

Linguagens de Programação

Java Orientado a Objetos

Prof. Waldeck Lindoso Jr.

Apresentação da disciplina



- **Apresentação do professor**

- Professor da Escola Técnica Estadual Porto Digital desde 2021
- Área de pesquisa geral: Métodos formais (UFRPE)
- Área de pesquisa específica: Especificação visual de propriedades para projetos robóticos

- **Conteúdo e Material**

- Link: https://drive.google.com/drive/folders/1haj1qfQx47D4xs-0La6Q_ekWFdidXYaPhSs4lkWXCfq8_Z7bgHUFEIwPOr8ERWVDE0xVrRuV?usp=sharing

- **Dúvidas? Interrompam a vontade...**



Horário das aulas



Horário / Dias	Seg	Ter	Qua	Qui	Sex
18:40 - 19:20	LP		LP		
19:20 - 20:00	LP		LP		
20:00 - 20:40	LP				LP
20:40 - 21:20					LP

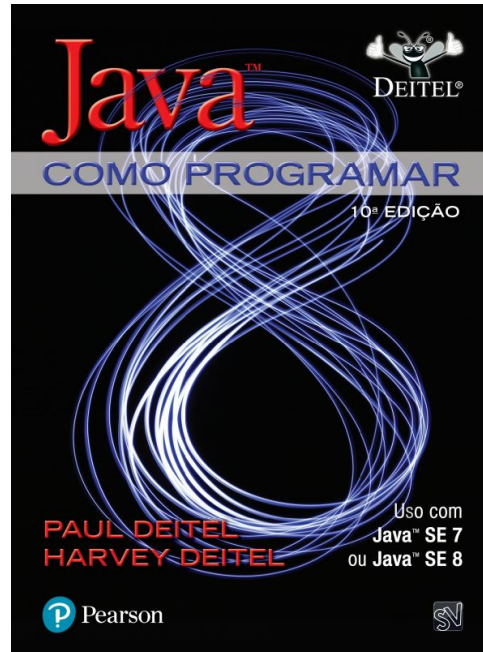
Ementa

Introdução aos conceitos de programação orientada a objetos, Tipos abstratos de dados, Classificação, Polimorfismo, Herança, Conceitos de objetos, Conceitos de Classes, Subclasses, Linguagem orientada a objetos, Aplicação de técnicas de programação orientada a objetos, Linguagem JAVA.



Literatura

DEITEL, Harvey M.; DEITEL, Paul J. **Como programar en Java**. Pearson educación, 2003.



Objetivos da disciplina

Ao término da disciplina os alunos serão capazes de:

- Desenvolver soluções em Java de acordo com o paradigma orientado a objetos
- Utilizar com destreza pelo menos uma ferramenta de desenvolvimento interativo (IDE), que neste caso será o Eclipse
- Testar e depurar programas em Java
- Entender as diferenças entre o paradigma orientado a objetos e outros paradigmas de programação



Avaliações

- Nota 1 = AT1 + AT2 (5pts + 5pts)
 - AT1 = Listas/atividades
 - AT2 = Listas/atividades
- Nota 2 = Prova (10pts)



Linguagens de programação

- Quantas linguagens de programação vocês conhecem?



Swift



python™



Perl



Linguagens de programação

- E quantos Frameworks em cima dessas linguagens vocês conhecem?



Linguagens de programação

E qual seria a melhor linguagem de programação?



Linguagens de programação

Nenhuma delas ...
... e todas elas!



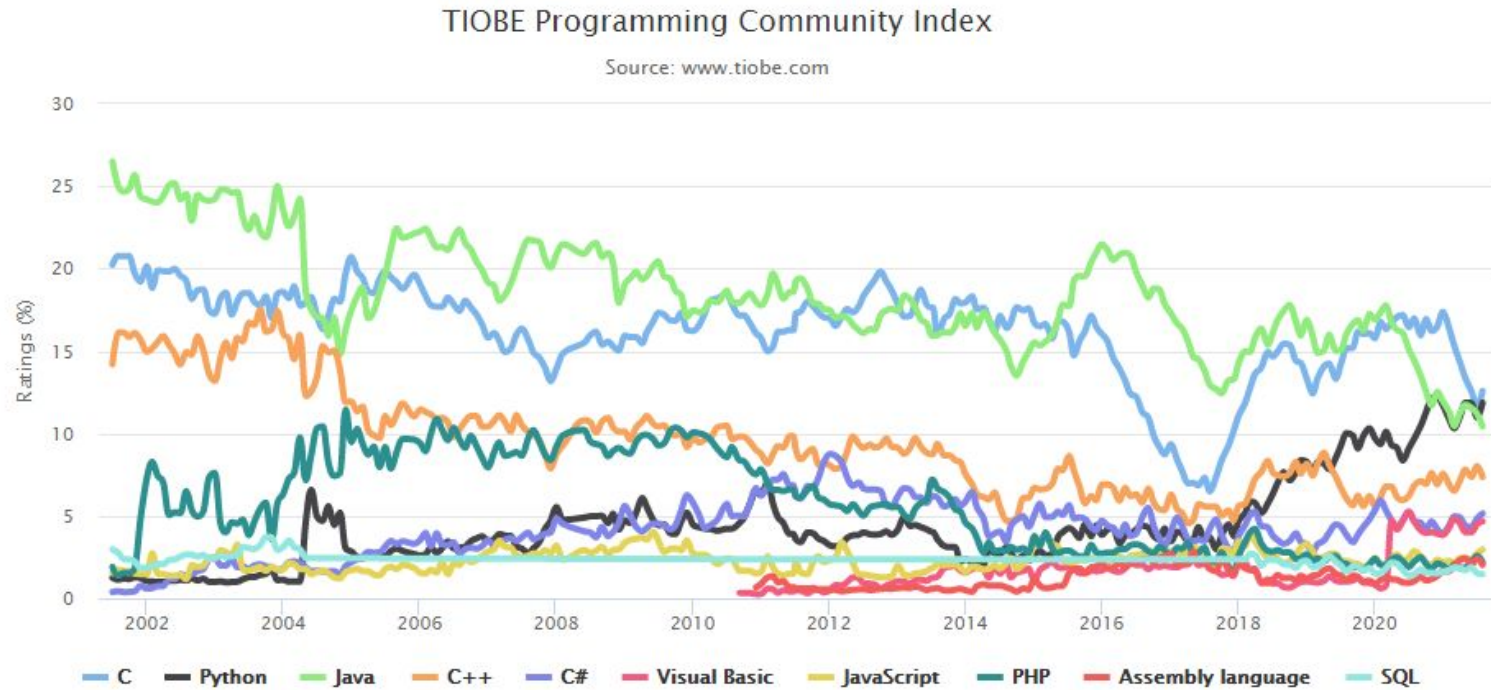
Porque Java?

1. Java é fácil de aprender
2. Excelente representante do mundo orientado a objetos
3. Linguagem aberta e livre
4. Poderosas ferramentas gratuitas e uma grande comunidade
 - 4.1. Eclipse, NetBeans, IntelliJ IDEA
 - 4.2. <https://www.tiobe.com/tiobe-index/>
5. Adotada por inúmeras universidades no mundo
6. Linguagem multiplataforma



Tiobe Index (Aug, 2021)

Fonte: <https://www.tiobe.com/tiobe-index/>



Conceitos

- O que é orientação a objetos?
 - É um paradigma de programação!
- E o que é um paradigma de programação?
 - Um padrão ou estilo de programação, suportado por linguagens com características comuns
 - Maneira de modelar o mundo real de forma computacional



Em uma linguagem mais popular...

- Orientação a objetos é considerar que tudo é um objeto:
 - Os sistemas são grandes aglomerados de objetos que se comunicam entre si
 - Os dados manipulados e comunicados pelos sistemas são objetos
 - Cada parte de um sistema (sub-sistema) também é um objeto



Objetos em um sistema

- Jogador, inimigo, equipe, armas, equipamentos



Tecnologia utilizada: Java

- Java Standard Edition (JSE)
 - contém os recursos necessários para desenvolver aplicativos de desktop e servidor
- Plataforma Android
 - voltado para desenvolvimento mobile Android
- Java Enterprise Edition (JEE)
 - é adequado para desenvolver aplicativos em rede distribuída e em grande escala e também aplicativos baseados na web



Android
Studio



maven


spring

Objetivos da disciplina

- Aprender como modelar o mundo real de maneira computacional usando o paradigma de orientação a objetos
- Fixação dos conceitos do paradigma OO
- Introdução de Linguagem OO
- Construção de aplicações OO



Nosso primeiro programa Java: Imprimindo uma linha de texto

```
1 // Figura 2.1: Welcome1.java
2 // Programa de impressão de texto.
3
4 public class Welcome1
5 {
6     // método main inicia a execução do aplicativo Java
7     public static void main(String[] args)
8     {
9         System.out.println("Welcome to Java Programming!");
10    } // fim do método main
11 } // fim da classe Welcome1
```

Welcome to Java Programming!

Comentando programas

1ª forma: começa com “//”

indicando que é um **comentário de fim de linha**, e termina no fim da linha, onde os caracteres // aparecem

```
// Figura 2.1: Welcome1.java
```

2ª forma: começam e terminam com delimitadores “/*” e “*/”

indicando que é um **comentários tradicionais**, O compilador ignora todo o texto entre os delimitadores

```
/* Esse é um comentário tradicional. Ele  
pode ser dividido em várias linhas */
```

Declarando uma classe

```
public class Welcome1
```

Declarando um método

```
public static void main(String[] args)
```

Gerando saída com System.out.println

```
System.out.println("Welcome to Java Programming!");
```



Modificando nosso primeiro programa Java

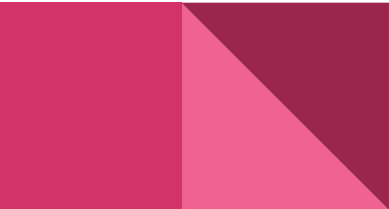
```
1 // Figura 2.3: Welcome2.java
2 // Imprimindo uma linha de texto com múltiplas instruções.
3
4 public class Welcome2
5 {
6     // método main inicia a execução do aplicativo Java
7     public static void main(String[] args)
8     {
9         System.out.print("Welcome to ");
10        System.out.println("Java Programming!");
11    } // fim do método main
12 } // fim da classe Welcome2
```

Welcome to Java Programming!

Welcome3.java

```
1 // Figura 2.4: Welcome3.java
2 // Imprimindo múltiplas linhas de texto com uma única instrução.
3
4 public class Welcome3
5 {
6     // método main inicia a execução do aplicativo Java
7     public static void main(String[] args)
8     {
9         System.out.println("Welcome\nto\nJava\nProgramming!");
10    } // fim do método main
11 } // fim da classe Welcome3
```

```
Welcome
to
Java
Programming!
```



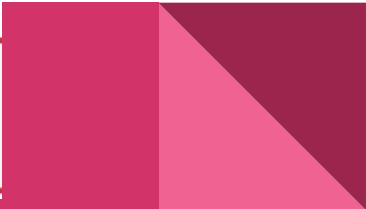
Outros exemplos de escapes

Sequência de escape	Descrição
<code>\n</code>	Nova linha. Posiciona o cursor de tela no início da <i>próxima</i> linha.
<code>\t</code>	Tabulação horizontal. Move o cursor de tela para a próxima parada de tabulação.
<code>\r</code>	Retorno de carro. Posiciona o cursor da tela no início da linha <i>atual</i> — <i>não</i> avança para a próxima linha. Qualquer saída de caracteres depois do retorno de carro <i>sobrescreve</i> a saída de caracteres anteriormente gerada na linha atual.
<code>\\</code>	Barras invertidas. Utilizadas para imprimir um caractere de barra invertida.
<code>\"</code>	Aspas duplas. Utilizadas para imprimir um caractere de aspas duplas. Por exemplo, <pre>System.out.println("\"entre aspas\");</pre> exibe "entre aspas".

Exibindo texto com printf

```
1  // Figura 2.6: Welcome4.java
2  // Exibindo múltiplas linhas com o método System.out.printf.
3
4  public class Welcome4
5  {
6      // método main inicia a execução do aplicativo Java
7      public static void main(String[] args)
8      {
9          System.out.printf("%s%n%s%n",
10             "Welcome to", "Java Programming!");
11     } // fim do método main
12 } // fim da classe Welcome4
```

Welcome to
Java Programming!



Entrada de dados com Scanner

```
1 // Figura 2.7: Addition.java
2 // Programa de adição que insere dois números, então exibe a soma deles.
3 import java.util.Scanner; // programa utiliza a classe Scanner
4
5 public class Addition
6 {
7     // método main inicia a execução do aplicativo Java
8     public static void main(String[] args)
9     {
10         // cria um Scanner para obter entrada a partir da janela de comando
11         Scanner input = new Scanner(System.in);
12
13         int number1; // primeiro número a somar
14         int number2; // segundo número a somar
15         int sum; // soma de number1 e number2
16     }
```

```
16
17     System.out.print("Enter first integer: "); // prompt
18     number1 = input.nextInt(); // lê primeiro o número fornecido pelo usuário
19
20     System.out.print("Enter second integer: "); // prompt
21     number2 = input.nextInt(); // lê o segundo número fornecido pelo usuário
22
23     sum = number1 + number2; // soma os números, depois armazena o total em sum
24
25     System.out.printf("Sum is %d\n", sum); // exibe a soma
26 } // fim do método main
27 } // fim da classe Addition
```

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

printf

Tabela - Tipos de dados básicos e representação

Linguagem C	Formato	Tipo de dados
<u>char</u>	%c	<u>caracter</u>
<u>int</u>	%d	<u>inteiro</u>
<u>float</u>	%f	<u>real</u>
<u>char[]</u>	%s	<u>cadeia de caracteres</u> (<u>string</u>)

Declarações de import

- Um dos pontos fortes do Java é seu rico conjunto de classes predefinidas que você pode reutilizar em vez de “reinventar a roda”. 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)**

```
import java.util.Scanner; // programa utiliza a classe Scanner
```



Documentação da Java API

<https://docs.oracle.com/javase/8/docs/api/>

The screenshot shows the Java Platform Standard Edition 8 API Specification website. The browser address bar displays <https://docs.oracle.com/javase/8/docs/api/>. The page has a dark blue header with navigation tabs: OVERVIEW, PACKAGE, CLASS, USE, TREE, DEPRECATED, INDEX, and HELP. The 'OVERVIEW' tab is selected. On the left, there is a sidebar with 'All Classes' and 'All Profiles' sections. The 'All Classes' section lists various classes like `AbstractAction`, `AbstractAnnotationValueVisitor6`, etc. The main content area is titled 'Java™ Platform, Standard Edition 8 API Specification'. It includes a description: 'This document is the API specification for the Java™ Platform, Standard Edition.' and a link to 'See: Description'. Below this, there is a 'Profiles' section with a bulleted list:

- compact1
- compact2
- compact3

. The 'Packages' section is highlighted with an orange tab. It contains a table with two columns: 'Package' and 'Description'. The table lists several packages including `java.applet`, `java.awt`, `java.awt.color`, `java.awt.datatransfer`, `java.awt.dnd`, and `java.awt.event`.

Package	Description
<code>java.applet</code>	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
<code>java.awt</code>	Contains all of the classes for creating user interfaces and for painting graphics and images.
<code>java.awt.color</code>	Provides classes for color spaces.
<code>java.awt.datatransfer</code>	Provides interfaces and classes for transferring data between and within applications.
<code>java.awt.dnd</code>	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
<code>java.awt.event</code>	Provides interfaces and classes for dealing with different types of events fired by AWT components.
<code>java.awt.font</code>	Provides classes and interface relating to fonts.

Conceitos de memória

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` e `number2`.

```
number1 = input.nextInt(); // lê primeiro o número fornecido pelo usuário
```

```
number2 = input.nextInt(); // lê o segundo número fornecido pelo usuário
```

`number1`

45

`number2`

72

Conceitos de memória

Depois de o programa obter os valores para number1 e number2, ele os adiciona e coloca o total na variável sum.

```
sum = number1 + number2; // adiciona números, depois armazena total na soma
```

number1

45

number2

72

sum

117

Aritmética

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

Operação Java	Operador
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Resto	%

Tomadas de decisão

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

Conclusão

Neste capítulo, você aprendeu muitos recursos importantes do Java, incluindo como exibir dados na tela em um Command Prompt, inserir dados a partir do teclado, realizar cálculos e tomar decisões. Os aplicativos apresentados aqui introduzem muitos conceitos básicos de programação.

