

Module 4: Classe Date

## Classes de data

•

"Classes de DATAS"

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.

O modo mais direto de criar uma data é:

## Date the Date Today = new Date()

.

Tempo é visto de forma diferente entre humanos e o computador;

#### Computador:

Valor único sempre crescente;

#### • Humanos:

- Linha de tempo continua;
- Sistemas de calendários;
- Unidades como Ano, Mês, Dia, Horas, etc..
- Zonas de tempo;

## Conceitos de tempo;

#### Instante:

- É um ponto em uma linha do tempo;
- Armazenado em NanoSegundos;

## Tempo parcial;

Não identifica um tempo especifico;

## Duração:

- Tamanho preciso de um tempo decorrido em nanoSegundos;
- Pode ser negativa;
- Não é baseado em ano, mês ou dias;

## Período;

- Tamanho de um tempo decorrido;
- É baseado em ano, mês ou dias;
- Leva em consideração as zonas de tempo;

### Epoca:

- 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;

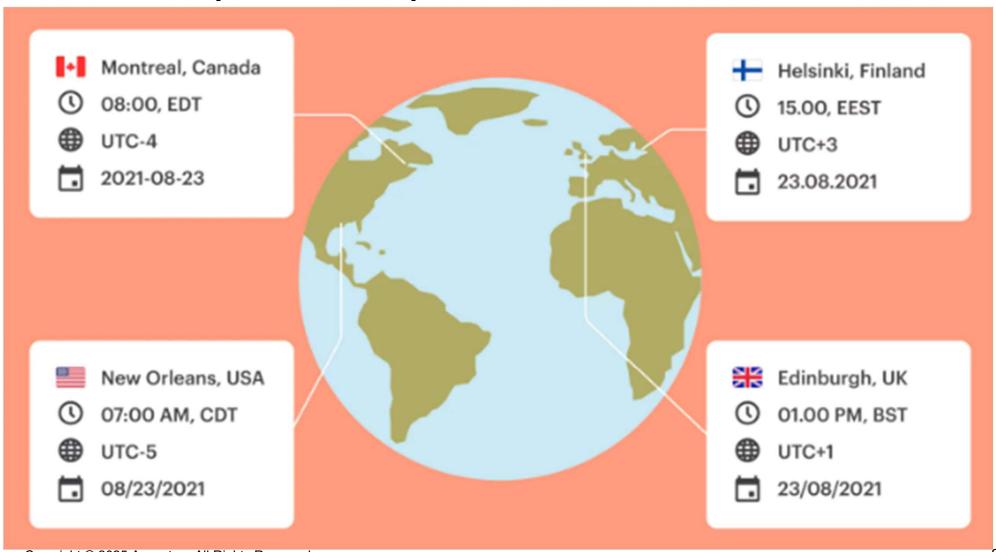
Sistema de contagem e tempo;

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

## 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

UTC Tempo Universal que substituiu o GMT

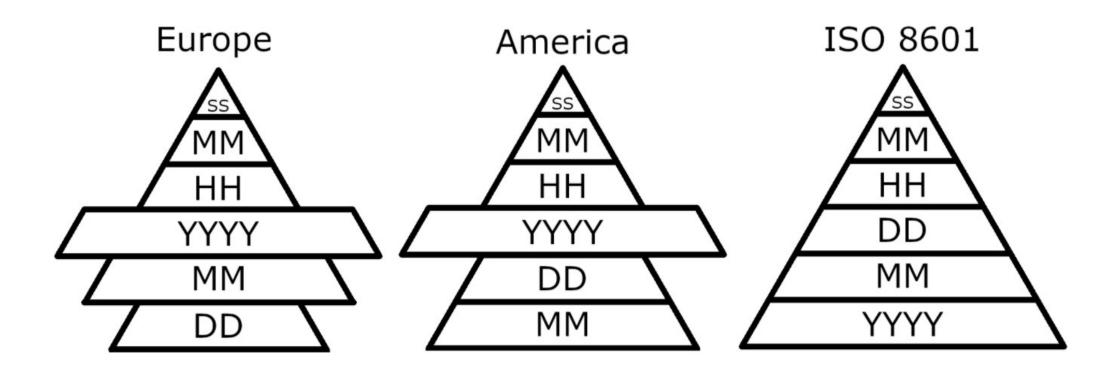


### ISO 8601

- Padrão internacional para representação das datas e tempos;
- Utiliza o calendário gregoriano;
- Ordenado do mais significativo para o menos significativo;
- ANO->MÊS->DIA->Hora->Minuto->...
- Cada data tem um número fixo de dígitos;
- Utiliza 4 digitos para o ano;



### ISO 8601



## **Packages**

#### Classes mais utilizadas:

- Calendar
- GregorianCalendar
- LocalDate
- Duration
- DateAndTimeFormater
- ZonedDateTime
- Date

- java.time instants, durations, dates, times, time zones, periods
- java.time.format formatting and parsing
- java.time.temporal field, unit, or adjustment access to temporals
- java.time.zone support for time zones
- java.time.chrono calendar systems other than ISO-8601

# 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");
```

- 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.

Para formatar uma data usamos uma classe formatadora;

Uma instancia da SimpleDateFormat;

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

### Formatos possíveis;

Letter	Description	Correct Usage With Examples
d	Day in month	dd → 10, 21
М	Month in year	MM → 09, 10 MMM → Sep, Oct MMMM → September, October
У	Year	yy → 15, 16 yyyy → 2015, 2016
E	Day name in week	E → Sat, Sun EEEE → Saturday, Sunday
Н	Hour in day (24hr format)	HH → 19, 20
h	Hour in day (12hr format)	hh → 07, 08 (AM or PM)
m	Minute in hour	mm → 31, 51
S	Second in minute	ss → 25, 33
а	AM or PM	a → AM/PM
z	Time Zone in text	z → IST zzzz → India Standard Time
Z	Time Zone in number	Z → +0530

## 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();

## **Classe DATE**

Recomenda-se evitar o uso da classe Date porque a maior parte de seus métodos estão marcados como deprecated, porém, ela ainda tem amplo uso legado nas bibliotecas do Java. Ela foi substituída no Java 1.1 pelo Calendar, para haver suporte correto à internacionalização do sistema de datas.

- 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

- 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.

- A classe Calendar numera os meses e os dias da semana da seguinte forma:
  - Os meses de janeiro (january) a dezembro (december) são numerados de 0 a 11.
  - Os dias da semana são numerados de 1 a 7, sendo o domingo (sunday) o primeiro dia.
    - Principais construtores:

GregorianCalendar()

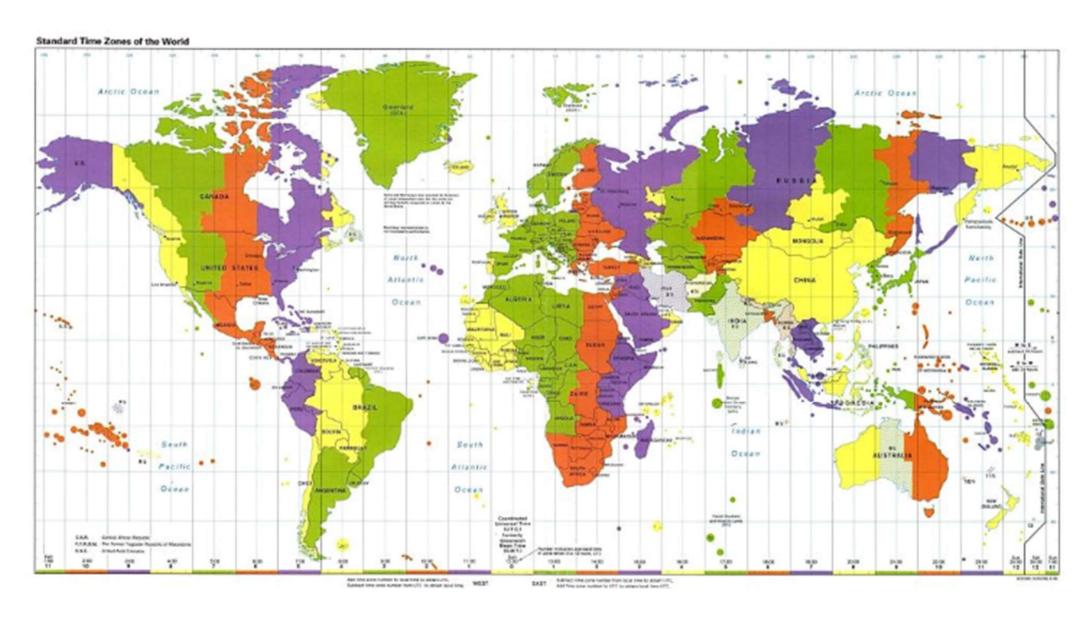
<u>GregorianCalendar</u>(int year, int month, int dayOfMonth)

GregorianCalendar(int year, int month, int dayOfMonth, int hourOfDay, int minute, int second)

Calendar c1 = new GregorianCalendar(2012, 4, 11); equivale a

Calendar c2 = new GregorianCalendar(2012, Calendar.MAY, 11);

CalendarioGregoriano.java



Método setTimeZone

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

## **Classe LocalDate**

- DiferencaDatas.java
- DataHoraLocal.java

## **Classe Duration**

244869676820706572666f726d616e63652e2044656c6976657265642e2f48696768207065726667766442e2f48696768207065726667766446566976642e2f4869676820706572666776644656697664465669766446566976644666697664466669766446666976644666697664466669766466669766466669766466697664666697666776676

Duracao.java

## **Classe DateTimeFormatter**

ZonedDateTimeExample.java

## **Questions and Comments**

 What questions or comments do you have?



