/TheVampi

Ensayo: Constructores

Programación Orientada a Objetos

Martinez Rivera Luis Fernando

“Nunca pares de aprender”

Introducción

En el video proporcionado por el profesor nos explica acerca del funcionamiento lógico e interno de los constructores en Java, específicamente se hace uso de ellos al crear objetos nuevos en una clase.

En este ensayo se explicará a manera profunda su funcionamiento, ya que superficialmente se puede creer que solo es seguir unas cuantas líneas de sintaxis, pero no, la realidad va más allá de eso, hay demasiada lógica interna involucrada en la creación de un objeto gracias a los constructores.

Desarrollo

Antes que nada, un pequeño recordatorio seria que una clase puede tener muchos objetos, pero un objeto solo puede pertenecer a una clase, esto es importante recordarlo ya que de aquí se desarrolla toda la explicación a continuación.

Al programar vamos a ocupar eventualmente crear o generar un objeto nuevo.

Existen dos maneras diferentes de crearlo.

- Opción 1 (manera no abreviada usando 2 líneas de código)

```
NombreClase nombreObjeto;
nombreObjeto = new NombreClase( [parámetros]);
```

- Opción 2 (manera abreviada usando una sola línea de código)

```
NombreClase nombreObjeto = new NombreClase( [parámetros]);
```

En cualquiera de las 2 opciones de creación de objetos, los parámetros son OPCIONALES, es decir, se puede crear un objeto sin parámetros, y también se puede crear un objeto especificando parámetros entre los paréntesis.

Ejemplificando…

Utilizaremos los fragmentos de código del video (créditos al profesor Benigno Molina) para explicar su creación con y sin parámetros. Así como los casos de errores donde no es valida la creación de un objeto.

```
public class NuevaClase
{
    public static void main(String [ ] args)
    {
        Alumno a1=new Alumno();
        Alumno a2=new Alumno(18,"Juan Pérez","17030025",96.74);
        Alumno a3=new Alumno(18,"Juan Pérez");
        Alumno a4=new Alumno("Juan Pérez","17030025",18,96.74);
        otras instrucciones
    }
}
```

```
public class Alumno
{
    public  int edad;
    String nombre, nroCtrl;
    double promedio;
     Alumno()
     {
     }
     Alumno(int e,String s, String n, doube p)
     {
          edad=e;
          nombre=s;
          nroCtrl=n;
          promedio=p;
     }
     ..... Aquí van los métodos restantes
}
```

Un dato muy importante por mencionar, es que los constructores en realidad son MÉTODOS de una clase como podemos observar en la imagen de la derecha.

Es decir, antes de usar un constructor para crear un objeto, ya debemos tener el método o constructor creado previamente en la clase donde queremos crear el objeto.

Aquí ya tenemos una clase con 2 constructores (métodos) creados, así como los atributos de clase e incluso un constructor que admite la entrada de parámetros:

```
public class Alumno
{
    public  int edad;
    String nombre, nroCtrl;
    double promedio;
     Alumno()
     {
     }
     Alumno(int e,String s, String n, doube p)
     {
          edad=e;
          nombre=s;
          nroCtrl=n;
          promedio=p;
     }
     ..... Aquí van los métodos restantes
}
```

Los constructores están nombrados con el mismo nombre de la clase.

En todos los ejemplos a explicar utilizaremos la creación abreviada de objetos con la sintaxis ya vista.

```
NombreClase nombreObjeto = new NombreClase( [parámetros]);
```

Creando el objeto a1

Aquí estamos creando un objeto sin parámetros.

```
Alumno a1=new Alumno();
```

Imagen 1

Este objeto efectivamente se puede crear, ya que SÍ hay un constructor llamado Alumno() en la clase (imagen 2) que en este caso no cuenta con parámetros Y TAMBIEN al UTILIZAR el constructor Alumno (en la imagen 1) tiene el mismo orden, sin parámetros.

```
Alumno()
{
}
```

Imagen 2

Esto al ejecutar crea un objeto con los atributos de clase previamente definidos, pero sin asignar ningún valor:

Creando el objeto a2

En este objeto a diferencia del anterior ya le estamos asignando 4 parámetros en orden cuando utilizamos el constructor.

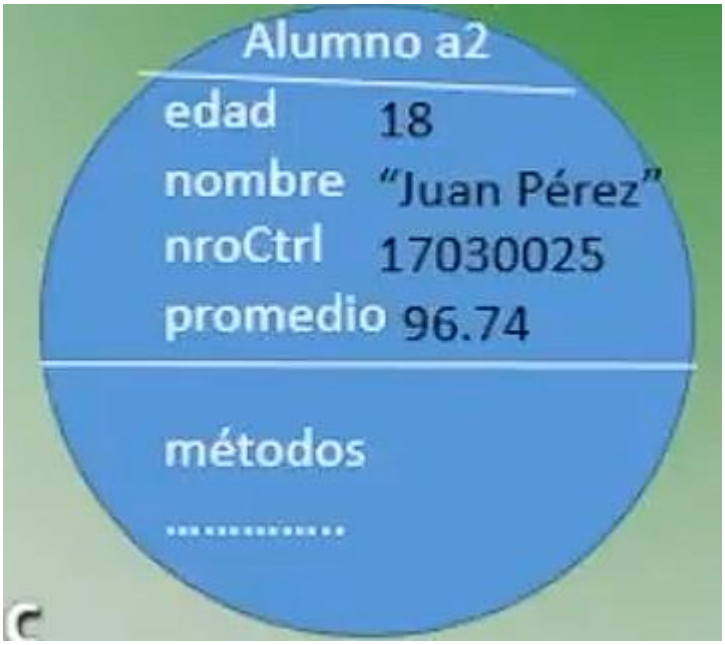
```
Alumno a2=new Alumno(18,"Juan Pérez","17030025",96.74);
```

Debemos comprobar en uno de los dos constructores previamente creados, que encajen estos atributos en orden. Es decir que vayan ordenados los tipos de datos como lo estamos requiriendo al construir el objeto (int, String, String, double) Estos se separan con comas al utilizar el constructor.

```
Alumno a2=new Alumno(18,"Juan Pérez","17030025",96.74);
```

```
Alumno(int e,String s, String n, doube p)
{
    edad=e;
    nombre=s;
    nroCtrl=n;
    promedio=p;
}
```

Al ejecutar esto, nos crea un objeto exitosamente, pero ahora asignando valores a sus atributos:



Creando el objeto a3

Este objeto lo estamos creando, pero a diferencia de los últimos, solo con 2 parámetros.

```
Alumno a3=new Alumno(18,"Juan Pérez");
```

Como podemos comprobar, no hay ningún constructor creado que se acople a los parámetros que estamos referenciando: en este caso requerimos un constructor que admita parámetros (int, String)

```
Alumno()
{
}
```

Este no nos sirve porque no admite parámetros.

```
Alumno(int e,String s, String n, doube p)
{
    edad=e;
    nombre=s;
    nroCtrl=n;
    promedio=p;
}
```

Este tampoco nos sirve porque a pesar de admitir parámetros los requiere en un orden (int, String, String, double). Y cuando usamos el constructor al crear el objeto estamos requiriendo (int, String) Cuando ponemos parámetros estos deben cumplirse todos de acuerdo con el tipo de dato y estar en orden.

Por lo tanto, marca error al ejecutar, NO SE PUEDE CREAR ESTE OBJETO.

Creando el objeto a4

Este objeto lo estamos creando, pero a diferencia de los últimos con 4 parámetros, y diferente orden.

```
Alumno a4=new Alumno("Juan Pérez","17030025",18,96.74);
```

Como podemos comprobar, no hay ningún constructor creado que se acople a los parámetros que estamos referenciando: en este caso requerimos un constructor que admita parámetros (String, String, int, double)

```
Alumno()
{
}
```

Este no nos sirve porque no admite parámetros.

```
Alumno(int e,String s, String n, doube p)
{
    edad=e;
    nombre=s;
    nroCtrl=n;
    promedio=p;
}
```

Este tampoco nos sirve porque a pesar de admitir parámetros los requiere en un orden (int, String, String, double). Y cuando usamos el constructor al crear el objeto estamos requiriendo (String, String, int, double) Cuando ponemos parámetros estos deben cumplirse todos de acuerdo con el tipo de dato y estar en orden.

Por lo tanto, marca error al ejecutar, NO SE PUEDE CREAR ESTE OBJETO.

Conclusión

Como pudimos observar crear un objeto no es solo seguir unas reglas de sintaxis, es seguir de acuerdo con una lógica más profunda y analítica.

En resumidas palabras: para crear un objeto requerimos de un constructor, y a su vez este constructor ya debe de estar definido y existente en la clase donde vamos a crear el objeto.

Al crear un objeto podemos incluir parámetros, si deseamos ponerlos deben de respetar el orden, deben cumplirse todos, y deben de estar de acorde a su tipo de dato. Todo esto último de acuerdo con el constructor ya creado en la clase.

Referencias:

Constructor5. (2017, 23 agosto). [Vídeo]. YouTube. Recuperado 11 de septiembre de 2022, de https://www.youtube.com/watch?v=0jy-DZEnuIw