



Programación III POO JAVA

Math Java.Lang
Operador Condicional ?:
Bucle for-each
ArrayList
Enumeraciones

Math Java.lang

- ▶ Clase Math de Java.lang. Sus métodos son estáticos, por lo que no es necesario crear un objeto de dicha clase. Su sintaxis general es:

`Math.metodo(argumentos);`

- ▶ `public static final double E = 2.7182818284590452354;`
 - ▶ `public static final double PI = 3.14159265358979323846;`
- ▶ `Math.min(x,y);`
- ▶ `Math.max(x,y);`
- ▶ `Math.random()*ValorLimite+ValorInicial`
- ▶ `Math.sqrt(x);` raíz cuadrada
- ▶ `Math.pow(base,exponente);` potencia
- ▶ `Math.pow (x, y)`

java.util.Random / java.lang

```
package bienvenida;  
  
import java.util.Random;  
  
public class NumRandom {  
    public static void metodos() {  
        Random rand = new Random();  
        int i = rand.nextInt(10)+1;  
        System.out.println(i);  
        int j = (int)Math.round((Math.random()*10 +1));  
        System.out.println(j);  
    }  
  
    public static void main(String []args){  
        metodos();  
    }  
}
```

t - Bienvenida (run) x

run:

3

4

BUILD SUCCESSFUL (total time: 0 seconds)

|

Operador Condicional ?:

```
String cadena="Saludo";  
System.out.println(cadena.equals("Saludo") ? "Las cadenas son iguales"  
: "Las cadenas no son iguales." );
```

```
int a=1, b=3, c, resul1, resul2;  
  
System.out.println( ( a<b ) ? ( c = a+b ) : ( c = b-a ) );  
resul1 = c;  
resul2 = ( a<b ) ? ( c = a+b ) : ( c = b-a );  
System.out.println("Resultado 1: " + resul1 + " Resultado 2: " + resul2);
```

Switch

```
public static void switchCadena(String cadena){  
    switch(cadena){  
        case "hola": System.out.println(cadena);break;  
        case "adios": System.out.println(cadena);break;  
        default: break;  
    }  
}
```

Bucle for-each

- Bucle for-each: se trata de un bucle diseñado con el propósito de recorrer un conjunto de objetos. Su sintaxis es:

```
for(Tipo elemento: colecciónElementos){  
    Grupo de sentencias;  
}
```

ArrayList

MÉTODO	DESCRIPCIÓN
size()	Devuelve el número de elementos (int)
add(X)	Añade el objeto X al final. Devuelve true.
add(posición, X)	Inserta el objeto X en la posición indicada.
get(posicion)	Devuelve el elemento que está en la posición indicada.
remove(posicion)	Elimina el elemento que se encuentra en la posición indicada. Devuelve el elemento eliminado.
remove(X)	Elimina la primera ocurrencia del objeto X. Devuelve true si el elemento está en la lista.
clear()	Elimina todos los elementos.
set(posición, X)	Sustituye el elemento que se encuentra en la posición indicada por el objeto X. Devuelve el elemento sustituido.
contains(X)	Comprueba si la colección contiene al objeto X. Devuelve true o false.
indexOf(X)	Devuelve la posición del objeto X. Si no existe devuelve -1

Enumeraciones

- ▶ Los tipos de datos enumerados son un tipo de dato definido por el programador. En su definición se debe indicar un conjunto de valores finitos sobre los cuales las variables/constantes de tipo enumeración deberán tomar valores.

Enumeraciones

Podemos considerar las enumeraciones como “constantes” al declarar solamente los métodos get, y definir estas constantes como finales.

```
package enumalumno;
public enum Alumno {
    A100("Juan", 34),
    A200("Guadalupe", 18),
    A300("Arturo", 38);

    private final String nombre;
    private final int edad;
    private Alumno(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }
    public String getNombre() {
        return nombre;
    }
    public int getEdad() {
        return edad;
    }
}
```

Enumeraciones

Podemos considerar las enumeraciones como “variables” al declarar los métodos set y get.

```
package enumalumno;
public enum Alumno {
    A100("Juan", 34),
    A200("Guadalupe", 18),
    A300("Arturo", 38);

    private String nombre;
    private int edad;
    private Alumno(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }
    public String getNombre() {
        return nombre;
    }
    public int getEdad() {
        return edad;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public void setEdad(int edad) {
        this.edad = edad;
    }
}
```

Enumeraciones

```
System.out.println(Alumno.A500.getNombre());
Alumno.A500.setNombre("Consuelo");
System.out.println(Alumno.A500.getNombre());

Alumno cons = Alumno.valueOf("A500");
System.out.printf("%-10s%-10s%5s\n",
    cons.toString(), cons.getNombre(), cons.getEdad());

for(Alumno alu: Alumno.values()){
    System.out.printf("%-15s%-30s%s\n",
        alu, alu.getNombre(), alu.getEdad() );
}
```

t - EnumAlumno (run) x

run:

Magda

Consuelo

A500	Consuelo	38
A100	Juan	34
A200	Guadalupe	18
A300	Arturo	38
A400	Miguel	18
A500	Consuelo	38
A600	Rosa	15
A700	Alejandro	14
A800	Victor	38

BUILD SUCCESSFUL (total time: 0 seconds)