

Programação Orientada a Objetos

Prof. José Antonio Gonçalves



POO – A linguagem de Programação Java

Conhecendo a Plataforma Java

Sabemos que a Orientação a Objetos é uma maneira, mais natural e simples, de se pensar, elaborar, documentar e construir aplicações e sistemas de informação.

Baseada nos conceitos de **abstração, atributos, classes, objetos**, etc. temos um novo paradigma para construção de aplicações e a tecnologia **Java** nos permite implementar todos estes conceitos, pois a linguagem é totalmente aderente às técnicas de Orientação a Objetos.

POO – A linguagem de Programação Java

APIs (interface de programas aplicativos):

Java 2 Standard Edition (J2SE): ferramentas e APIs essenciais para qualquer aplicação Java (inclusive GUI).

Java 2 Enterprise Edition (J2EE): ferramentas e APIs para o desenvolvimento de aplicações distribuídas.

Java 2 Micro Edition (J2ME): ferramentas e APIs para o desenvolvimento de aplicações para aparelhos portáteis.

POO – A linguagem de Programação Java

Tipos de programas Java

Stand-Alone: Aplicação baseada na J2SE, que tem total acesso aos recursos do sistema, memória, disco, rede, dispositivos, etc. Uma estação de trabalho pode executar uma aplicação de Automação Comercial.

Java Applets: Pequenas aplicações, que não têm acesso aos recursos do hardware e depende de um navegador que suporte a J2SE para serem executados, geralmente usados para jogos, animações, teclados virtuais, etc.

Java Servlets: Programas escritos e preparados para serem executados dentro de servidores web baseados em J2EE. Geralmente usados para gerar conteúdo dinâmico de web sites.

Java Midlets: Pequenas aplicações, extremamente seguras e construídas para serem executadas dentro da J2ME, geralmente celulares, palm tops, controladores eletrônicos, computadores de bordo, etc.

Java Beans: Pequenas aplicações que seguem um padrão bem rígido de codificação e que têm o propósito de serem reaproveitados em qualquer tipo de programa Java, podendo ser chamados por aplicações Stand-Alone, Applets, Servlets e Midlets.

POO – A linguagem de Programação Java

Lista de Palavras Reservadas

Nenhuma das palavras reservadas podem ser usadas com nome de objetos, classes, atributos ou métodos.

abstract	double	int	static
boolean	else	interface	super
break	extends	long	switch
byte	final	native	synchronized
case	finally	new	this
catch	float	null	throw
char	for	package	throws
class	goto*	private	transient
const*	if	protected	try
continue	implements	public	void
default	import	return	volatile
do	instanceof	short	while

* reservadas, entretanto não são usadas, pois não representam instruções da JRE;

POO – A linguagem de Programação Java

Como comentado anteriormente, o Java usa as duas formas computacionais de execução de software, compilação e interpretação.

O uso das duas formas faz com que o processo de desenvolvimento do software se torne mais acelerado, pois linguagens somente compiladas, necessitam de instruções adicionais da plataforma operacional para serem executados, e as linguagens somente interpretadas geralmente são muito lentas. Por isso o Java num primeiro momento é compilado (transformar código fonte) em instruções da Java Virtual Machine (bytecode).

Como o processo de execução é dividido em dois, temos a possibilidade de que o interpretador seja escrito para várias plataformas operacionais.

Ou seja, o mesmo bytecode Java não importando onde foi compilado, pode ser executado em qualquer JRE. Os fabricantes de hardware e sistemas operacionais, em conjunto com a JavaSoft, desenvolveram várias JREs para vários ambientes operacionais. Por isso, aplicações 100% Java são 100% portáteis.

POO – A linguagem de Programação Java

Compilando...

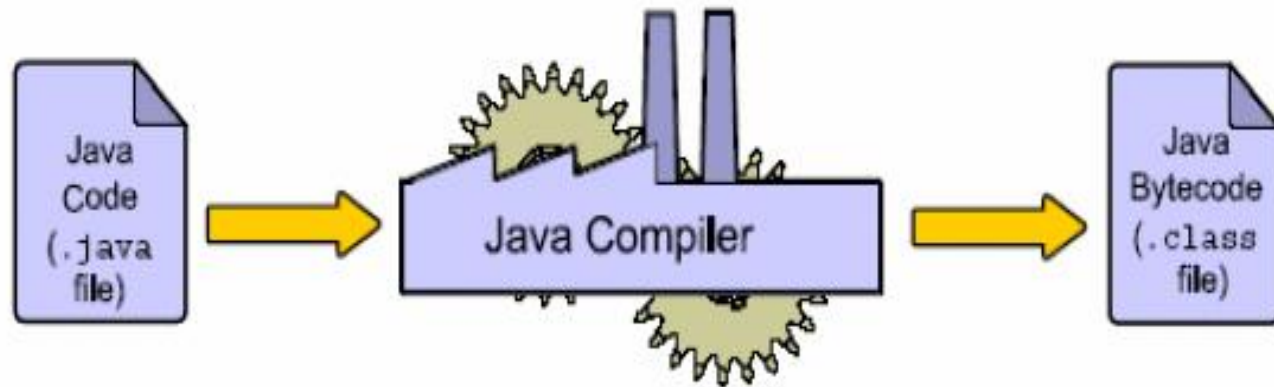


Figura 2

POO – A linguagem de Programação Java

Executando...

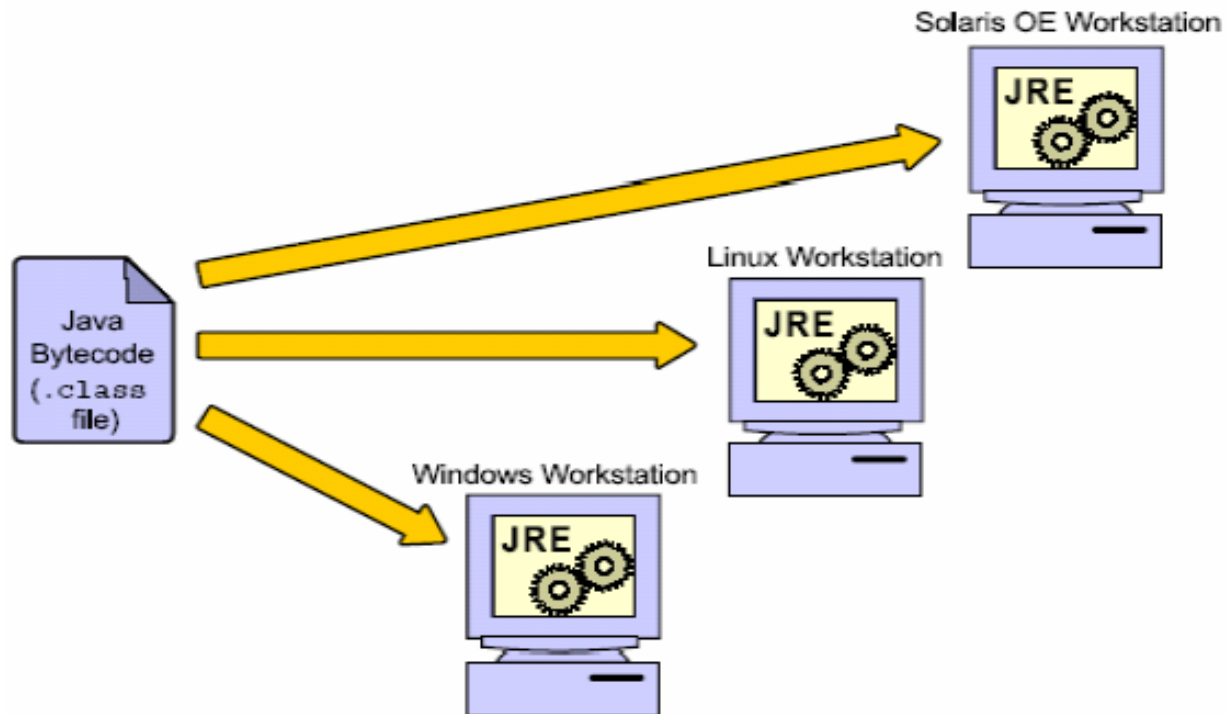


Figura 3

POO – A linguagem de Programação Java

Integrado a JRE, existe ainda um componente chamado **Garbage Collector**, que é responsável por coletar da memória objetos e dados que estão sendo mais usados.

O **Garbage Collector**, tem seu próprio algoritmo e ciclo de execução, por isso não temos controle sobre ele. Assim, em Java, o programador não precisa se preocupar em alocar e desalocar memória quando lida com objetos. No C ou C++, é de responsabilidade do programador o gerenciamento de memória.

Em Java, quando não precisamos mais de um objeto, basta anular sua referência elegendo-o a ser coletado, e o Garbage Collector fará o resto, garantindo a saúde da memória em que a aplicação está sendo executada.

Para cada conjunto, sistema operacional e hardware, existe uma JRE otimizada, capaz de garantir a execução de sistemas baseados em Java.

POO – A linguagem de Programação Java

Existem um documento formal, chamado Java Coding Conventions™, presente no site da JavaSoft que contém todas as regras de codificação e nomenclatura.

Anatomia de uma classe Java.

Todas as aplicações Java, independente do seu tipo, obrigatoriamente deve estar definida dentro de uma classe Java. E os objetos do nosso sistema, são representados em classes.

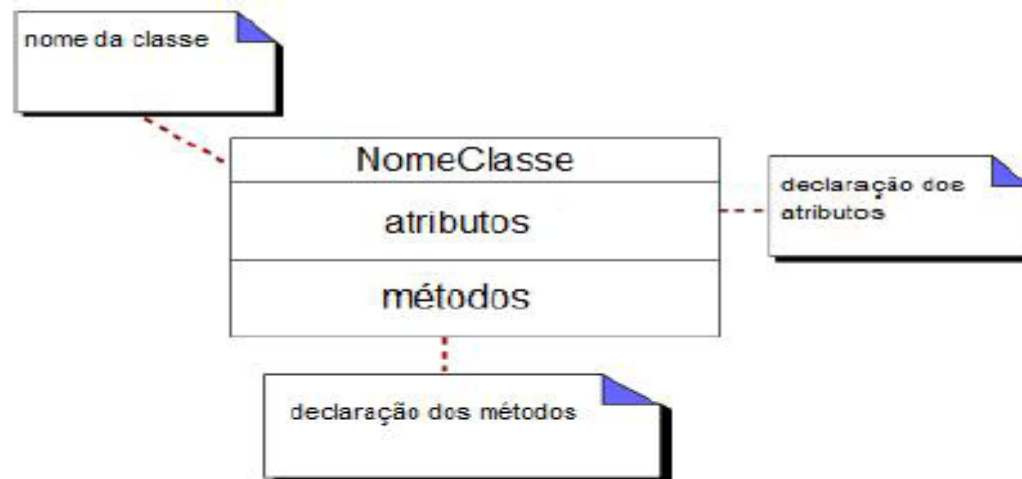


Figura 5

Nome da classe

Um identificador para a classe, que permite referenciá-la posteriormente.

POO – A linguagem de Programação Java

Definindo uma classe

Uma classe define-se como abaixo e ela é composta por membros, atributos e/ou métodos.

```
[modificador] class [identificador] {  
  
    // conjunto de atributos  
  
    // conjunto de métodos  
}
```

Examinando o código abaixo podemos identificar os conceitos citados acima:

```
public class PrimeiroProgramaJava {  
  
    public static void main(String args[]) {  
        System.out.println("Bem vindo a Universidade Java!");  
    }  
}
```

POO – A linguagem de Programação Java

Identificamos o seguinte:

`public, static` – **modificadores;**

`PrimeiroProgramaJava, main` – **identificadores;**

`public static void main(String args[]) { }` – **método;**

`String args[]` – **argumento do método;**

`String` – **tipo de dados;**

`void` – **tipo de retorno do método;**

`class` – **tipo para classes;**

`System.out.println` – **chamada de um método de uma outra classe;**

POO – A linguagem de Programação Java

Definindo Atributos

O conjunto de propriedades da classe. Para cada propriedade, especifica-se:

```
[modificadores] tipo [nome] [= valor_inicial];
```

nome: um identificador para o atributo.

tipo: o tipo do atributo (inteiro, real, caráter, outra classe, etc.)

modificadores: opcionalmente, pode-se especificar o quão acessível é um atributo de um objeto a partir de outros objetos. Valores possíveis são:

- (privativo), nenhuma visibilidade externa;
- + (público), visibilidade externa total;
- # (protegido), visibilidade externa limitada.

POO – A linguagem de Programação Java

Definindo Métodos

O conjunto de funcionalidades da classe. Para cada método, especifica-se sua **assinatura**, composta por:

```
[modificadores] tipo_de_retorno [nome] ([argumentos]) {  
    // corpo do método;  
}
```

nome: um identificador para o método.

tipo_de_retorno: quando o método tem um retorno, o tipo desse valor.

lista de argumentos: quando o método recebe parâmetros para sua execução, o tipo e um identificador para cada parâmetro.

modificadores: como para atributos, define o quão visível é um método a partir de objetos de outras classes.

POO – A linguagem de Programação Java

Entendendo uma aplicação Java Stand-Alone.

Para que uma aplicação Java possa ser considerada como Stand-Alone e para que ela possa ser executada diretamente pelo interpretador, como vimos anteriormente, ela deve possuir esse método, da forma como está escrito.

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

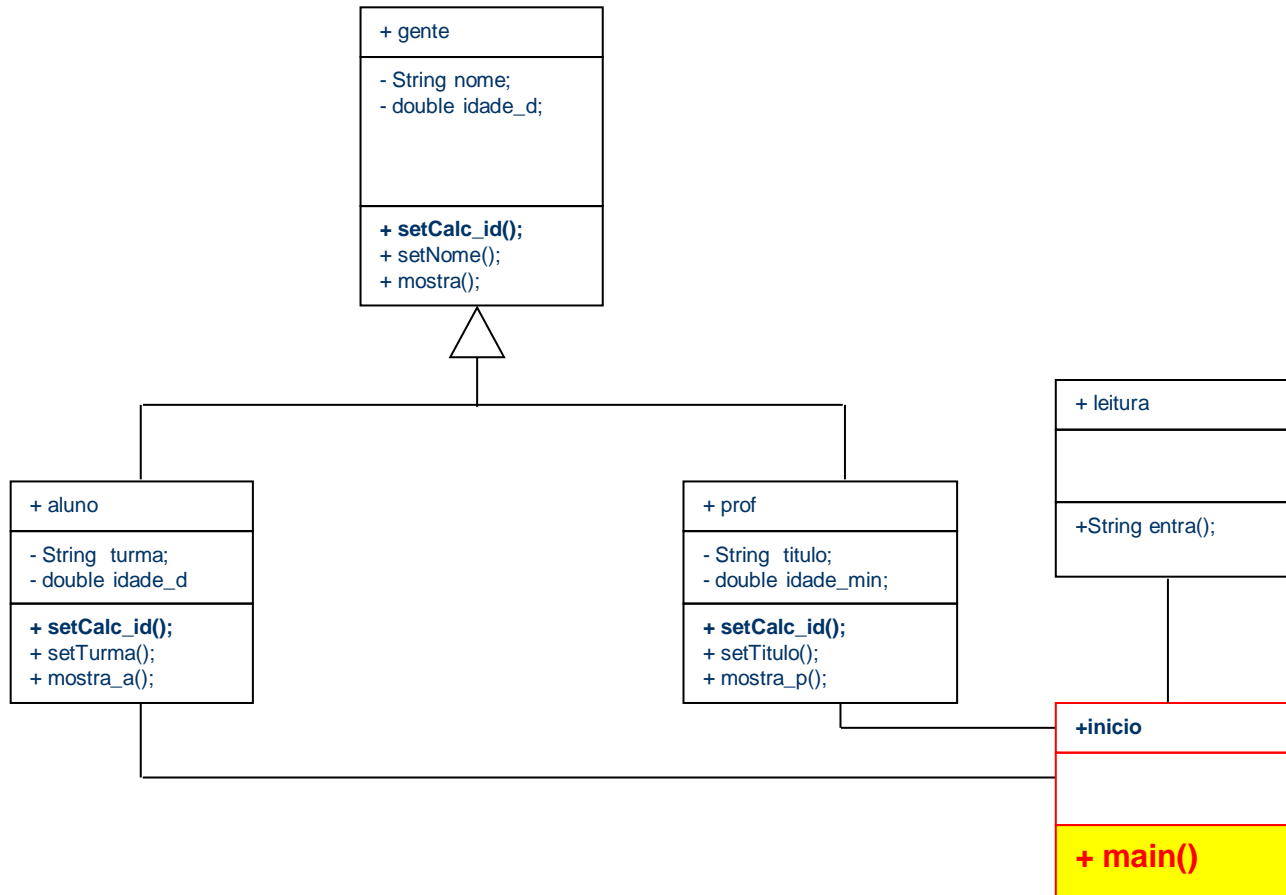
É a partir dessa linha de código, que o programa Java começa a ser executado. O interpretador Java carrega a classe e chama o método main(String args[]), ele tem o papel de ser o start-point (iniciar execução) de uma aplicação Stand-Alone, e quando a execução chegar ao fim do método main, a aplicação é encerrada.

POO – A linguagem de Programação Java

O método main():

Responsável pelo “start” do sistema.

Todas a classes **podem** ter o método main(), porém caso seja um conjunto de classes que fazem parte de um mesmo sistema, apenas uma será responsável pelo start do aplicativo.



POO – A linguagem de Programação Java

- Java reconhece letras maiúscula e minúscula (sensitive case)
- O nome do arquivo “.java” de ser o mesmo nome da classe (descrito internamente dentro do código)

POO – A linguagem de Programação Java - Prática

Saída de dados (na tela):

```
import java.io.*;
public class Saida {
    public static void main(String arg[]){
        System.out.println("Olá Mundo !");
    }
}
```

Exerc: Usando o modelo anterior:

- a.1)_Acrescente mais uma linha de exibição no programa anterior.
- a.2)_Crie um programa que imprima todos os seus dados cadastrais
- a.3)_Crie um programa que imprima a frase “Meu nome é: ” que através de parâmetros passados durante a chamada (interpretação: java ...) do programa, imprima na tela esta frase + o nome passado.

POO – A linguagem de Programação Java - Prática

Declaração de Variáveis e Operadores Matemáticos:

```
import java.io.*;
public class Operad {
    public static void main(String arg[]){
        System.out.println("Impressão de Resultado de Cálculos:");
        System.out.println(" ");
        int a=3,b=2;
        int c=a+b;
        System.out.println("O resultado da soma de A e B é: "+c);
    }
}
```

Exerc: Usando o modelo anterior:

b.1)_ Implemente as outras operações matemática “simples” a saber

Multiplicação (*), divisão (/) e subtração (-), sendo com primeiro termo da operação a variável “a” e o segundo termo a variável “b”, imprimindo seu resultado a cada operação.

POO – A linguagem de Programação Java - Prática

Laços (for):

```
Import java.io.*;
public class Laco1 {
    public static void main(String arg[]){
        int a=0;
        int b=10;
        for(a=0;a<=b;a++){
            System.out.println("O Valor de A dentro do FOR é: "+a);
        }
    }
}
```

Exerc: Usando o modelo anterior:

- c.1)_ Usando o laço “for”, construa um programa que ao invés de incrementar a variável “a” até que atinja o valor de da variável “b”, faça o inverso, decemente “b” até que acheque ao valor de “a”.

POO – A linguagem de Programação Java - Prática

Laços (while):

```
Import java.io.*;
public class Laco2 {
    public static void main(String arg[]){
        int a=0,
        b=15;
        while(a<=b){
            a++;
            System.out.println("O Valor de A dentro do while é: "+a);
        }
    }
}
```

Exerc: Usando o modelo anterior:

d.1)_ Da mesma forma que implementou o exercício c.1, implemente o d.1 usando a estrutura de repetição “while”

POO – A linguagem de Programação Java - Prática

Laços (do..while):

```
Import java.io.*;
public class Laco3 {
    public static void main(String arg[]){
        int a=0,
        b=15;
        do{
            System.out.println("O valor de A dentro do DO é: "+a);
            a++;
        }while(a<=b);
        System.out.println("");
    }
}
```

Exerc: Usando o modelo anterior:

e.1)_ Da mesma forma que implementou o exercício c.1, implemente o e.1 usando a estrutura de repetição “do..while”

POO – A linguagem de Programação Java - Prática

Seleção (if..elseif..else):

```
Import java.io.*;
public class Selec1 {
    public static void main(String arg[]){
        int a=0, b=20;
        for(a=0;a<=b;a++){
            if(a==(b/2)){
                System.out.println("O valor de A é igual a metade de B, logo A vale: "+a+" e B vale: "+b+");
            }
            else if(a!=(b/2)){
                System.out.println("O valor de A é diferente a metade de B, logo A vale: "+a+" e B vale: "+b);
            }
        }
    }
}
```

Exerc: Usando o modelo anterior:

f.1)_ Construa um programa que leia 2 valores inteiros e diga se o 1º é: maior, igual ou menor que o 2º valor.

POO – A linguagem de Programação Java - Prática

Seleção (ternário):

```
public class Selec2{  
    public static void main(String args[]){  
        int a=5;  
        int b=2;  
        int c = (a>b ? a : b);  
        System.out.println(c);  
        // ou → System.out.println(a>b ? a : b );  
    }  
}
```


POO – A linguagem de Programação Java - Prática

Seleção (switch..case):

```
Import java.io.*;
public class Selec3 {
    public static void main(String arg[]){
        int valor = (int)(Math.random()*5); //casting
        switch(valor){
            case 0:
                System.out.println("Primeira Opção (Valor igual a zero) "+valor);
                break;
            case 1:
                System.out.println("Segunda Opção (Valor igual a um) "+valor);
                break;
            default:
                System.out.println("Outras Opções (Valor maior que um) "+valor);
                break;
        }
    }
}
```

POO – A linguagem de Programação Java - Prática

Operadores Lógicos (&& e ||):

```
Import java.io.*;
public class Logic {
    public static void main(String arg[]){
        int a=0,c=0, valor=15;
        float b=15;
        for(a=0;a<=b;a++){
            valor--;
            c++;
            if(c<=(b/2)&& c<=valor){
                System.out.println("C eh MENOR que a metade de B 'E' MENOR IGUAL a VALOR");
                System.out.println("C vale.....: "+c);
                System.out.println("B / 2 vale.....: "+(b/2));
                System.out.println("VALOR vale...: "+valor);
            }else if (c>=(b/2)|| (c>=valor)){
                System.out.println("C eh MAIOR que a metade de B 'OU' MAIOR IGUAL a VALOR");
                System.out.println("C vale.....: "+c);
                System.out.println("B / 2 vale.....: "+(b/2));
                System.out.println("VALOR vale...: "+valor);
            }
        }
    }
}
```

POO – A linguagem de Programação Java - Prática

Entrada de dados via console:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class EntDados {
    public static void main(String arg[]){
        System.out.println("Entre com um valor: ");
        InputStreamReader c = new InputStreamReader(System.in);
        BufferedReader cd = new BufferedReader(c);
        String s = "";
        try{
            s = cd.readLine();
        }
        catch(IOException e){
            System.out.println("Erro de entrada");
        }
        System.out.println("O valor de entrada foi: "+s);
    }
}
```

POO – A linguagem de Programação Java - Prática

Entrada de dados via console (convertendo “*tipo*”):

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class EntDados {
    public static void main(String arg[]){
        System.out.println("Entre com um valor: ");
        InputStreamReader c = new InputStreamReader(System.in);
        BufferedReader cd = new BufferedReader(c);
        String s = "";
        try{
            s = cd.readLine();
        }
        catch(IOException e){
            System.out.println("Erro de entrada");
        }
        int w = Integer.parseInt(s);
        System.out.println("O valor de entrada foi: "+w);
    }
}
```

POO – A linguagem de Programação Java - Prática

Exercícios:

G1)_ Construa um programa que:

- a)_ Permita o usuário entrar com 2 valores (que sejam armazenados em duas variáveis, por exemplo, “a” e “b”);
- b)_ Apresente “menu de opções” com 4 opções:

Exemplo:

Menu de Opções:

- 1 – Somar (a+b)**
- 2 – multiplicar (a*b)**
- 3 - subtrair (a-b)**
- 5 – dividir (a/b)**

Escolha uma opção:

- c)_ Após o usuário escolher uma opção, apareça uma linha informando o resultado do cálculo que foi escolhido:

Exemplo (sendo escolhida a 1ª opção e “a” valendo 2 e “b” valendo 1):

“A soma de a + b é: 3”

Para tanto use as estruturas de entrada de dados via console, conversão de tipos e a estrutura de seleção switch..case.

POO – A linguagem de Programação Java - Prática

Alguns métodos da Classe Math.

```
public class Matem {
    public static void main(String arg[]){

        double f=2;
        double g=1.34;
        double h=9;

        System.out.println("\nO seno da var 'f' eh: " + Math.sin(f));
        System.out.println("\nO cosseno da var 'f' eh: " + Math.cos(f));
        System.out.println("\nA tangente da var 'f' eh: " + Math.tan(f));
        System.out.println("\nO valor mínimo entre as vars 'f' 'g' eh: " + Math.min(f,g));
        System.out.println("\nO valor máximo entre as vars 'f' 'g' eh: " + Math.max(f,g));
        System.out.println("\nO valor arredondado da var 'g'(1.434) eh: " + Math.round(g));
        System.out.println("\nO valor arredondado para cima da var 'g'(1.434) eh: " + Math.ceil(g));
        System.out.println("\nO valor arredondado para baixo da var 'g'(1.434) eh: " + Math.floor(g));

        System.out.println("\nA raiz da var 'h' eh: " + Math.sqrt(h));
        System.out.println("\nA var 'f' elevada a var 'h' eh: " + Math.pow(f,h));
        System.out.println("\nO valor absoluto da var 'g'(1.434) eh: " + Math.abs(g));
        System.out.println("\nO valor do 'PI' eh: " + Math.PI);
        System.out.println("\nGerar um valor aleatório ente 0 e 1: " + Math.random());

        System.out.println("\n\nA próxima linha não faz parte da classe Math, trata-se do operador 'Módulo'");
        System.out.println("\nO resto da divisão entre as vars 'h' e 'f' eh: "+ (f%h));
    }
}
```

POO – A linguagem de Programação Java - Prática

Execícios:

h1)_ Utilizando a Linguagem de programação Java, implemente um programa que permita a entrada de um valor inteiro e informe se este valor é “par” ou “ímpar”.

POO – A linguagem de Programação Java - Prática

Capturando a “data” do Sistema

```
import java.util.GregorianCalendar;
```

```
public class CaptData{
```

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

```
        GregorianCalendar data = new GregorianCalendar(); //trabalhando com datas
```

```
        int dia = data.get(data.DAY_OF_MONTH);
```

```
        int mes = data.get(data.MONTH)+1;
```

```
        int ano = data.get(data.YEAR);
```

```
        System.out.println("\nDATA ATUAL DO SISTEMA: "+dia+"/"+mes+"/"+ano+"\n\n");
```

```
        System.out.println("\nDATA ATUAL DO SISTEMA: "+data.get(data.DAY_OF_MONTH)+  
        "/" +data.get(data.MONTH)+ "/" +data.get(data.YEAR)+"\n\n");
```

```
    }
```

```
}
```


POO – A linguagem de Programação Java - Prática

Alguns métodos da Classe String.

```
public class Stringe {  
    public static void main(String arg[]){  
        String frase="Eis-me aqui SENHOR, envia-me a miM";  
        System.out.println("\t\t\t\t\t Eis-me aqui SENHOR, envia-me a mim");  
        System.out.println("\n\n A frase anterior tem "+frase.length()+" caracteres");  
        System.out.println("\n\n Transformando-a toda em MAIUSCULA:");  
        System.out.print(frase.toUpperCase());  
        System.out.println("\n\n TRANSFORMANDO-A TODA EM minuscula:");  
        System.out.print(frase.toLowerCase());  
        System.out.println("\n\n Procurando a letra da 10 posição:");  
        System.out.print(frase.charAt(13));  
        System.out.println("\n\n Retornando a substring dentro da frase:");  
        System.out.println(frase.substring(0,11));  
        System.out.println("\n\n 1ª Vez que encontrou a letra 'O' na frase:");  
        System.out.println(frase.indexOf('O'));  
        System.out.println("\n\n Tirando os espaços em branco:");  
        System.out.println(frase.trim());  
        System.out.println("\n\n Trocando as letras 'm' por 'M':");  
        System.out.println(frase.replace('m','M'));  
        System.out.println("\n\n Compara frases:");  
        System.out.println(frase.compareTo("M"));  
    }  
}
```

Array unidimensional (Vetor) – Entrada automática.

```
public class Vet1{
    public static void main(String arg[]){
        int larg=30;
        int vtr[]= new int[larg];
        System.out.println("O tamanho do vetor eh: "+vtr.length);
        int cont=0;
        for(int i=0;i<vtr.length;i++){
            cont=cont+2;
            vtr[i]=cont;
        }
        for(int i=0;i<vtr.length;i++){
            System.out.println("Vetor no local "+i+" tem valor "+vtr[i]);
        }
    }
}
```

POO – A linguagem de Programação Java - Prática

Array Bidimensional (Matriz) – Entrada automática.

```
public class Matriz {  
    public static void main(String arg[]){  
        int lin=5, col=5, cont=1;  
        int mtrz[][]=new int[lin][col];  
        System.out.println(mtrz.length);  
        for(int l=0;l<lin;l++){  
            for(int c=0; c<col;c++){  
                mtrz[l][c]=cont++;  
            }  
        }  
  
        for(int l=0;l<lin;l++){  
            for(int c=0; c<col;c++){  
                System.out.println("O valor da "+l+"ª linha e "+c  
coluna eh: "+mtrz[l][c]);  
            }  
        }  
    }  
}
```

POO – A linguagem de Programação Java - Prática

Exercício (sempre utilizando a linguagem Java):

Strings:

1)_ Construa um programa que permita ao usuário entrar com determinada frase, depois permita “escolher” uma letra qualquer e: caso a letra escolhida esteja na frase (seja maiúscula ou minúscula) diga quantas vezes ela apareceu e em que posição da frase. Senão, apareça uma frase informando que esta letra não existe na frase.

Matrizes:

1)_ Crie um programa, no qual terá um vetor de inteiros, cujo tamanho será definido pelo valor de uma variável local, que permita ao usuário entrar com os valores. Depois, estes valores serão apresentados na ordem inversa à da entrada.

2)_ Faça o mesmo procedimento do exercício anterior, porém desta vez estará usando uma matriz bidimensional