




CURSO DE PROGRAMAÇÃO EM JAVA

Aula 9 
Módulos de
programa em Java



*A melhor maneira de desenvolver e manter um programa grande é construí-lo a partir de pequenas e simples partes, ou **módulos**. Essa técnica se chama **dividir para conquistar**.*

1.

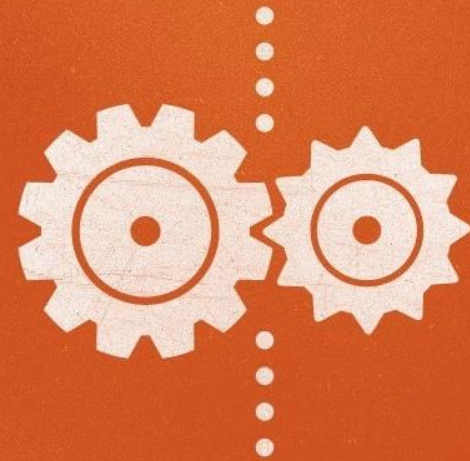
Módulos de programa em JAVA

Há 3 tipos de módulos em Java – métodos, classes e pacotes.

Módulos de programa em JAVA

Os programas Java são escritos combinando novos **métodos** e **classes** que você escreve com métodos e classes predefinidas disponíveis no **Java Application Programming Interface** (Java API ou biblioteca de classes Java) e em várias outras bibliotecas de classes. Em geral, as classes relacionadas são agrupadas em **pacotes** de modo que possam ser importadas nos programas e reutilizadas.

PROGRAMMER

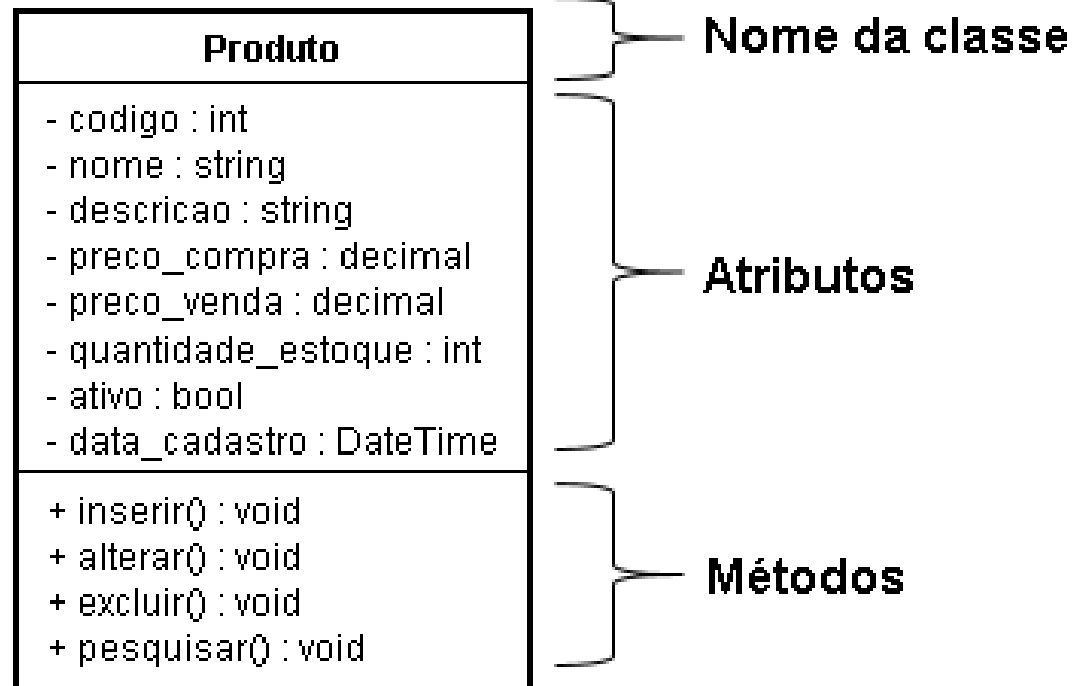


</CODE>

Classes

Uma classe é uma estrutura que abstrai um conjunto de objetos com características similares. Uma classe define o **comportamento** de seus objetos - através de **métodos** - e os **estados** possíveis destes objetos - através de **atributos**.

Em outras palavras, uma classe se descreve nos serviços oferecidos por seus objetos onde os quais informam onde eles podem ser armazenados.

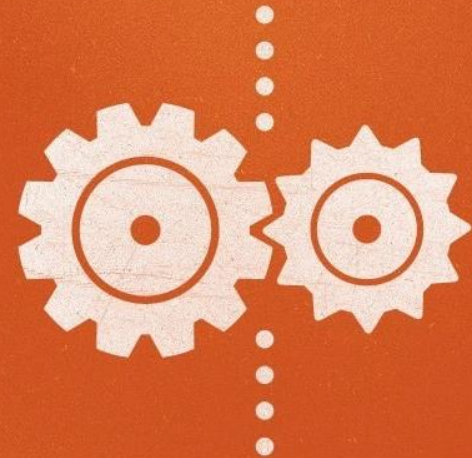


Métodos

Também chamados de **funções** ou **procedimentos**. Eles ajudam a modularizar um programa separando suas tarefas em unidades específicas.

As instruções no corpo dos métodos são escritas apenas uma vez , permanecem ocultas de outros métodos e podem ser reutilizadas a partir de várias localizações em um programa.

PROGRAMMER

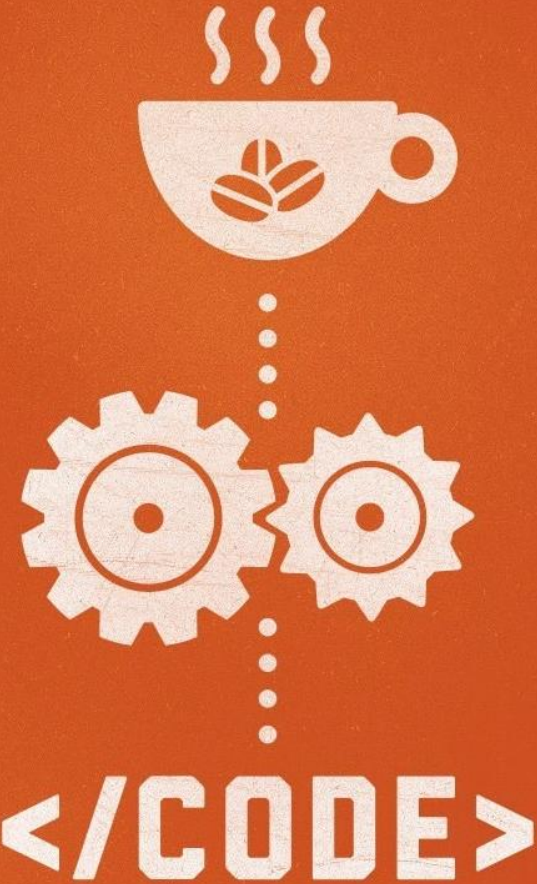


</CODE>

Dividir para conquistar

- Torna o desenvolvimento do programa mais gerenciável
- Constroi programas a partir de peças mais simples e menores
- Capacidade de reutilização de código
- Mais fácil de testar e deputar o código

PROGRAMMER



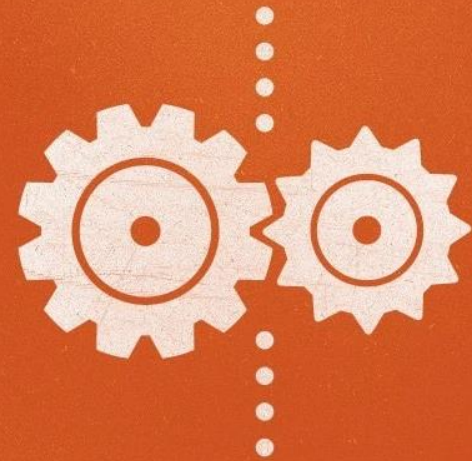
O que são pacotes em Java?

Um pacote (package) em Java é um diretório em que está armazenada uma ou mais classes. Os pacotes costumam agrupar classes de mesmas afinidades.

A palavra reservada `package` define em qual pacote se encontra a classe.

```
package javainiciante;  
  
public class Aluno {  
    //Estrutura da classe  
}
```

PROGRAMMER



</CODE>

O que são pacotes em Java?

Para utilizar classes de um pacote, é usada a palavra reservada `import` do seguinte modo:

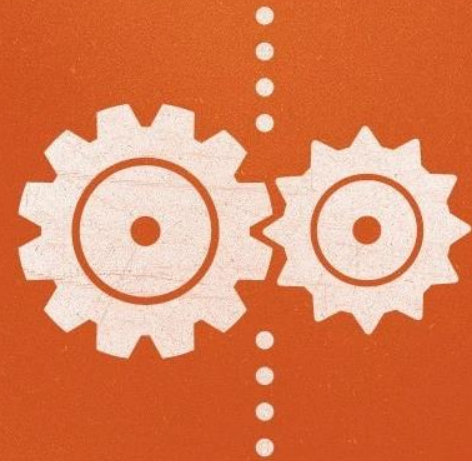
`import nomePacote.nomeClasse`

Ex1.: classe Teste importando a classe Aluno:

```
import javainiciante.Aluno;

public class Teste {
    Aluno aluno = new Aluno();
    //Estrutura da classe
}
```

PROGRAMMER



</CODE>

2.

Métodos static

Métodos static

Além dos métodos que são executados a partir de chamadas feitas por objetos específicos:

(**objeto.metodo(args)**), existem métodos que realiza uma tarefa que **não depende** do conteúdo de nenhum **objeto**, logo não precisa se chamado a partir de um objeto:

(**classe.metodo(args)**), e eles são conhecidos como **métodos static**.



Métodos static

A **classe Math** possui vários exemplos de **métodos static**, como por exemplo o **pow**, usada para elevar um valor a uma potência, para utilizá-lo **não precisamos declarar nenhum objeto**, basta apenas especificar o nome da classe seguindo de um ponto e o nome do método:

classe.método(argumentos);

Exemplo:

Math.sqrt(900.0) //retorna 30.0



Métodos static

```
//Sintaxe:
Public nomeDaClasse{

    public static void nomeDoMetodo (Parametros) {
        //Instruções
    }

    public static void main(String[] args){
        //Não precisamos criar um objeto para
        utilizar do método
        nomeDaClasse.nomeDoMetodo (Parametros);
    }
}
```


Métodos da classe Math

Método	Descrição	Exemplo
<code>abs(x)</code>	valor absoluto de x	<code>abs(23.7)</code> é 23.7 <code>abs(0.0)</code> é 0.0 <code>abs(-23.7)</code> é 23.7
<code>ceil(x)</code>	arredonda x para o menor inteiro não menor que x	<code>ceil(9.2)</code> é 10.0 <code>ceil(-9.8)</code> é -9.0
<code>cos(x)</code>	co-seno trigonométrico de x (x em radianos)	<code>cos(0.0)</code> é 1.0
<code>exp(x)</code>	método exponencial e^x	<code>exp(1.0)</code> é 2.71828 <code>exp(2.0)</code> é 7.38906
<code>floor(x)</code>	arredonda x para o maior inteiro não maior que x	<code>floor(9.2)</code> é 9.0 <code>floor(-9.8)</code> é -10.0
<code>log(x)</code>	logaritmo natural de x (base e)	<code>log(Math.E)</code> é 1.0 <code>log(Math.E * Math.E)</code> é 2.0
<code>max(x,y)</code>	maior valor de x e y	<code>max(2.3, 12.7)</code> é 12.7 <code>max(-2.3, -12.7)</code> é -2.3
<code>min(x,y)</code>	menor valor de x e y	<code>min(2.3, 12.7)</code> é 2.3 <code>min(-2.3, -12.7)</code> é -12.7
<code>pow(x,y)</code>	x elevado à potência de y (isto é, x^y)	<code>pow(2.0, 7.0)</code> é 128.0 <code>pow(9.0, 0.5)</code> é 3.0
<code>sin(x)</code>	seno trigonométrico de x (x em radianos)	<code>sin(0.0)</code> é 0.0
<code>sqrt(x)</code>	raiz quadrada de x	<code>sqrt(900.0)</code> é 30.0
<code>tan(x)</code>	tangente trigonométrica de x (x em radianos)	<code>tan(0.0)</code> é 0.0

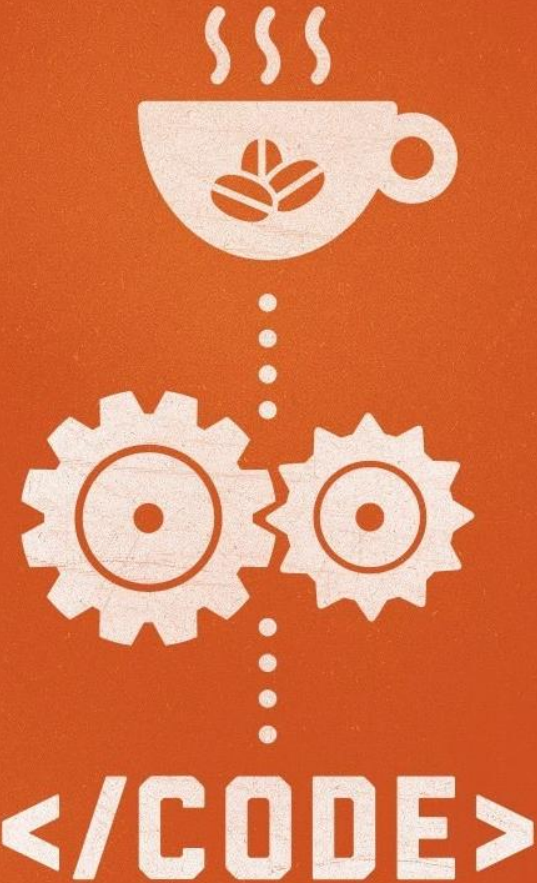
3.

Pacotes de Java API

Pacotes de Java API

As classes e interfaces que compõem a API do Java estão divididas em pacotes, e cada pacote agrupa um conjunto de classes e interfaces que possuem propósitos comuns. Lembre-se de que um pacote também pode conter outros pacotes, de modo que eles acabam formando uma estrutura complexa de compartimentos.

PROGRAMMER



Pacotes de Java API

Pacote	Descrição
applet	Provê as classes necessárias para criar um applet e as classes que um applet usa para se comunicar com seu contexto.
awt	Contém classes e interfaces utilizadas para desenhar gráficos e imagens e construir GUIs.
beans	Contém classes relacionadas ao desenvolvimento de componentes beans baseados na arquitetura denominada como JavaBean.
io	Provê entrada e saída para o sistema através de fluxo, serialização e arquivos de sistema.
lang	Provê classes que são fundamentais ao desígnio de Java enquanto linguagem de programação.
math	Provê classes para executar aritmética de inteiros de precisão arbitrária e aritmética decimal de precisão.
net	Provê classes para implementação de aplicações de redes.
nio	Define “buffers”, que são recipientes para dados e proveem uma prévia dos outros pacotes NIO.

Pacotes de Java API

rmi	Provê classes e interfaces para implementar aplicativos que utilizem invocação remota de métodos (RMI – Remot Method Invocation.)
Security	Provê classes e interfaces para implementar procedimentos de segurança de informações.
sql	Provê classes e interfaces para acessar e processar dados armazenados em uma fonte de dados, normalmente um banco de dados relacional.
text	Provê classes e interfaces para controlar texto, datas, números e mensagens de modo independente de idiomas naturais.
util	Contém a estrutura de coleções, modelo de eventos, facilidades com data e hora, internacionalização e classes de utilidades diversas.

Alguns métodos da classe `java.util.Random`.

Assinatura	Descrição
<code>boolean nextBoolean()</code>	Retorna um valor booleano aleatório.
<code>double nextDouble()</code>	Retorna um valor aleatório entre 0,0 e 0,9 como um tipo <code>double</code> .
<code>float nextFloat()</code>	Retorna um valor aleatório entre 0,0 e 0,9 como um tipo <code>float</code> .
<code>int nextInt()</code>	Retorna um número inteiro aleatório.
<code>int nextInt(int n)</code>	Retorna um número inteiro aleatório entre 0 e o valor especificado através do parâmetro <code>n</code> .
<code>long nextLong()</code>	Retorna um número inteiro longo aleatório.



DESAFIO

E aí, vamos praticar?

Ultrapassando Z

Faça um programa que leia dois inteiros: X e Z (devem ser lidos tantos valores para Z quantos necessários, até que seja digitado um valor maior do que X para ele). Conte quantos números inteiros devem ser somados em sequência (considerando o X nesta soma) para que a soma ultrapasse a Z o mínimo possível. Escreva o valor final da contagem.

A entrada pode conter, por exemplo, os valores 21 21 15 30. Neste caso, é então assumido o valor 21 para X enquanto os valores 21 e 15 devem ser desconsiderados pois são menores ou iguais a X. Como o valor 30 está dentro da especificação (maior do que X) ele será válido e então deve-se processar os cálculos para apresentar na saída o valor 2, pois é a quantidade de valores somados para se produzir um valor maior do que 30 ($21 + 22$).

Ultrapassando Z

Entrada: Contém somente valores inteiros, um por linha, podendo ser positivos ou negativos. O primeiro valor da entrada será o valor de X. A próxima linha da entrada irá conter Z. Se Z não atender a especificação do problema, ele deverá ser lido novamente, tantas vezes quantas forem necessárias.

Saída: Imprima uma linha com um número inteiro que representa a quantidade de números inteiros que devem ser somadas, de acordo com a especificação acima.

Entrada:

3
1
20

Saída:

5

Soma de Pares Consecutivos

O programa deve ler um valor inteiro X indefinidas vezes. (O programa irá parar quando o valor de X for igual a 0). Para cada X lido, imprima a soma dos 5 pares consecutivos a partir de X, inclusive o X, se for par. Se o valor de entrada for 4, por exemplo, a saída deve ser 40, que é o resultado da operação: $4+6+8+10+12$, enquanto que se o valor de entrada for 11, por exemplo, a saída deve ser 80, que é a soma de $12+14+16+18+20$.

Entrada: O arquivo de entrada contém muitos valores inteiros. O último valor do arquivo é zero.

Saída: Imprima a saída conforme a explicação acima e o exemplo abaixo

Entrada:

4
11
0

Saída:

40
80

Obrigado!

Alguma pergunta?

Você pode nos contatar em:
ywassef@hotmail.com