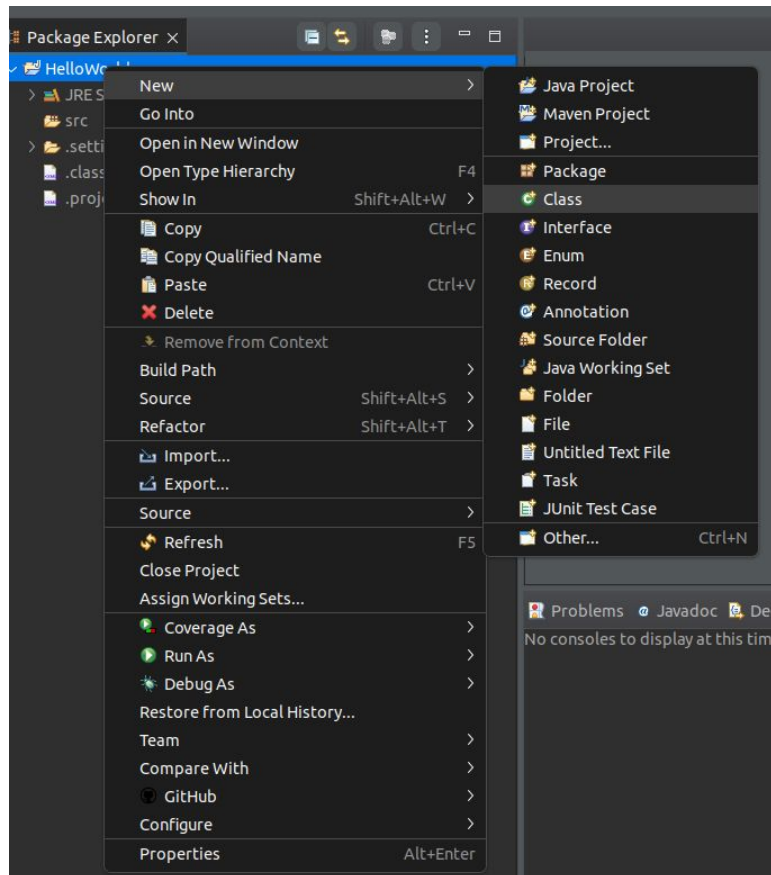
Primeiro Programa em Java

- Esqueleto do projeto criado
- JRE system library
- pasta src
- .settings
- classpath
- .project

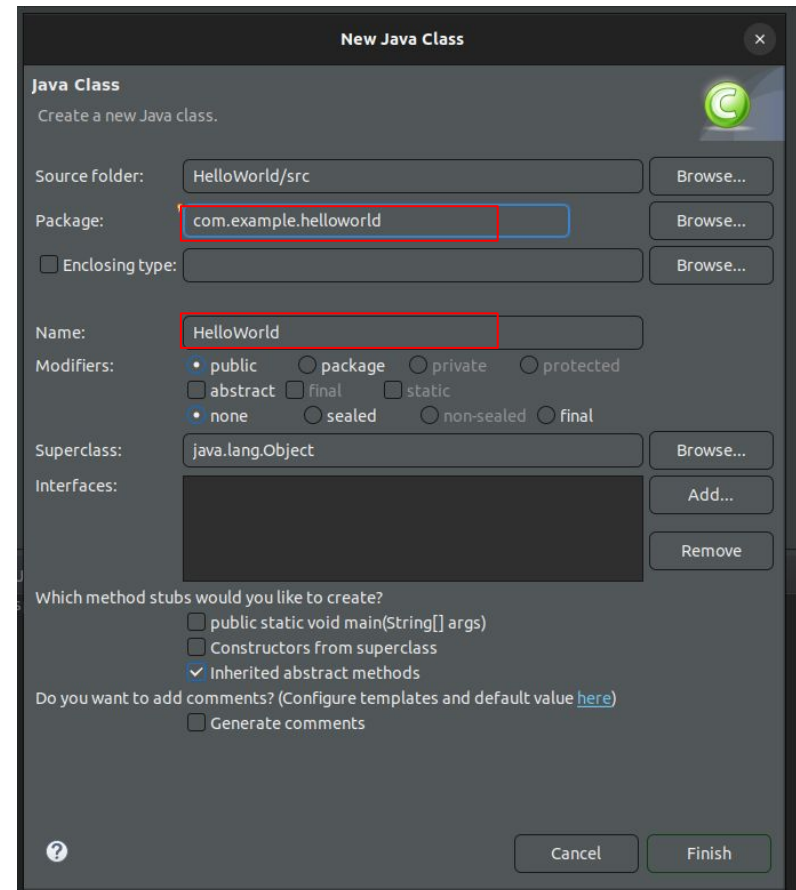


Primeiro Programa em Java

- Criar a classe principal
 - botão direito no nome do projeto
 - New -> Class

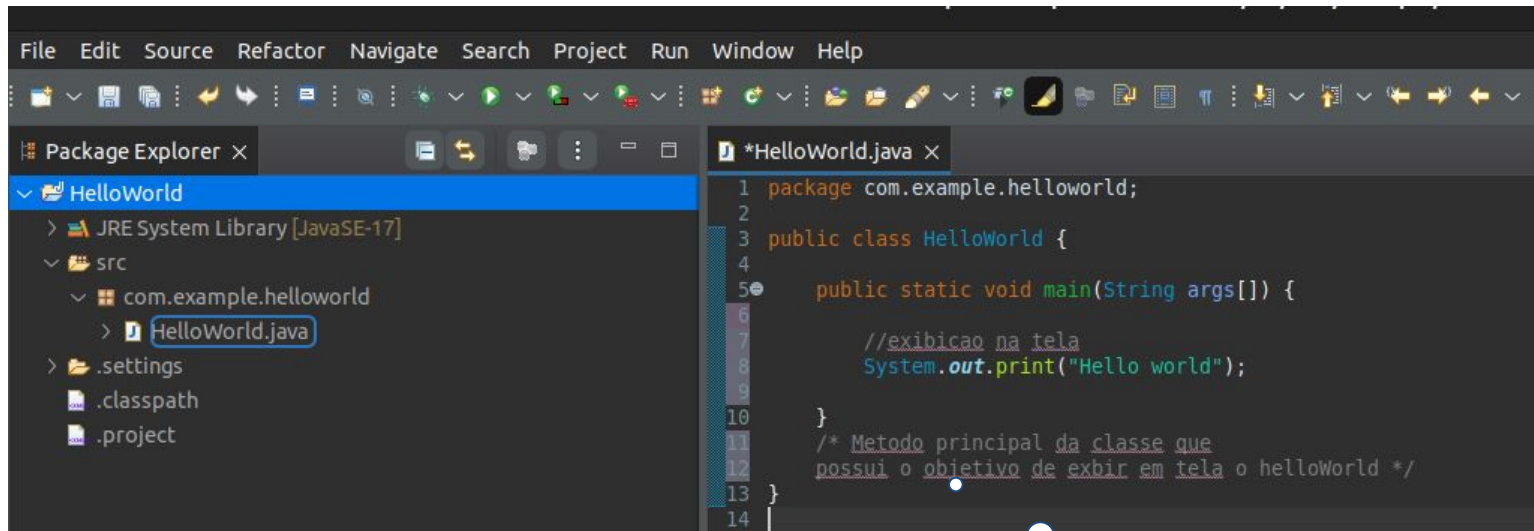


- Definição da Classe principal
 - Package -> Name -> Finish



Primeiro Programa em Java

- Editando a Classe principal
 - `public static void main (String arg[])`
 - `public class`
 - Chaves `{}`
 - Comentários:
 - `//`
 - `/* */`



```
1 package com.example.helloworld;
2
3 public class HelloWorld {
4
5     public static void main(String args[]) {
6
7         //exibicao na tela
8         System.out.print("Hello world");
9
10    }
11    /* Metodo principal da classe que
12       possui o objetivo de exibir em tela o helloWorld */
13 }
14
```

Algumas organizações exigem que todo programa comece com um comentário que informa o objetivo e o autor dele, a data e a hora em que foi modificado pela última vez.

Primeiro Programa em Java

Indentação e espaçamento

Difícil Legibilidade ❌

```
*HelloWorld.java x
1 package com.example.helloworld;
2
3 public class HelloWorld {
4 public static void main(String args[]) {
5 System.out.print("Hello world");
6 }
7 }
8
```

Melhor Legibilidade ✅

```
HelloWorld.java x
1 package com.example.helloworld;
2
3 public class HelloWorld {
4
5     public static void main(String args[]) {
6         System.out.print("Hello world");
7     }
8 }
9
```

Nome da Classe

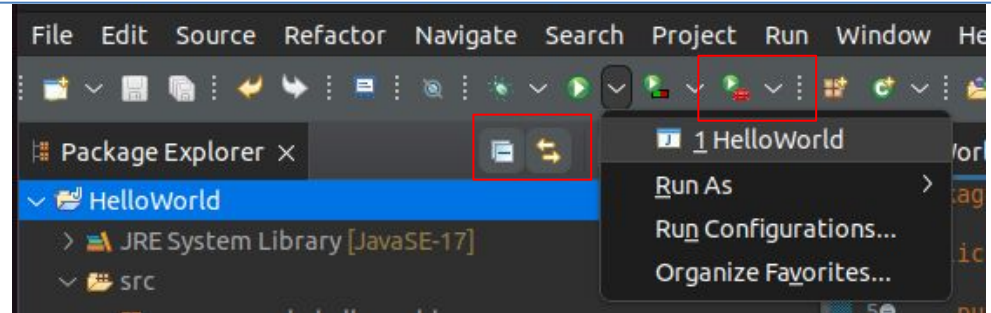
CamelCase: Usado em classes e métodos. Nesse formato, cada palavra no nome começa com letra maiúscula, exceto a primeira palavra. Por exemplo:

- PagamentoContas
- LoadCredentials
- UserUpdate

Primeiro Programa em Java

Execução (crt + f11)

- resultado visível no console



Verifique a diferença de resultado entre

- `System.out.println("Hello world");`
- `System.out.print("Hello world");`
- Dica:

- Faça o teste :

```
//exibicao na tela
System.out.print("Hello world");
System.out.print("|tudo bem pessoal?");
```

- Em seguida:

```
//exibicao na tela
System.out.println("Hello world");
System.out.println("tudo bem pessoal?");
```

- Será que o resultado é o mesmo do exemplo acima?

```
//exibicao na tela
System.out.println("Hello world\ntudo bem pessoal?");
```

```
//exibicao na tela
System.out.printf("%s\n%s", "Hello world", "tudo bem pessoal?");
```

A barra invertida (\) é um **caractere de escape**, significado especial no print e println. Outros exemplos comuns:

- `\t`
- `\"`

Primeiro Programa em Java

Declarações de import

- conjunto de classes predefinidas que você pode reutilizar
- Essas classes são agrupadas em pacotes.
- Chamados de grupos de classes relacionadas e, coletivamente, são chamadas de biblioteca de classes Java, ou Java Application Programming Interface (Java API).
- Java API documentação: <http://docs.oracle.com/javase/7/docs/api/index.html>
- `import java.util.Scanner`: Ler informação do teclado

Primeiro Programa em Java

Ler informação do teclado

- Criar uma nova classe chamada Adicao
- escrever o seguinte código
-

Tipos primitivos disponíveis: int, boolean, byte, char, short, long, float e double

```
package com.example.helloworld;
import java.util.Scanner;

public class Adicao {

    public static void main(String[] args) {
        int number1; // primeiro número a somar
        int number2; // segundo número a somar
        int sum; // soma de number1 e number2

        Scanner input = new Scanner(System.in);

        System.out.print("Enter first integer: "); // prompt
        number1 = input.nextInt(); // lê o primeiro número fornecido pelo usuário

        System.out.print("Enter second integer: "); // prompt
        number2 = input.nextInt(); // lê o segundo número fornecido pelo usuário
        sum = number1 + number2; // adiciona números e, então, armazena o total na soma
        System.out.printf("Sum is %d", sum); // exibe a soma
    }
}
```


Conceitos de memória

```
int number1; // primeiro número a somar
int number2; // segundo número a somar
int sum; // soma de number1 e number2

System.out.print("Enter first integer: ");
number1 = input.nextInt(); // lê o primeiro

System.out.print("Enter second integer: ");
number2 = input.nextInt(); // lê o segundo

sum = number1 + number2; // adiciona número
System.out.printf("Sum is %d", sum); // exi
```

Os nomes de variável como number1, number2 e sum na verdade correspondem às posições na memória do computador. Cada variável tem um **nome**, um **tipo**, um **tamanho** (em bytes) e um **valor**.

O número digitado pelo usuário é colocado em uma localização de memória correspondente ao nome number1 (**Processo Destrutivo**). Suponha que seja number1 = 45 e number2 = 72.

Realiza a soma e então substitui qualquer valor anterior de sum.

number1

45

number2

72

sum

117

Operadores

Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Resto	%

- A **divisão de inteiros** produz um quociente inteiro. Por exemplo, a expressão $7 / 4$ é avaliada como 1 e a expressão $17 / 5$, como 3. Qualquer parte fracionária na divisão de inteiros é simplesmente truncada (isto é, descartada)
- Os parênteses são utilizados para agrupar termos em expressões Java da mesma maneira como em expressões algébricas. Por exemplo: $a * (b + c)$

Regras de precedência de operador

Multiplicação Divisão Resto	Avaliado primeiro. Se houver vários operadores desse tipo, eles são avaliados da esquerda para a direita.
Adição Subtração	Avaliado em seguida. Se houver vários operadores desse tipo, eles são avaliados da esquerda para a direita.
Atribuição (=)	Avaliado por último.

Regras de precedência de operador

Em qual a ordem que será realizado a seguinte operação?

```
z = p * r % q + w / x - y;
```

Principais tipos primitivos em JAVA

Nome	Tamanho	Range de valores
int	32 bits	-2,147,483,648 a 2,147,483,647
float	32 bits	Aproximadamente 6-7 dígitos decimais
double	64 bits	Aproximadamente 15 dígitos decimais
char	16 bits	'\u0000' (0) a '\uffff' (65,535)
boolean	1 bit	true ou false.

Existem ainda outros tipos: byte, short, long, Big Decimal

- Uma condição é uma expressão que pode ser true ou false
- Tomar uma decisão com base no valor de uma condição, por meio de ifs !
- Se a condição em uma estrutura if for verdadeira, o corpo da estrutura if é **executado**. Se a condição for falsa, o corpo **não é executado**.

```
if(condição){
    //comando a ser executado SE a condição for cumprida
}
```

- Operadores de igualdade e relacionais

Operador algébrico	Operador de igualdade ou relacional Java	Exemplo de condição em Java	Significado da condição em Java
<i>Operadores de igualdade</i>			
=	==	x == y	x é igual a y
≠	!=	x != y	x é não igual a y
<i>Operadores relacionais</i>			
>	>	x > y	x é maior que y
<	<	x < y	x é menor que y
≥	>=	x >= y	x é maior que ou igual a y
≤	<=	x <= y	x é menor que ou igual a y

- Criar uma nova classe chamar Comparacao

```
package com.example.helloworld;
import java.util.Scanner; // programa utiliza a classe Scanner

public class Comparacao {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int number1, number2; // primeiro e segundo número a comparar

        System.out.print("Enter first integer: "); // prompt
        number1 = input.nextInt(); // lê o primeiro número fornecido pelo usuário

        System.out.print("Enter second integer: "); // prompt
        number2 = input.nextInt(); // lê o segundo número fornecido pelo usuário

    } // fim do método main

} // fim da classe Comparacao
```

Desafio !

1- Identifique e corrija os erros em cada uma das seguintes instruções:

- A. `if (c < 7);`
`System.out.println("c is less than 7");`
- B. `if (c => 7)`
`System.out.println("c is equal to or greater than 7");`

2- Supondo que $x = 2$ e $y = 3$, o que cada uma das instruções a seguir exibe?

- A. `System.out.printf("x = %d%n", x);`
- B. `System.out.printf("Value of %d + %d is %d%n", x, x, (x + x));`
- C. `System.out.printf("x =");`
- D. `System.out.printf("%d = %d%n", (x + y), (y + x));`

3- Declare a ordem de avaliação dos operadores em cada uma das seguintes instruções Java e mostre o valor de x depois que cada instrução é realizada:

- A. $x = 7 + 3 * 6 / 2 - 1$
 - B. $x = 2 \% 2 + 2 * 2 - 2 / 2;$
 - C. $x = (3 * 9 * (3 + (9 * 3 / (3))));$
-