



High performance. Delivered.

Application Delivery Fundamentals: Java

Module 5-6: Date

[illegible]

“Classes de DATAS”

Classes com Data

A **data** representa o tempo, um tempo é composto por ano, mês, dia atual, minuto atual, entre outras propriedades que essa **classe** possui. Na linguagem Java, as datas são instâncias da classe Date.

```
Date theDateToday = new Date()
```


- Copyright © 2023 Accenture All Rights Reserved.

- Copyright © 2023 Accenture All Rights Reserved. 5

Classes com Data

- **Эпока:**

- Ponto de referência na linha do tempo;
- Divide a linha de tempo em eras;
- Inicia em uma era particular;

- **Zonas de tempo;**

Região com um tempo padrão uniforme para propósitos legais, comerciais, sociais e políticos (fuso horário);

- **Cronologia;**

Sistema de contagem e tempo;

- ISO8601 (default): IsoChronology
- Chinese: MinguoChronology
- Japanese: JapaneseChronology
- Thai Buddhist: ThaiBuddhistChronology
- Islamic: HijrahChronology

Classes com Data

- **UTC** Tempo Universal que substituiu o **GMT**
 - **GMT** is Greenwich Mean Time
 - Mean solar time at the Royal Observatory in Greenwich
 - **UTC** is Coordinated Universal Time
 - Precisely defined with atomic time
 - Does not change with seasons
 - Replaced GMT as reference time scale on 1 January 1972

- Montreal, Canada**
⌚ 08:00, EDT
🌐 UTC-4
📅 2021-08-23

Helsinki, Finland
⌚ 15.00, EEST
🌐 UTC+3
📅 23.08.2021

New Orleans, USA
⌚ 07:00 AM, CDT
🌐 UTC-5
📅 08/23/2021

Edinburgh, UK
⌚ 01.00 PM, BST
🌐 UTC+1
📅 23/08/2021

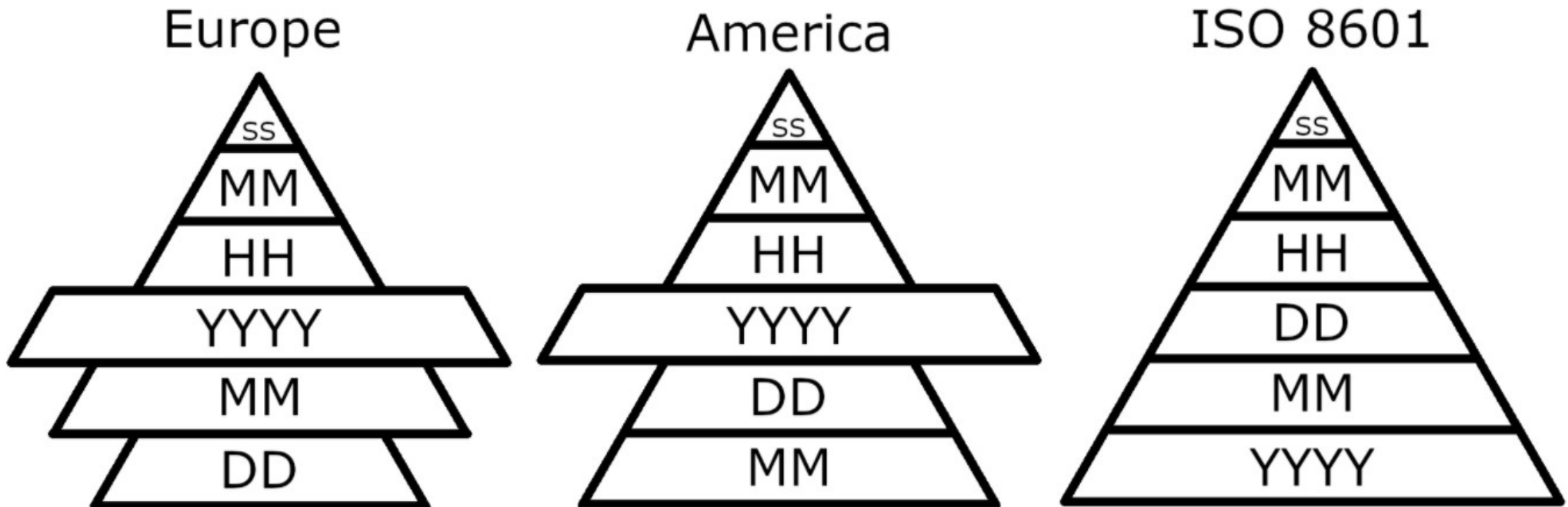
Classes com Data

- **ISO 8601**



Classes com Data

- **ISO 8601**



Packages

Classes mais utilizadas:

Formatação

Formatando Datas

- Para formatar datas em java use a classe `java.text.SimpleDateFormat`. Esta classe possui dois métodos básicos: `parse(String)` para converter um texto em uma data e `format(Date)` para formatar a data em um texto. Por exemplo:

```
SimpleDateFormat sdf = new  
    SimpleDateFormat("dd/MM/yyyy HH:mm");  
String texto = sdf.format(new Date());  
Date dia = sdf.parse("16/06/1971 23:00");
```

Formatação

- A classe `java.text.DateFormat` permite converter Strings de inputs do usuário para Dates, seguindo um determinado formato.
- Serve também para fazer o inverso: pegar uma data e gerar uma String de saída em um certo formato.
- A classe `SimpleDateFormat`, que estende `DateFormat`, é a que oferece a forma mais fácil de fazer tudo isso.

[illegible]

```
SimpleDateFormat formatador = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
```


Classe DATE

Calendário e Datas

- A classe `java.util.Date` representa uma data e a classe `java.util.Calendar` representa um calendário para trabalhar com as datas.
- `Date hoje = new Date();`
- `Calendar c = Calendar.getInstance();`

- ClasseData.java
- DataFormatada.java
- DataPadrao.java

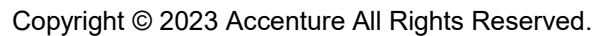
Classe Calendar

Calculando Datas

- A classe Calendar permite a definição de data com o método set e o cálculo de datas com o método add(int, int). Por exemplo:
- Calendario.java

Classe GregorianCalendar

- A classe abstrata **Calendar** encapsula um momento no tempo representado em milissegundos. Também provê métodos para manipulação desse momento.
- A subclasse concreta de Calendar mais usada é a **GregorianCalendar**, que representa o calendário usado pela maior parte dos países.

[illegible]

- ```
dateFormat.setTimeZone(TimeZone.getTimeZone("Asia/Bangkok"));
dateFormat.format(new Date());
```





[illegible]

- Duracao.java



[illegible]

- 
- A large, stylized question mark shape composed of numerous smaller question words. The most prominent words are "WHAT?", "HOW?", "WHY?", "WHERE?", "WHO?", and "WHEN?". Other visible words include "WHICH?", "WHOM?", "WHATEVER?", "WHETHER?", "WHOSE?", "WHY?", "HOW?", "WHERE?", "WHO?", "WHEN?", "WHICH?", "WHOM?", "WHATEVER?", "WHETHER?". The words are arranged in a way that they all contribute to the overall shape of the question mark, with some words appearing more frequently than others. The colors of the words vary, including shades of blue, green, yellow, orange, red, and black.

