

# Disciplina Regular 1

# Fundamentos do

# Desenvolvimento Java

Graduação em Engenharia de Software - 2020

# Etapa 7 Aula 2

Classes Úteis

# Competências Trabalhadas Nesta Etapa

- Utilizar classes padrão fundamentais do Java:
  - Utilizar os construtores com parâmetros da classe String.
  - Utilizar os métodos da classe String.
  - Compreender as principais diferenças entre as classes String e StringBuilder.
  - Utilizar os métodos da classe StringBuilder.
  - Manipular data e hora com a classe Date.
  - Formatar data e hora com DateFormat e SimpleDateFormat.

No Moodle esse conteúdo se refere à etapa 6 e 8

# Classes Úteis

# String

- Uma string é uma sequência de caracteres.
- A classe **String** é usada para representar *strings* em Java.
- São objetos somente de leitura, ou seja, não podem ter seu valor modificado após a criação.

# String

- Algumas formas de criação.

```
String autor = "João Ubaldo Ribeiro"; //forma simplificada
```

```
//Concatenação de objetos String
```

```
String autor = "João " + "Ubaldo " + "Ribeiro";
```

```
//Forma comum de criação de objetos usando o operador new.
```

```
String autor = new String("João Ubaldo Ribeiro");
```

# String

```
int      length ();
```

```
String   concat (String);
```

```
String   replace (char, char);
```

```
String   toLowerCase ();
```

```
String   toUpperCase ();
```

```
String   trim ();
```

```
char     charAt (int);
```

```
char []  toCharArray ();
```

# String

```
String    substring (int, int)
int       indexOf   (String);
int       lastIndexOf (String);
boolean   equals    (String)
int       compareTo (String);
boolean   startsWith (String);
boolean   endsWith  (String);

boolean   matches (String);
String[]  split (String);
```



# String

- O método **substring()** pode receber um ou dois parâmetros inteiros e retorna um subconjunto da String original:

```
String url = "http://www.infnet.com.br/curso/javaprogramming1/index.htm";  
String dominio, teste;  
  
dominio = url.substring(7,24); //objeto - ponto - nome_metodo - (parâmetros)  
System.out.println("Dominio = " + dominio);  
  
teste = url.substring(25);  
System.out.println("teste = " + teste);
```

```
Dominio = www.infnet.com.br  
teste = curso/javaprogramming1/index.htm  
Press any key to continue . . .
```

# String

- Método `valueOf()` – converte qualquer variável de tipo primitivo em uma `String`.

```
String cargaHoraria = String.valueOf(40);  
String preço = String.valueOf(5.2);
```

# StringBuffer / StringBuilder

- Representa uma sequência mutável de caracteres Unicode. Por ser mutável, é a candidata natural quando uma série de concatenações se faz necessária.  
Não tem relação alguma com a classe String.

```
//Método para ler um string do final para o início
```

```
public String reverter(String s) {  
    StringBuffer sb = new StringBuffer();  
    for (int i = s.length() - 1; i >= 0; i--)  
        sb.append(s.charAt(i));  
    return sb.toString();  
}
```

# StringBuffer

```
StringBuffer append (...);  
StringBuffer insert (int pos, ...);  
  
char      charAt (int pos);  
void      setCharAt (int pos, char Ch);  
  
StringBuffer delete (int ini, int fim);  
StringBuffer reverse();
```

# Manipulação de Números

# Classe Math

- Definida no pacote `java.lang`.
- Nenhum objeto dessa classe pode ser criado, pois seu construtor é privado.
- Todos os seus métodos são estáticos, por isso o acesso é feito com o nome da classe.
- Não pode ser estendida por herança.
- Possui duas constantes para cálculos.

# Classe Math

ceil(), floor(), round().

abs(), min(), max().

sin(), cos(), tan(), asin(), acos(), atan(),

toDegrees(), toRadians().

log(), exp().

sqrt(), pow(), random().

Constantes PI, E

# Classe Math

```
//Sorteando um número entre 50 e 80
double valSorteado = Math.random() ;
int resultado = 50 + (int) (valSorteado*30);
System.out.println("Sorteio entre 50 e 80: " + resultado );

//Maior entre dois números
int maior;
maior = Math.max(resultado,65);
System.out.println("Entre 65 e " + resultado +" o maior eh " + maior);

//Área de um círculo de raio 5
double area = Math.PI * Math.pow(5,2);
System.out.println("Area de um circulo de raio 5 eh " + area);
```



# Wrappers

- Permite a utilização de tipos primitivos como objetos.

boolean	Boolean
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double

# Internacionalização

# Formatação de Números

- Para formatar números devem ser utilizadas as classes **NumberFormat** ou **DecimalFormat**.
- A classe **NumberFormat** possui métodos para retornar os formatos predefinidos de valores numéricos, monetários e percentuais.
- A classe **DecimalFormat** possibilita a customização da formatação.

# Formatação de Números

- Obter uma instância da classe para o tipo de formatação desejada, utilizando os métodos estáticos.

```
getCurrencyInstance() ; // Dinheiro  
getNumberInstance() ; // Número  
getPercentInstance() ; // Percentual
```

- Formatar o valor desejado com o método format.

# Formatação de Datas

- A classe Date obtém a data do sistema ou cria uma data específica no tempo (data/hora).
- Vários métodos obsoletos (@Deprecated).
- Métodos mais usados:

Date()  
Date(long)  
after, before,  
clone, compareTo, equals, hashCode  
getTime, setTime, toString

# Formatação de Datas

- A classe **DateFormat** e sua subclasse **SimpleDateFormat** são usadas para a formatação de datas.

- Podem ser criadas instâncias com data e com data e hora:

```
DateFormat df = DateFormat.getDateInstance();  
DateFormat df = DateFormat.getDateTimeInstance();
```

- Podem ser passados como parâmetros o objeto **Locale** desejado e o tipo de formatação: SHORT, MEDIUM, LONG, FULL.

# Classe Locale

- Classe que representa uma região e um idioma.
- Muito usada em métodos de formatação.
- Para criar um Locale diferente, basta passar as siglas referentes à região e ao idioma ou usar uma das constantes disponíveis.
- O Locale padrão é o configurado no SO. Para alterar, utilize:  
`Locale.setDefault(novo_locale);`

# Exercício

- Criar um programa que use o método `getAvailableLocales` da Classe `Locale` para exibir uma lista com as seguintes informações das regiões:
  - Nome do País.
  - Nome do Idioma.
  - Moeda Formatada.
  - Data Completa Formatada.