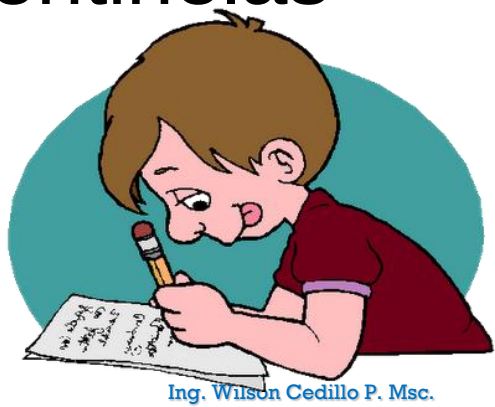
Primeros de Bachillerato
APLICACIONES INFORMÁTICAS

Fecha : 29 de mayo de 2016.
U.T. 5 : Estructuras de control y ampliación del lenguaje java.
Conceptos: Visibilidad y la clase Math.


```
static int ej1(int n1, int n2){
    int c=0;
    int s=0;
    double r;
    System.out.println("Los resultados de la serie son:");
    do{
        c+=1;
        r= Math.pow(n1, c);
        s+=r;
        System.out.println(n1+" ^ "+c+" = "+r);
    }while(c<n2);
    return s;
}//Fin ejercicio 1
```

RECORDANDO

- Acumuladores
- Centinelas
- Bucles controlados por centinelas



Encapsulamiento y Visibilidad

- Una de las actividades comunes en la programación orientada a objetos es el denominado **encapsulamiento** de las clases, esta actividad busca en forma general mostrar solo los aspectos necesarios, ocultando todos los que se consideran internos y que no deben ser mostrados.
- Para lograr el encapsulamiento se utiliza la **Visibilidad**.
- La **Visibilidad** establece que básicamente hay 3 modificadores de acceso a los atributos y métodos de nuestras clases, estas son: Publico, Protegido y Privado.

Visibilidad y los modificadores de acceso

Los modificadores de acceso preceden a la declaración de un elemento de la clase (ya sea atributo o método), de la siguiente forma:

[**modificador**] **tipo_variable** nombre;

[**modificador**] **tipo_devuelto** nombre_metodo (lista_Argumentos);

- ✓ **(-) private:** El campo o método sólo es visible dentro de la clase donde se define.
- ✓ **(+) public:** El campo o método es visible en cualquier clase.
- ✓ **(#) protected:** El campo o método es visible en la clase en donde se define y en cualquiera de sus subclases.
- ✓ **Ninguna de las anteriores**: El campo o método es visible en cualquiera de las clases pertenecientes al paquete en donde se define. se suele conocerse como default o package-private.

Visibilidad y los modificadores de acceso

Los distintos modificadores de acceso quedan resumidos en la siguiente tabla:

	LA MISMA CLASE	OTRA CLASE DEL MISMO PAQUETE	SUBCLASE DE OTRO PAQUETE	OTRA CLASE DE OTRO PAQUETE
public	X	X	X	X

Los modificadores de acceso en el diagrama de clases

Juan Armijos**:alumno**
(+)nombres: Juan Diego **(+) apellidos:** Armijos Goercke **(+) dirección:** Hermano Miguel 1226 **(+) teléfono:** 0987654321 **(+) Curso:** Primero "E2"
(-)cambiarDireccion() (+)cambiarTeléfono() (#)cambiaCurso()

El modificador static

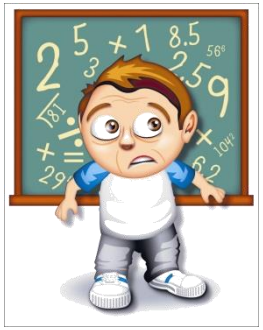
- ✓ Los **atributos miembros** de una clase pueden ser **atributos de clase** o **atributos de instancia**; se dice que son atributos de clase si se usa la palabra clave **static**.
- ✓ El modificador **static** sirve para crear miembros que pertenecen a la clase, y no a una instancia de la clase. Esto implica, entre otras cosas, que no es necesario instanciar la clase para poder acceder a estos atributos y métodos.

```
package Actividad12;
import java.util.Scanner;
public class Animal {
    private static String nombre;
    private static String color;
    p  static int patas;
    public static void main(String[] args) {
        String r;
        Scanner leer; //1er paso. DECLARAR UNA VARIABLE TIPO SCANNER
        leer = new Scanner(System.in);
        System.out.println("BIENVENIDOS AL ZOOLOGICO");
        System.out.print("INGRESE NOMBRE DEL ANIMAL: ");
```

Si el método ***main*** es static, no podrá utilizar métodos que no sean static

La clase Math

La clase java.lang.Math es una clase utilitaria cuyos métodos nos permiten realizar algunos cálculos matemáticos comunes.



Math es ***public*** para que se pueda llamar desde cualquier sitio y ***static*** para que no haya que inicializarla.

CONSTANTES MATEMÁTICAS

Math tiene una referencia a las dos constantes más utilizadas en matemática con una precisión de 15 decimales: la constante **Math.PI** con valor 3.14159265358979323846 y **Math.E**, la base de los logaritmos naturales, con valor 2.718281828459045354.

```
public class Aritmetica {
    public static void main(String[] args) {
        System.out.println("La constante Pi es --> " + Math.PI);
        System.out.println("La base logarítmica E es --> " + Math.E);
    }
}
```

```
run:
La constante Pi es --> 3.141592653589793
La base logarítmica E es --> 2.718281828459045
GENERACIÓN CORRECTA (total time: 1 second)
```

Métodos de la clase Math

Para conversiones

✓ **Método *Math.toRadians***

public static double ***toRadians***(double **angGr**)

Convierte la medida **angGr** de un ángulo de grados a radianes. Por ejemplo:

```
double angGrados = 45; //grados
double angRadianes = Math.toRadians(angGrados);
System.out.println("Grados a Radianes: " + angGrados + "º = " + angRadianes + " rad");
```

De la misma manera el método ***Math.toDegrees*** convierte de radianes a grados

Métodos de la clase Math

Funciones Trigonométricas

Las funciones trigonométricas aceptan y devuelven los ángulos en radianes por lo que siempre hay que convertir desde/hacia grados.

✓ **Método *Math.cos***

public static double ***cos***(double **a**)

Calcula el coseno de **a**. Por ejemplo para calcular el coseno de 30º:

```
double angRadianes = Math.toRadians(30);
double coseno = Math.cos(angRadianes);
System.out.println("El coseno de 30 grados es " + coseno);
```

De la misma manera los métodos ***Math.sin***, ***Math.tan***, ***Math.acos***, ***Math.asin***, ***Math.atan*** calculan el resto de funciones trigonométricas

Métodos de la clase Math

Funciones Matemáticas

✓ Método *Math.pow*

public static double ***pow***(double **a**, double **b**)

Devuelve el valor del primer argumento (**a**) elevado a la potencia el segundo argumento (**b**), es decir, a^b. Por ejemplo para calcular la potencia de 5^3:

```
double potencia = Math.pow(5,3);
System.out.println("La potencia de 5 elevado a la 2 es" + potencia);
```

✓ Método *Math.sqrt*

public static double ***sqrt***(double **a**)

Devuelve el resultado de calcular la raíz cuadrada de **a** (Número positivo).

Actividad en clases

Desarrollar un método en el que se ingrese un número **n** par mayor a 4, el método devolverá el resultado de acuerdo a la siguiente serie:

$$r = \frac{1}{\sqrt{n}} - \frac{1}{\sqrt{n-2}} + \frac{1}{\sqrt{n-4}} - \frac{1}{\sqrt{n-6}} + \frac{1}{\sqrt{n-8}} \dots\dots\dots \frac{1}{\sqrt{2}}$$

Por ejemplo si **n** es igual a 10 el resultado sería:

$$r = \frac{1}{\sqrt{10}} - \frac{1}{\sqrt{8}} + \frac{1}{\sqrt{6}} - \frac{1}{\sqrt{4}} + \frac{1}{\sqrt{2}}$$

Gracias !!!

Usted es libre de

: Copiar, distribuir y comunicar públicamente éstas diapositivas de resumen

Bajo las condiciones siguientes:

Reconocimiento: Debe reconocer los créditos del documento al autor

No comercial: No puede utilizar éste documento para fines comerciales

Sin obras derivadas: NO SE PUEDE ALTERAR, TRANSFORMAR O GENERAR UNA OBRA DERIVADA A PARTIR DE ÉSTE DOCUMENTO

Comentarios o sugerencias a los correos: ing.wilsoncedillo@msn.com ó wilsoncp@uets.edu.ec