

Grupo 3

Athos Franco, Alexssandra Pimentel,
Eduardo Silveira, Viviane Magalhães, Guilherme Simões



Power Classes

`java.util.Calendar`

`java.time.LocalDate`

`java.time.LocalDateTime`



Calendar

java.util.Calendar

//Descrição

Essa classe pode produzir os valores de todos os campos de calendário necessários para implementar a formatação de data e hora, para uma determinada língua e estilo de calendário. Por exemplo, japonês, americano, italiano, brasileiro entre outros.

Hoje a maioria dos métodos da **classe Date** estão classificados como deprecated (descontinuado), ou seja, são métodos que não são mais utilizados, por isso essa classe foi substituída pela classe Calendar, para haver suporte correto à internacionalização do sistema de datas.

java.util.Calendar

//Instanciação

A classe Calendar é uma **classe abstrata**, portanto, não pode ser instanciada. Para obter uma instância da classe Calendar é necessário usar o método estático **getInstance()**.

```
1  import java.util.Calendar;
2
3  public class Data_Calendar{
4
5      public static void main(String[] args) {
6          Calendar c = Calendar.getInstance();
7          System.out.println("Data e Hora atual: "+c.getTime());
8      }
9  }
```

java.util.Calendar

//Métodos mais utilizados

getInstance()

Utilizado para instanciar o Calendar, retorna uma instância do calendário, sendo do tipo Calendar. É um método sobrecarregado.

getTime()

Retorna um objeto Date com os dados de hoje.

getAvaliableLocales(Locale locale)

Retorna uma array com a lista de locales disponíveis para uso.

get(int field)

Retorna um integer com a informação referente ao campo informado. É um método sobrecarregado.

Métodos sobrecarregados são escritos com o mesmo nome, mas com uma lista de argumentos diferente. São geralmente usados dentro de uma mesma classe.

```
getInstance()
```

Gets a calendar using the default time zone and locale.

```
getInstance(Locale aLocale)
```

Gets a calendar using the default time zone and specified locale.

```
getInstance(TimeZone zone)
```

Gets a calendar using the specified time zone and default locale.

```
getInstance(TimeZone zone, Locale aLocale)
```

Gets a calendar with the specified time zone and locale.

java.util.Calendar

//Demonstração de Uso

Usar a classe Calendar para imprimir uma mensagem com as informações da data de hoje.

Exemplo: "Boa tarde! Hoje é Terça, 22/02/2022. São 16:30 horas."

LocalDate

java.time.LocalDate

//Descrição

A classe LocalDate pode conter apenas a data.

É um objeto de data e hora imutável que representa uma data, geralmente vista como ano-mês-dia. Outros campos de data, como dia do ano, dia da semana e semana do ano, também podem ser acessados. Por exemplo, o valor "2 de outubro de 2007" pode ser armazenado em um arquivo LocalDate.

java.time.LocalDate

//Métodos mais utilizados

now ()

Retorna a data atual do relógio do sistema.

of ()

Passando os valores de Ano, Mês e Dia para o método

parse ()

Ele pode ser usado para converter uma data do tipo string em uma instância localdate. A string deve representar uma data válida no formato ISO LocalDate. Por exemplo, 2015-02-10 é válido, enquanto 20-10-2015 é uma string de data inválida.

ofInstant ()

Instant , ZoneId, é usado para criar uma instância de LocalDate a partir de um Instant e ID de zona. Esses dois parâmetros são passados para o método e com base neles, uma instância de LocalDate é criada.

java.time.LocalDate

Sobrecarga de método

Temos o método of(LocalDate, LocalTime) sobrecarregado recebendo como parâmetro objetos LocalDate e LocalTime.

```
LocalDateTime dateTime = LocalDateTime.of( 2014 , 12 , 15 , 10 ,45 , 0 );  
System.out.println("Date e Hora com parâmetro: "+dateTime);
```

```
LocalDate data = LocalDate.of(2014,11,12);  
LocalTime time = LocalTime.of( 14 , 15 , 40);  
LocalDateTime dateTime2 = LocalDateTime.of( data , time);
```

```
System.out.println("Date e Hora: "+dateTime2);
```


java.time.LocalDate

//Demonstração de Uso

Usar a classe LocalDate e diferentes métodos para imprimir a data de hoje, ontem e amanhã.

Exemplo:

Date de Hoje: 22/02/2022

Data de Ontem: 21/02/2022

Data de Amanhã: 23/02/2022

LocalDateTime

java.time.LocalDateTime

//Descrição

A classe LocalDateTime contém a data e a hora. Por exemplo, diga seja 2021-02-28T19:32:25.457826.

Nesse formato, "T" é utilizado para separar a data da hora, sendo esse o padrão utilizado pelos navegadores atuais usando o html.

java.time.**LocalDateTime**

//Alguns métodos utilizados

Boolean equals (object obj)

É utilizado para verificar se esta data-hora é igual a outra data-hora.

LocalDateTime.now()

\\2022-02-22T17:41:51.047569

LocalTime plusHours (longhoursToadd)

Retorna uma cópia deste LocalTime com o número especificado de minutos adicionados.

DateTimeFormatter

Utilizado para formatar desta vez usando o formatador especificado.

java.time.**LocalDateTime**

//Métodos mais utilizados

now()

Retorna a data atual, no formato
YYYY/MM/DD e HH/MM/SS.

of(int year, int month, int dayOfMonth, int hour, int minute)

Retorna a data especificada, composta
através dos valores passados pelos
parâmetros.

parse(CharSequence text)

Obtém uma instância de LocalDateTime de
uma string de texto como 2007-12-
03T10:15:30

plusDays(long days)

Adiciona a quantidade de dias
especificados no parâmetro.

java.util.**LocalDateTime**

//Demonstração de Uso

Usar a classe LocalDateTime para imprimir a data e o horário nesta formatação
YYYY-MM-DD-HH-MM-SS.

Exemplo: "22-02-2022 11:42:32"

//Referências

- **Calendar (Documentação Oracle)**

<https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html>

- **LocalDate (Documentação Oracle)**

<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html>

- **LocalDateTime (Documentação Oracle)**

<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDateTime.html>

- **Classe Calendar e exemplos de uso**

<https://www.geeksforgeeks.org/calendar-class-in-java-with-examples/?ref=lbp>

- **Utilizando LocalDate, LocalDateTime e LocalTime na Prática**

<https://blog.cvinicius.com.br/2018/08/utilizando-localdate-localdatetime-e.html>

- **LocalDate e exemplos de uso com ano bissexto**

<https://www.logicbig.com/how-to/code-snippets/jcode-java-8-date-time-api-localdate-isleapyear.html>

Obrigado!