

A series of overlapping geometric shapes, primarily triangles and quadrilaterals, in shades of teal and purple, located in the top-left corner of the slide.

AceLeraDev Java

Módulo 1 - Introdução a linguagem Java

A series of overlapping geometric shapes, primarily triangles and quadrilaterals, in shades of teal and purple, located in the bottom-right corner of the slide.



- Profissional com mais de 6 anos de experiência, tendo convivido com diversas linguagens (Java, .NET, Ruby). Atuou no desenvolvimento de projetos com Cognitive Services e Cloud dentre esses projetos, um premiado em Cannes. Hoje é uma entusiasta em jogos, colabora com a comunidade auxiliando, apoiando novas meninas na área

- Devs JavaGirl, MTAC.
- Coordenadora da Trilha de Modern Web durante 3 anos e atualmente na trilha Java.

Ementa geral do programa

Módulo 1: Introdução a linguagem de programação

Módulo 2: Programação Orientada a objetos I

Módulo 3: Programação Orientada a objetos II

Módulo 4: Modelagem de banco de dados relacional

Módulo 5: Manipulação de dados em banco de dados relacional

Módulo 6: Criação de APIs REST

Módulo 7: Documentação e autenticação de APIs

Módulo 8: Git, boas práticas e clean code

Módulo 9: Deploy de aplicações

Módulo 10: Preparação para o demo day



Tópicos desta aula:

- Básico da Linguagem Java


Tópico 1: Pacotes e classes

Tópico 2: Variáveis e métodos

Tópico 3: Métodos, Construtores e sobrecarga de método

Tópico 4: Operações aritméticas


Tópico 6: Estruturas de repetição





Classes

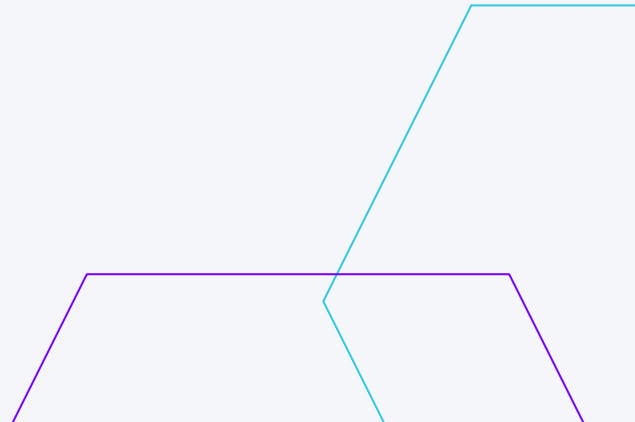
Além da especificação de atributos, a definição de uma classe descreve também qual o comportamento de objetos da classe, ou seja, que funcionalidades podem ser aplicadas a objetos da classe. Essas funcionalidades são descritas através de métodos.





Pacotes

É muito comum para novo desenvolvedores que vem da linguagem estruturada (C++ por exemplo) que todas as classes sejam criadas em um único pacote, o que não é ideal em linguagens como Java.



Como poderíamos organizar esses packages?

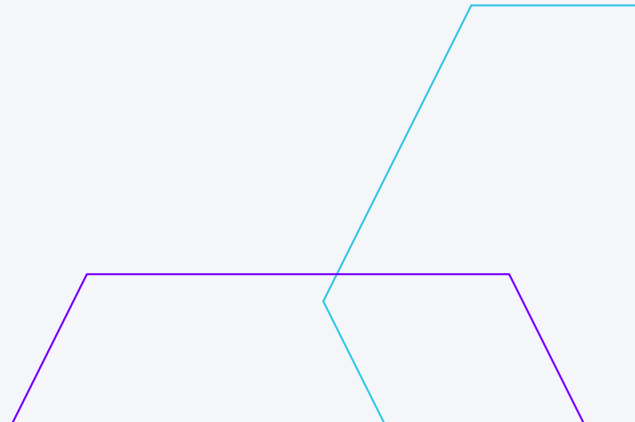
Utils	- Enums, format String, Date.
Mapper	- @Mapper (MapperStruct)
Repository	- @Repository / @MongoRepository
Entity	- @Entity / @Document
Controllers	- @Controllers / @RestController
Service	- @Component / @Services
ServiceImpl	- Implementação de interfaces caso utilize

Mas pode pensar na hierarquia que atenda seu projeto.



SHOW THE CODE

Vamos gerar um projeto java.






Variáveis

Variáveis globais - são aquelas declaradas fora do escopo das funções.

Variáveis locais - são aquelas declaradas no início do bloco e seus escopos são restritos aos blocos em que foram declaradas.



*Constantes

```
private static final int CONSTANTE= 7;
```

Onde usar?

Blocos de inicialização, variáveis/métodos e classes internas.

Exercícios

1. Escreva um algoritmo que armazene o valor 10 em uma variável A e o valor 20 em uma variável B. A seguir (utilizando apenas atribuições entre variáveis) troque os seus conteúdos fazendo com que o valor que está em A passe para B e vice-versa. Ao final, escrever os valores que ficaram armazenados nas variáveis.

2. Analise os algoritmos ao lado e diga o que será impresso na tela ao serem executados:

a)

$A \leftarrow 10$
$B \leftarrow 20$
Escrever B
$B \leftarrow 5$
Escrever A, B

b)

$A \leftarrow 30$
$B \leftarrow 20$
$C \leftarrow A + B$
Escrever C
$B \leftarrow 10$
Escrever B, C
$C \leftarrow A + B$
Escrever A, B, C

c)

$A \leftarrow 10$
$B \leftarrow 20$
$C \leftarrow A$
$B \leftarrow C$
$A \leftarrow B$
Escrever A, B, C

d)

$A \leftarrow 10$
$B \leftarrow A + 1$
$A \leftarrow B + 1$
$B \leftarrow A + 1$
Escrever A
$A \leftarrow B + 1$
Escrever A, B

e)

$A \leftarrow 10$
$B \leftarrow 5$
$C \leftarrow A + B$
$B \leftarrow 20$
$A \leftarrow 10$
Escrever A, B, C

f)

$X \leftarrow 1$
$Y \leftarrow 2$
$Z \leftarrow Y - X$
Escrever Z
$X \leftarrow 5$
$Y \leftarrow X + Z$
Escrever X, Y, Z

Variáveis de tipos primitivos

```
byte = 127;
```

```
short = 1000;
```

```
Int = 100000000;
```

```
long = 1000000000000000L;
```

```
float = 10.10F // ponto flutuante
```

```
double = 100000000000.299 //ponto flutuante
```

```
char = 'S';
```

```
boolean = true;
```

STRING

O que já vimos :

```
String nome = "Camila";  
System.out.println("Hello World, " + nome);
```

Operadores Aritméticos

ADIÇÃO	$a+1$
SUBTRAÇÃO	$b-1$
MULTIPLICAÇÃO	$c*m$
DIVISÃO	d/e
RESTO	$f\%g$

$*$ / $\%$	Avaliado primeiro
$+-$	Avaliado em seguida
$=$	Último avaliado

*Wrapper Class

<https://www.devmedia.com.br/wrappers-em-java-aprenda-como-utilizar/30275>

char	Char
byte	Byte
short	Short
long	Integer
float	Float
double	Double
boolean	Boolean

Modificadores de acesso

	private	default	protected	public
mesma classe	sim	sim	sim	sim
mesmo pacote	não	sim	sim	sim
pacotes diferentes (subclasses)	não	não	sim	sim
pacotes diferentes (sem subclasses)	não	não	não	sim

Métodos, Construtores e sobrecarga de método

modificador de acesso + retorno + nome do método

```
public void metodo() { //métodos e variáveis se iniciam com letra minúscula  
}
```

Operadores relacionais

>	Utilizado quando desejamos verificar se uma variável é maior que outra.
>=	Utilizado quando desejamos verificar se uma variável é maior ou igual a outra.
<	Utilizado quando desejamos verificar se uma variável é menor que outra.
<=	Utilizado quando desejamos verificar se uma variável é menor ou igual a outra.

Operadores de igualdade

Operador de igualdade	Operador de igualdade	Exemplo de condição em Java	Significado da condição em Java
Operadores de igualdade			
=	==	x == y	x é igual a y
?	!=	x != y	x é diferente de y
Operadores relacionais			
>	>	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

Operadores lógicos

Operadores “OU” e “E”

||

&&

Condicional

```
public void metodo() {  
    boolean condicao = true;  
    if (condicao) {  
  
    } else if (!condicao) {  
  
    } else {  
  
    }  
}
```

Estruturas de repetição

- foreach
- do while
- while
- switch
- break



Revisão do que vimos hoje:

Tópico 1: Pacotes e classes

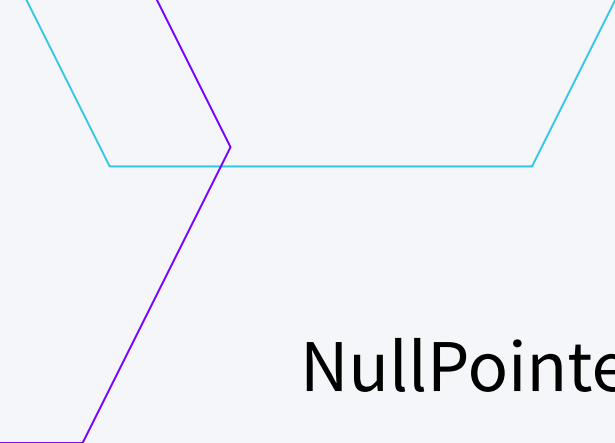
Tópico 2: Variáveis e métodos

Tópico 3: Métodos, Construtores e sobrecarga de método

Tópico 4: Operações aritméticas

Tópico 6: Estruturas de repetição





NullPointerException

Exception in thread "main" java.lang.NullPointerException

at Tarefas.nullpointer(Tarefas.java:30)

at Tarefas.metodo(Tarefas.java:25)

at hello.main(hello.java:11)



Arrays vs ArrayList

```
public void metodo() {  
    String[] array = new String[10];  
    ArrayList<String> arrayList = new ArrayList<>();  
  
    // List<String> lista = new ArrayList<>();  
  
    array[0] = this.nome;  
    arrayList.add(this.nome);  
  
}
```

Exercícios

3) Os pares de instruções abaixo produzem o mesmo resultado?

$$A \leftarrow (4/2)+(2/4) \quad \text{e} \quad A \leftarrow 4/2+2/4$$

$$B \leftarrow 4/(2+2)/4 \quad \text{e} \quad B \leftarrow 4/2+2/4$$

$$C \leftarrow (4+2)*2-4 \quad \text{e} \quad C \leftarrow 4+2*2-4$$

4) Reescreva as instruções abaixo com o **mínimo de parênteses possível**, mas **sem alterar o resultado**:

$A \leftarrow 6*(3+2)$	$F \leftarrow (6/3)+(8/2)$
$B \leftarrow 2+(6*(3+2))$	$G \leftarrow (((3+(8/2))*4)+(3*2))$
$C \leftarrow 2+(3*6)/(2+4)$	$H \leftarrow (6*(3*3)+6)-10$
$D \leftarrow 2*(8/(3+1))$	$I \leftarrow (((10*8)+3)*9)$
$E \leftarrow 3+(16-2)/(2*(9-2))$	$J \leftarrow ((-12)*(-4))+(3*(-4))$

5) Escreva um algoritmo para **ler** um valor (do teclado) e **escrever** (na tela) o seu **antecessor**.

6) Escreva um algoritmo para ler as dimensões de um retângulo (base e altura), calcular e escrever a área do retângulo.

7) Faça um algoritmo que leia a idade de uma pessoa expressa em anos, meses e dias e escreva a idade dessa pessoa expressa apenas em dias. Considerar ano com 365 dias e mês com 30 dias.

8) Escreva um algoritmo para ler o número total de eleitores de um município, o número de votos brancos, nulos e válidos. Calcular e escrever o percentual que cada um representa em relação ao total de eleitores.

9) Escreva um algoritmo para ler o salário mensal atual de um funcionário e o percentual de reajuste. Calcular e escrever o valor do novo salário.

10) O custo de um carro novo ao consumidor é a soma do custo de fábrica com a porcentagem do distribuidor e dos impostos (aplicados ao custo de fábrica). Supondo que o percentual do distribuidor seja de 28% e os impostos de 45%, escrever um algoritmo para ler o custo de fábrica de um carro, calcular e escrever o custo final ao consumidor.

11) Uma revendedora de carros usados paga a seus funcionários vendedores um salário fixo por mês, mais uma comissão também fixa para cada carro vendido e mais 5% do valor das vendas por ele efetuadas. Escrever um algoritmo que leia o número de carros por ele vendidos, o valor total de suas vendas, o salário fixo e o valor que ele recebe por carro vendido. Calcule e escreva o salário final do vendedor.



Alguma dúvida ou questionamento?

Abram o coração S2



Alguns links...

feedback semanal

<https://docs.google.com/forms/d/1PR7Xh9uUAXWnRBiVyalwtOX-LpMhSMM-p58zSEp3m30/prefill>

Orientações Projeto final

https://docs.google.com/document/d/1yk4_gxgSpEXK_y6J62HvOQjMAdwaEZO7F05Tii44lwq/edit