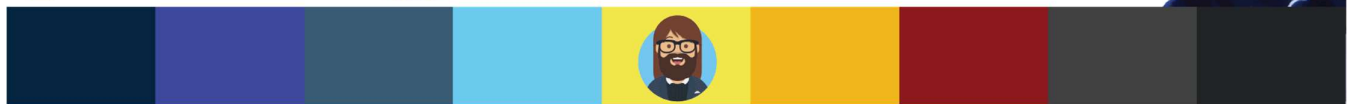


## Introdução a linguagem de programação Java – Imports

Marcos Vinícius da Silva Santos e Marcos Antonio dos Santos

```
/**
 *
 * CREATIVE COMMONS (CC) 2020-2021 MARCOS VINÍCIUS DA SILVA SANTOS AND MARCOS ANTONIO DOS SANTOS
 *
 * LICENSED UNDER THE CREATIVE COMMONS, VERSION 4.0; YOU MAY NOT USE THIS FILE EXCEPT IN COMPLIANCE WITH THE LICENSE.
 * YOU MAY OBTAIN A COPY OF THE LICENSE AT
 * HTTPS://CREATIVECOMMONS.ORG/LICENSES/BY-NC-SA/4.0/
 * HTTPS://CREATIVECOMMONS.ORG/LICENSES/BY-NC-SA/4.0/LEGALCODE
 * ATTRIBUTION-NONCOMMERCIAL-SHAREALIKE 4.0 INTERNATIONAL (CC BY-NC-SA 4.0)
 *
 * LICENCIADO PELA CREATIVE COMMONS, VERSÃO 4.0; VOCÊ NÃO PODE USAR ESTE ARQUIVO, EXCETO EM CONFORMIDADE COM A LICENÇA.
 * VOCÊ PODE OBTER UMA CÓPIA DA LICENÇA EM
 * HTTPS://CREATIVECOMMONS.ORG/LICENSES/BY-NC-SA/4.0/DEED.PT\_BR
 * HTTPS://CREATIVECOMMONS.ORG/LICENSES/BY-NC-SA/4.0/LEGALCODE.PT
 * ATRIBUIÇÃO-NÃOCOMERCIAL-COMPARTILHAIGUAL 4.0 INTERNACIONAL (CC BY-NC-SA 4.0)
 *
 * UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING, SOFTWARE DISTRIBUTED UNDER THE LICENSE IS DISTRIBUTED ON AN "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED.
 * SEE THE LICENSE FOR THE SPECIFIC LANGUAGE GOVERNING PERMISSIONS AND LIMITATIONS UNDER THE LICENSE.
 */
```

# profdigital.com.br



### ISENÇÃO DE RESPONSABILIDADE E OUTRAS INFORMAÇÕES LEGAIS

*Todas as opiniões quando expressas neste documento ou no site em que é apresentado este documento não representam as opiniões de nenhuma entidade com a qual os autores tenham sido associados, são agora associados ou serão associados futuramente.*

*Qualquer ação envolvendo programação que você execute com as informações deste documento é feita estritamente sob a sua responsabilidade para o desenvolvimento de habilidades técnicas e profissionais, portanto, os autores não se responsabilizam por quaisquer erros ou omissões, ou pelos resultados obtidos com o uso das informações contidas nesse documento.*

*Todas as informações neste documento são fornecidas "no estado em que se encontram", sem a garantia dos resultados obtidos com o uso destas informações. Parte das informações aqui apresentadas foram obtidas em diversas fontes e, mesmo com todo o cuidado em sua coleta e manuseio, os autores não se responsabilizam pela publicação acidental de dados incorretos que possam levar a qualquer tipo de dano ou prejuízo.*

*Os autores recomendam que em caso de dúvidas o utilizador deste documento busque saná-las recorrendo à documentação oficial do fabricante que é disponibilizada nos respectivos sites.*

# Java Imports

A linguagem de programação Java tem uma instrução que permite importar um conjunto de recursos para facilitar o desenvolvimento de uma aplicação.

Utilizamos alguns destes conjuntos de recursos para o desenvolvimento do Globo de Bingo recentemente:

- `import java.util.Scanner;`
- `import java.util.Random;`
- `import java.util.List;`
- `import java.util.ArrayList;`

As importações em Java fazem com que pacotes de códigos (classes, funcionalidades e métodos) sejam integradas ao projeto que estamos desenvolvendo e nos poupa um enorme tempo de desenvolvimento. Esses pacotes foram codificados ao longo dos anos pela comunidade de desenvolvedores Java.

Se você precisa desenvolver um projeto que necessite trabalhar com datas será necessário importar pacotes de códigos para atender a essa necessidade:

- `import java.util.Calendar;`
- `import java.util.Date;`

Se você tiver um olho aguçado, verá as importações estão organizadas logo no início dos nossos códigos.

Com esses imports podemos integrar em nossos aplicativos coisas como:

- Ler o conteúdo de um arquivo / Criar um arquivo e preencher com conteúdo
- Comparar datas entre si (ou seja, veja se uma data é anterior ou posterior à outra)
- Enviar e-mails para qualquer pessoa

## Olhar atento

Observando os exemplos de importações anteriores veremos que todos tem em comum o pacote “java.util”, porém, cada uma atende a uma determinada Classe. “Scanner”, “Random”, “ArrayList”, “List”, “Calendar” e “Date”. Existem diversas classes prontas e neste endereço da documentação Java poderá encontrar mais <https://docs.oracle.com/javase/8/docs/api/java/util/package-summary.html>.

A classe “Scanner” é usada para escutar a nossa entrada de dados que é feita em nossos exemplos por meio do teclado.

A classe “Random” é usada para gerar um fluxo de números pseudo-aleatórios.

A classe “ArrayList” é usada para dimensionar ou redimensionar uma lista de valores.

A classe “List” é usada para representar uma lista (coleção) de valores ordenados (sequências).

A classe “Calendar” é usada para obter um instante específico no tempo e um conjunto de dados do calendário, como YEAR, MONTH, DAY\_OF\_MONTH, HOUR e assim por diante, e também para manipular os campos do calendário, como obter a data da próxima semana.

A classe “Date” representa um instante específico no tempo, com precisão de milissegundos.

A motivação pela qual precisamos dizer ao Java que usaremos essas classes ou outras é uma questão de brevidade e especificidade na criação do nosso código, onde buscamos aumentar a nossa produtividade e assim, manter o foco naquilo que precisa ser feito para resolver o problema de alguém.

## Por que eu preciso do Java Imports?

Bem, sabe como é... a verdade é que você realmente não PRECISA usar importações. Você pode referir-se ao nome completo do pacote da classe que deseja usar. Veja abaixo o código:

```
public class MyProgram
{
    public static void main(String[] args)
    {
        java.util.Calendar calendar = java.util.Calendar.getInstance();
        calendar.set(1968, 12, 13);
        java.util.Date firstDate = calendar.getTime();

        calendar.set(1968, 12, 10);
        java.util.Date secondDate = calendar.getTime();

        System.out.println("Is firstDate before secondDate? " + firstDate.before(secondDate));
        System.out.println("Is firstDate after secondDate? " + firstDate.after(secondDate));
    }
}
```

Resultado do processamento:

```
Is firstDate before secondDate? false
Is firstDate after secondDate? true
```

Lembre-se que: before → antes de

Lembre-se que: after → depois

Observe que no código anterior as **instruções de importação no início do código sumiram** e agora prefixamos para cada Date e Calendar a referência com **java.util**. Este é um código correto e irá compilar (rodar/executar/performar) perfeitamente. Mas é um pouquinho chato ter que escrever java.util cada vez que você faz uma referência a Date ou a Calendar. Portanto, é por isso que temos instruções de importação em Java no início, isso apenas torna a vida mais fácil do programador!

Veja a seguir o código com as importações declaradas no início:

```
import java.util.Calendar;
```

```
import java.util.Date;
```

```
public class MyProgram
{
    public static void main(String[] args)
    {
        Calendar calendar = Calendar.getInstance();
        calendar.set(1968, 12, 13);
        Date firstDate = calendar.getTime();

        calendar.set(1968, 12, 10);
        Date secondDate = calendar.getTime();

        System.out.println("Is firstDate before secondDate? " + firstDate.before(secondDate));
        System.out.println("Is firstDate after secondDate? " + firstDate.after(secondDate));
    }
}
```

Resultado do processamento:

```
Is firstDate before secondDate? false
Is firstDate after secondDate? true
```

Reforçando que em Java, a instrução **import** é escrita diretamente após a instrução do pacote (se existir) e antes da definição da classe. Por exemplo,

```
package package.name;

import package.ClassName; // importe apenas uma classe

public class MyClass {
```

```
// corpo do código  
}
```

## Importando todas as Classes em um Pacote

Existe uma outra forma que funciona como um “atalho” para importar classes em Java, ao usar o caractere curinga (\*). Digamos, por exemplo, que você deseja apenas importar SOMENTE TODAS as classes que pertencem ao pacote **java.util**, você pode apenas usar o código **import java.util.\***. Isso importará tudo em java.util. No entanto, este “método de importação” não é o mais recomendado e usado.