Java

Constructor de objetos

Cuando se construye un objeto es necesario inicializar sus variables con valores coherentes, seleccionados después de realizar la abstracción. Los lenguajes orientados a objetos emplean los constructores. Un constructor es un método perteneciente a la clase que posee unas características especiales:

- Se llama igual que la clase.
- No devuelve nada, ni siquiera void.
- Pueden existir varios, pero siguiendo las reglas de la sobrecarga de funciones.
- De entre los que existan, tan sólo uno se ejecutará al crear un objeto de la clase.

Dentro del código de un constructor generalmente suele existir inicializaciones de variables y objetos, para conseguir que el objeto sea creado con dichos valores iniciales.

Para definir los constructores se emplea la siguiente sintaxis:

```
[modifVisibilidad] nombreConstructor (listaParámetros)
{
}
```

Para modifVisibilidad se aplica las mismas normas que para atributos y métodos:

- public: indica que es un método accesible desde afuera de una instancia del objeto.
- private: indica que a través de una instancia no es accesible el atributo. Al heredar el atributo se convierte en inaccesible.
- protected: indica que a través de una instancia cualquiera no es accesible el atributo. Al heredar si se puede usar desde la clase derivada o subclase.
- Sin especificar: indica visibilidad de paquete, se puede acceder a través de una instancia, pero sólo de clases que se encuentren en el mismo paquete.

nombreConstructor debe de coincidir con el nombre de la clase.

listaParámetros es la lista de los parámetros que tomará la función separados por comas y definidos cada uno de ellos como:

tipo nombreParámetro

El constructor posee un par de llaves, dentro de las cuales estará el código que se ejecutará al ser llamado.

Práctica:

- Vamos a agregar a la clase Persona un par de constructores, uno que la inic
  nombre un parámetro recibido en el constructor:

```
public Persona(){
        edad = 0;   //esto ocurre en forma automática
        nombre = null;  //esto ocurre en forma automática
}
public Persona(String nombre){
        edad = 0;  //esto ocurre en forma automática
        this.nombre = nombre;
}
public Persona(String nombre, int edad){
        this.edad = edad;
        this.nombre = nombre;
}
public Persona(Persona persona){
        this.edad = persona.edad;
        this.nombre = persona.nombre;
}
```

- y vamos a crear una clase ArranqueConstructor con el siguiente código:

```
public class ArranqueConstructor {
    public static void main (String arg[]){
        Persona per1 = new Persona();
        System.out.println( per1.nombre);
        System.out.println(per1.edad);
        Persona per2 = new Persona("Luis",13);
        System.out.println( per2.nombre);
        System.out.println(per2.edad);
        Persona per3 = new Persona(per1);
        System.out.println( per3.nombre);
        System.out.println(per3.edad);

    }
}
```