ISUTC INSTITUTO **SUPERIOR** DE
TRANSPORTES E **COMUNICAÇÕES**

Programação 1

Sumário:

- Boas práticas de programação,
- Casting,
- Class Math,
- Class Random,

Objectivos:

- Conhecer os diversos tipos de comentários existentes na linguagem java;
- Identificar melhores locais para os utilizar;
- Conhecer o modelo de funcionamento de uma operação do tipo ***cast***.
- Usar os diferentes elementos da ***class Math e class Random*** para operações de grande complexidade ou em situações necessárias

Comentários: definição

São técnicas utilizadas para ocultar linhas de código, para que o compilador não as processe ou documentar o código. Estes podem ser segmentadas em:

- **Comentários por linhas [ex. //]** – muitas vezes utilizados para comentar *statements*, por cada linha de código inscrito dentro de um programa.
- **Comentários por blocos [ex. /* e */]** – utilizado para comentar um grupo de *statements* dentro de um programa.
- **Comentários para documentacão automática [ex. /** e */]** – utilizado para documentar códigos de forma automática [frequentimente usado em ferramenta de desenvolvimento avançadas].

```
// este é comentário de 1 linha

/*
    Este é um comentário
    com mais de
    1 linha
*/

/*
* Este é um comentário
*  tambem com mais
*  de 1 linha
*/

 /**
   * modifica o texto do menu
   * @param texto texto do menu
   **/
   public modificaTextoMenu(String texto) { /* codigo */ }
```

Casting: definição

Processo que permite em tempo real de execução de um programa em java converter uma variável definida em um determinado tipo de dado para outro.

Representa –se em*:*

[TIPO_DE_DADO] x = ([TIPO_DE_DADO_de_x]) y ;

x e y = variáveis

Alguns pormenores para realizar o casting:

- Analisar os tipos de dados intervenientes;
- Testar a compactibilidade em função da sua árvore genética.

Casting: árvore genética

DE \ PARA	byte	short	char	int	long	float	double
byte		Implícito	char	Implícito	Implícito	Implícito	Implícito
short	byte		char	Implícito	Implícito	Implícito	Implícito
char	byte	short		Implícito	Implícito	Implícito	Implícito
int	byte	short	char		Implícito	Implícito	Implícito
long	byte	short	char	int		Implícito	Implícito
float	byte	short	char	int	long		Implícito
double	byte	short	char	int	long	float	

Casting: exemplos

```
char a = 'a';
int b = 'b';
float c = 100;

System.out.println(a); //Imprime a
System.out.println(b); //Imprime 98
System.out.println(c); //Imprime 100.0

int d = (int) 5.1987;
float e  = (float) 5.0;
int f = (char) (a + 5);
char g = (char) 110.5;

System.out.println(d); //Imprime 5
System.out.println(e); //Imprime 5.0
System.out.println(f); //Imprime 102
System.out.println(g); //Imprime n
```

Class Math: definição

A classe *Math* proporciona-nos uma série de operações e constantes matemáticas que são facilmente acessadas estaticamente [ou seja, não precisamos instanciar uma classe para podermos utilizar seus métodos].

Método	Descrição
Math.random()	Retorna um numero aleatório que vai de zero ate um (0 será incluido mas o 1 não sera)
Math.sqrt(double x)	Retorna a raiz quadrada do número passado.
Math.PI	Retorna o valor da constante PI.
Math.ceil(double x)	Retorna o maior número inteiro (menor que passado como parâmetro).
Math.floor(double x)	Retorna o maior número inteiro (não menor que o passado com parâmetro).
Math.round(double x)	Retorna o long mais próximo do parametro passado.
Math.pow(double x, double y)	Para uma estrutura de potenciação x^y.

Class Math: exemplos

```
double a = 5.2
double b = 5.6
System.out.print("a = "+Math.ceil(a));  // imprime 6.0
System.out.print("b = "+Math.ceil(b));  // imprime 6.0

System.out.print("a = "+Math.floor(a));  // imprime 5.0
System.out.print("b = "+Math.floor(b));  // imprime 5.0

int aleatorio = (int) (Math.random() * 100);
System.out.print("aleatorio = "+aleatorio);
// imprime um número aleatorio no intervalo de 0 ate 99
```

Class Random: import java.util.Random;

A classe *Random* proporciona-nos a geração de números aleatórios. Os números aleatórios são utilizados de diversas formas em programas de computador. Eles são importantes no **desenvolvimento de jogos**, na área de **segurança de informações** (ex: para gerar senhas ou textos de campos captcha).

Método	Descrição
nextInt()	Retorna um número inteiro (negativo ou positivo) aleatório.
nextInt(int x)	Retorna um número inteiro (negativo ou positivo) aleatório no intervalo de [0,x-1]
nextBoolean()	Retorna booleanos (*true* ou *false*)
nextFloat()	Retornam números reais entre 0 e 1
nextDouble()	Retornam números reais entre 0 e 1

Class Random: import java.util.Random;

```
import java.util.Random;
public class Test{
    public static void main(String[] args){
        Random random = new Random();
        int x = random.nextInt(10);
        boolean y = random.nextBoolean();
        double z = random.nextDouble();
        float w = random.nextFloat();
```

Exemplos

1. Crie uma classe que simule a jogada de um dado (de seis lados) dez vezes e mostre o resultado na tela.
2. Faça um programa que permite de determinar as raízes de uma equação quadrática.

GARANTE O TEU FUTURO

COM UMA FORMAÇÃO SÓLIDA

ISUTC

Prolong. da Av. Kim Il Sung (IFT/TDM) Edifício D1
Maputo, Moçambique

www.facebook.com/isutc

www.transcom.co.mz/isutc