

# Linguagem de Programação Java

O livro é um mestre que fala mas que não responde.

Platão

# CLASSES ÚTEIS



# Classes Úteis

Neste capítulo, estudaremos algumas Classes existentes na biblioteca Java, e que são muito úteis no dia a dia de um desenvolvedor.



# Exemplo ***16\_ClassesUteis***

Crie o projeto *16\_ClassesUteis* e o pacote *controller*.



Trabalharemos diretamente em Classes dentro do pacote *controller*, para que sejam demonstradas as funcionalidades das Classes Úteis.

# Classe *ExemploMetodoString*

Digitar dentro do método *main*:

```
String url = "http://www.senai.rj.br/ctti/turma2010011";  
String email = "aluno@senai.br";
```

```
String dominio;  
dominio = url.substring(7,22);  
System.out.println("Domínio: " + dominio);
```

Variável *dominio* recebe os caracteres entre as posições 7 e 22 da variável *url*

```
String complemento;  
complemento= url.substring(23);  
System.out.println("Complemento do Domínio: " + complemento);
```

Variável *complemento* recebe os caracteres a partir da posição 23 da variável *url*

# Classe *ExemploMetodoString* (Cont.)

Continuando o código:

```
String nomeUsuario;  
nomeUsuario = email.substring(0, email.indexOf('@'));  
System.out.println("Usuário e-mail: " + nomeUsuario);
```

Variável *nomeUsuario* recebe os caracteres da posição 0 até a posição anterior ao @ da variável *email*

```
String maiuscula, minuscula, palavra1, palavra2;  
palavra1 = "carro";  
palavra2 = "CASA";  
maiuscula = palavra1.toUpperCase();  
System.out.println("maiuscula: " + maiuscula);  
minuscula = palavra2.toLowerCase();  
System.out.println("minuscula: " + minuscula);
```

O conteúdo da variável *palavra1* será convertido em letras maiúsculas.

O conteúdo da variável *palavra2* será convertido em letras minúsculas.

# Classe *ExemploClasseMath*

Digitar dentro do método *main*:

```
// Sorteando um numero entre 50 e 80
```

```
double valSorteado = Math.random();
```

```
int resultado = 50 + (int) (valSorteado * 30);
```

```
System.out.println("Sorteio entre 50 e 80: " + resultado);
```

```
// Sorteando a mega-sena
```

```
for (int i = 0; i < 6; i++){
```

```
    double numeroSena = Math.random();
```

```
    int numeroSorteado = (int) (numeroSena * 61);
```

```
    System.out.println("Número Sorteado: " + numeroSorteado);
```

```
}
```

*Math.random()* faz um sorteio de um número decimal entre 0 e 1

# Classe *ExemploClasseMath* (Cont.)

Continuando o código:

```
// Maior entre dois números
```

```
int maior;
```

```
maior = Math.max(resultado, 65);
```

```
System.out.println("Entre 65 e " + resultado +  
    " o maior valor é " + maior);
```

```
// Área de um círculo de raio 5
```

```
double area = Math.PI * Math.pow(5, 2);
```

```
System.out.println("Área de um círculo de " +  
    "raio 5 é " + area);
```

*Math.PI* = valor de Pi  
*Math.pow* = Potência



# Classe *ExemploClasseMath* (Cont.)

Continuando o código:

```
double sen, cos, tan;  
double a=60;  
sen = Math.sin(Math.toRadians(a));  
cos = Math.cos(Math.toRadians(a));  
tan = Math.tan(Math.toRadians(a));  
System.out.println("Seno: " + sen);  
System.out.println("Cosseno: " + cos);  
System.out.println("Tangente: " + tan);
```

//Raiz Quadrada

```
double raiz = Math.sqrt(4);  
System.out.println("Raiz quadrada de 4 é " + raiz);
```

# Projeto *InfoNote\_09*

## Objetivos:

- Elaborar rotina que converte o nome do Cliente para o padrão americano (Sobrenome, Nome)
- Criar a Classe GerarSenha, que gera uma senha com caracteres aleatórios caso o usuário deixe sua senha em branco.
- Tornar funcionais as duas tarefas acima.



Pressione o botão vermelho para  
abrir o documento passo a  
passo desta tarefa.

# Formatação de Números



- Para formatar números deve-se utilizar as classes *NumberFormat* ou *DecimalFormat*.
- A classe *NumberFormat* possui métodos para retornar os formatos pré-definidos de valores numéricos, monetários e percentuais.
- A classe *DecimalFormat* possibilita que a formatação seja personalizada.

# Utilização da Classe *NumberFormat*

Permite obter uma instância da classe para o tipo de formatação desejada:

- `NumberFormat.getCurrencyInstance ( );` Monetário
- `NumberFormat.getNumberInstance ( );` Numeral
- `NumberFormat.getPercentInstance ( );` Percentual

Permite formatar o valor desejado através do método `format ( )` da Classe `String`.

# Utilização da Classe *DecimalFormat*

- A formatação pode ser personalizada com uma máscara.
- O método de formatação também é *format*.
- Caracteres de formatação:

<b><i>Caracter</i></b>	<b><i>Funcionalidade</i></b>
0	Dígito
,	Separador de milhar
#	Dígito, em branco se for zero
-	Mostra sinal sempre
%	Porcentagem
.	Ponto decimal

# Classe *ExemploFormatacaoNumero*



Início:

```
public static void main(String[] args) {
```

```
    NumberFormat f = NumberFormat.getCurrencyInstance();
```

```
    double valor = 55008.537;
```

```
    System.out.println("Sem formatação: " + valor);
```

```
    System.out.println("Formatado: " + f.format(valor));
```

```
    NumberFormat p = NumberFormat.getPercentInstance();
```

```
    double por cento = 0.05;
```

```
    System.out.println("Sem formatação: " + por cento);
```

```
    System.out.println("Formatado: " + p.format(por cento));
```

# Classe *ExemploFormatacaoNumero*

```
NumberFormat n = NumberFormat.getNumberInstance();  
double numero = 100.50;  
System.out.println("Sem formatação: " + numero);  
System.out.println("Formatado: " + n.format(numero));  
  
double valor2 = 12000.5;  
  
DecimalFormat df = new DecimalFormat("R$ ###,###,###.00");  
System.out.println("Sem formatação: " + valor2);  
System.out.println("Formatado: " + df.format(valor2));
```

**Fim:**

# Classe *ExemploInternacionalizacao*

```
public static void main(String[] args) {  
  
    Locale.setDefault(Locale.FRANCE);  
    NumberFormat f = NumberFormat.getCurrencyInstance();  
    double valor = 55008.537;  
    System.out.println("Sem formatação: " + valor);  
    System.out.println("Formatado: " + f.format(valor));  
  
    //Normas: ISO 639 -> Idioma ; ISO 3166 -> Localidade  
    Locale local = new Locale("pt", "BR");  
    NumberFormat nf = NumberFormat.getCurrencyInstance(local);  
    System.out.println("Sem Formatação: " + valor);  
    System.out.println("Formatado: " + nf.format(valor));  
  
}
```



# Classe *ExemploFormatData*

## Parte 1

```
public static void main(String[] args) {  
  
    Date data = new Date();  
    System.out.println("Data " + data);  
  
    @Deprecated  
    Date dataInserida = new Date(2011,9,14);  
    System.out.println("Data Inserida" + dataInserida);  
    //Esta Classe está obsoleta.  
  
    Calendar data2 = new GregorianCalendar();  
    System.out.println("Data: " + data2);  
}
```

# Classe *ExemploFormatData*

Parte 2

```
data2.set(2011, 3, 19);  
System.out.println("Mês :" + data2.get(Calendar.MONTH));  
System.out.println("Dia da Semana: " +  
    data2.get(Calendar.DAY_OF_WEEK));  
System.out.println("Dia do Mês: " +  
    data2.get(Calendar.DAY_OF_MONTH));  
System.out.println("Dia do Ano: " + data2.get(Calendar.DAY_OF_YEAR));  
  
DateFormat df3 = DateFormat.getDateInstance(DateFormat.FULL);  
Calendar data3 = new GregorianCalendar();  
data3.set(2010,8,25);  
System.out.println("Data 3 (FULL): " +  
    df3.format(data3.getTime()));
```



# Classe *ExemploFormatData*

## Parte 3

```
DateFormat df4 = DateFormat.getDateInstance(DateFormat.LONG);  
Calendar data4 = new GregorianCalendar();  
data4.set(2010,8,25);  
System.out.println("Data 4 (LONG): " + df4.format(data4.getTime()));  
  
DateFormat df5 = DateFormat.getDateInstance(DateFormat.MEDIUM);  
Calendar data5 = new GregorianCalendar();  
data5.set(2010, 8, 25);  
System.out.println("Data 5 (MEDIUM): "+ df5.format(data5.getTime()));
```

# Classe *ExemploFormatData*

Parte 4 - Fim

```
DateFormat df6 = DateFormat.getDateInstance(DateFormat.SHORT);  
Calendar data6 = new GregorianCalendar();  
data6.set(2010,8,25);  
System.out.println("Data 6 (SHORT): "+ df6.format(data6.getTime()));  
  
TimeZone tz = TimeZone.getTimeZone("GMT-3:00");  
TimeZone.setDefault(tz);  
System.out.println(DateFormat.getTimeInstance().format(new Date()));
```

# Projeto *InfoNote\_10*

## Objetivos:

- Inserir ao lado do slogan da InfoNote a data e a hora do sistema.
- Alterar o método *toString* da Classe *Notebook* para que ele formate o preço do Notebook sempre em Real.
- Desafio: Fazer a formatação de números e preços nas demais Classes do projeto.

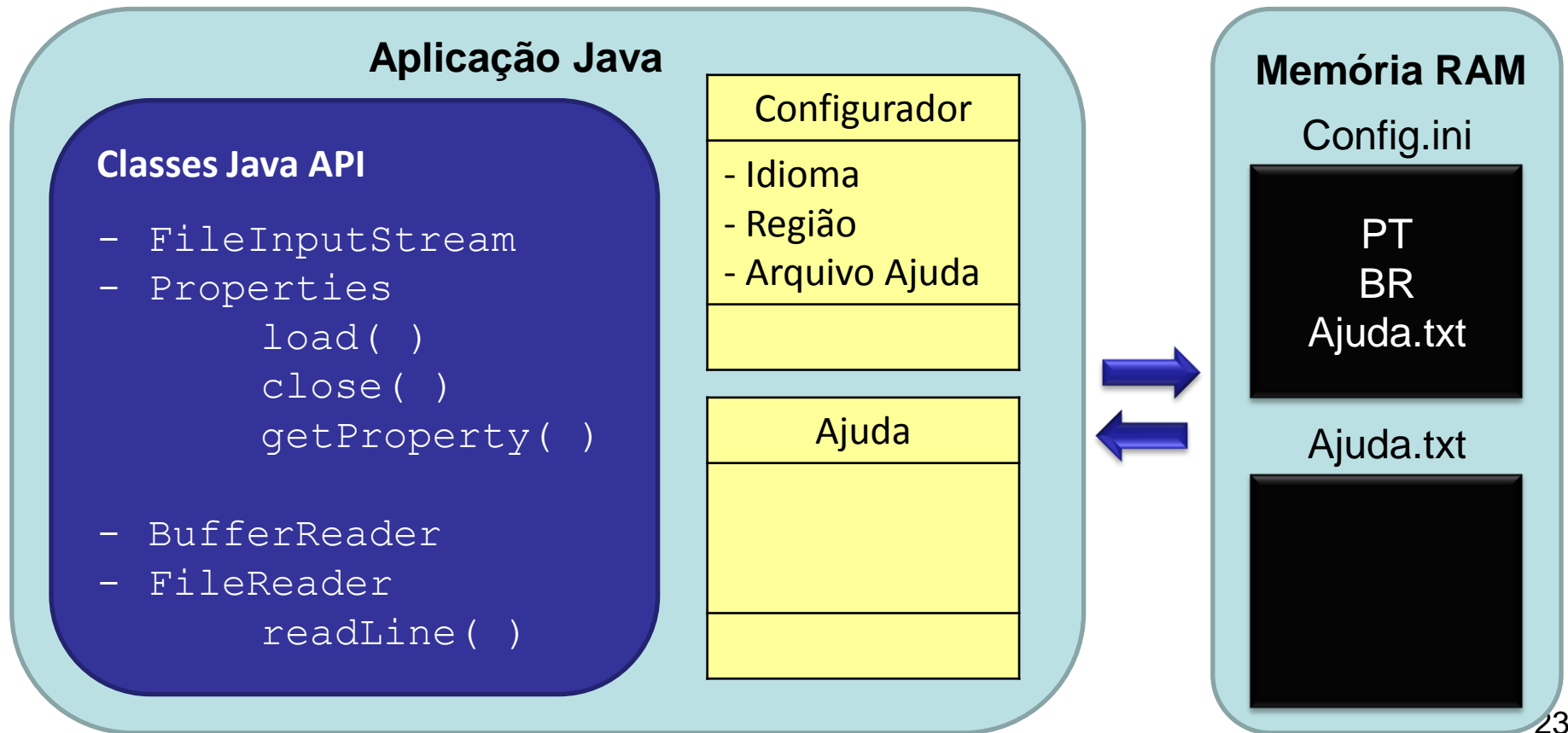


Pressione o botão vermelho para  
abrir o documento passo a  
passo desta tarefa.

# Streams (Fluxos)

- Streams são canais de comunicação entre um programa e uma fonte de dados.
- As fontes de dados podem ser: arquivos, trechos de memória e conexões de rede.
- Em Java não importa qual seja a fonte, todo o tratamento de dados é igual.
- Fontes das quais lemos dados são fluxos de entrada e aquelas nas quais gravamos dados são fluxos de saída.

# Projeto *InfoNote\_11*



# Projeto *InfoNote\_11* (Cont.)

## Objetivos:

- Elaborar arquivo de ajuda para o sistema.
- Elaborar arquivo *config.ini*, com indicadores de configurações.
- Criar a Classe *Configurador* no pacote *util*, para definição de propriedades do sistema. Neste exemplo, para o idioma.
- Criar a Classe *Ajuda* no pacote *util*, para mostrar a ajuda do sistema.



Pressione o botão vermelho para  
abrir o documento passo a  
passo desta tarefa.



*JAVA2 - TI - 03.3 - Instruções Projeto InfoNote\_11.pdf*



# Dúvidas?



# Bibliografia



Java Como Programar 8ª Edição  
Paul Deitel e Harvey Deitel  
Ed. Pearson



Java 7 Ensino Didático  
Sérgio Furgeri  
Ed. Érica