APRENDERAPROGRAMAR.COM

EJERCICIO Y EJEMPLO RESUELTO: USO DE LA INTERFAZ CLONEABLE DE JAVA. MÉTODO CLONE() PARA CLONAR OBJETOS. (CU00912C)

Sección: Cursos
Categoría: Lenguaje de programación Java nivel avanzado I

Fecha revisión: 2039

Resumen: Entrega nº12 curso "Lenguaje de programación Java Nivel Avanzado I".

Autor: Manuel Sierra y José Luis Cuenca

EJERCICIO Y EJEMPLO RESUELTO: IMPLEMENTACIÓN DE LA INTERFAZ CLONEABLE

A continuación vamos a realizar un ejercicio acerca de cómo una clase debe hacer uso de la interfaz Cloneable del api de Java. Esto permite que en la susodicha clase se pueda realizar la copia de objetos tal cual, permitiendo hacer por tanto duplicados exactos, aunque eso sí, objetos al fin y al cabo distintos. Recuerda que esto no se puede hacer simplemente usando =. Al usar esta interfaz es como si al clonar un objeto tuviéramos un gemelo, son parecidos pero personas distintas.

En nuestro caso diríamos que son objetos gemelos, pero objetos distintos.

Vamos a partir del siguiente código sobre el cuál iremos haciendo las modificaciones necesarias:

```
/* Ejemplo Clase e Interfaz Cloneable aprenderaprogramar.com */

public class Persona {

  public int dni, edad;

    public Persona( int d, int e){
       this.dni = d;
       this.edad = e;
  }

}
```

Como vemos tenemos una clase Persona que nos viene identificada por los atributos dni y edad de tipo int. Dni representa un número de "documento nacional de identidad" y edad representa la edad en años. Ten en cuenta que dni se podría tratar como String, en este caso a efectos de este ejemplo no nos resulta relevante tratarlo de una forma u otra.

Toda clase sobre la que deseemos poder hacer una copia de objetos debe implementar la interfaz Cloneable. Esto nos obliga a escribir en cabecera de la clase que se implementa la interfaz y a definir el método clone(). La clase Persona nos quedará de la siguiente manera:

```
/* Ejemplo Clase e Interfaz Cloneable aprenderaprogramar.com */
public class Persona implements Cloneable{
  public int dni, edad;
    public Persona( int d, int e) {    this.dni = d;      this.edad = e;
  }

    public Persona clone()  {
    Persona clon = new Persona(this.dni,this.edad);
    return clon;
 }
}
```

Vemos los dos detalles a que nos referíamos: en cabecera de clase hemos escrito implements Cloneable, declaración por la que indicamos que esta clase implementa la interfaz y por tanto va a disponer de un método clone para clonar objetos.

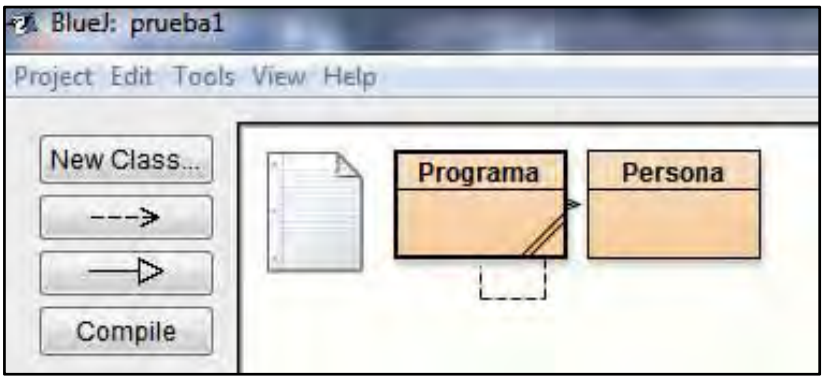
Por otro lado, definimos un método público (ha de ser obligatoriamente público puesto que es la sobreescritura de un método público) que devuelve un objeto de la clase Persona y que se invoca sin parámetros.

¿Qué código incluimos en el método? El código necesario para crear un nuevo objeto Persona con los mismos atributos que tenga el objeto sobre el que se invoque el método. El nuevo objeto lo hemos llamado clon (lo podríamos haber nombrado de cualquier manera como "duplicado", "gemelo", etc.). Y el nuevo objeto se genera invocando al constructor de clase y pasándole los parámetros necesarios para que sea un objeto igual a aquél sobre el que se invoca.

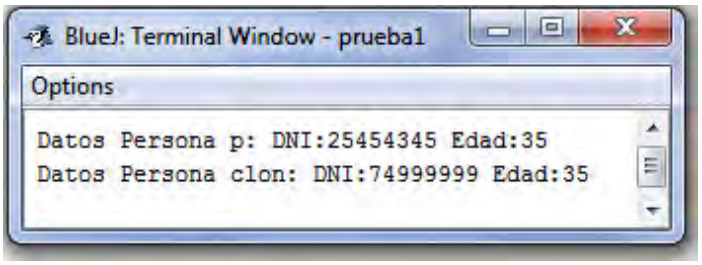
Una vez hemos definido el método, vamos a ver cómo hacer uso de él. Para ello vamos a crear la siguiente clase llamada Programa que va a contener nuestro método de ejecución public static void main (String arg[]) que como ya sabemos es el método por el que empieza a ejecutarse el código en Java.

```
/* Ejemplo Clase e Interfaz Cloneable aprenderaprogramar.com */
public class Programa {

   public static void main(String arg[]) {

      Persona p = new Persona(74999999,35);
      Persona p2 = p.clone();
       // hemos clonado en el objeto p2 los datos de la Persona p , por tanto p2
      // tiene como dni 74999999 y una edad de 35
      // a continuación vamos a cambiar el dni de p

      p.dni=25454345;

      System.out.println("Datos Persona p: DNI:"+p.dni+" Edad:"+p.edad);
      System.out.println("Datos Persona p2 clon: DNI:"+p2.dni+" Edad:"+p2.edad);

    }
}
```

Por tanto nuestro ejemplo tendrá 2 clases definidas en BlueJ de la siguiente manera:



La salida del programa es la siguiente, donde podemos ver como hemos clonado en el clon p2 todos los datos de la persona p previamente, y donde para distinguirlos hemos cambiado posteriormente el dni de p por 25454345 antes de imprimir sus datos.



A modo de conclusión, cuando queramos que los objetos de una clase puedan ser copiados debemos implementar la interfaz Cloneable y el método clone(). Para ello los pasos a seguir serán similares a lo que hemos visto aquí. Esto suele ser útil por ejemplo cuando tenemos una clase con muchos atributos o propiedades, y al trabajar con ella por ejemplo para calcular algún dato no queremos trabajar con ella directamente para no modificar el objeto original. Una opción es crear un objeto clonado y trabajar directamente sobre éste, sin afectar al original en ningún atributo. En realidad, la clonación de objetos será algo que nos puede resultar útil para muchas cosas.

Y por último, recordar que tratar de realizar una asignación de tipo p2 = p1 sería erróneo, ya que no generaríamos un clon sino dos variables que estarían apuntando al mismo objeto, no a distintos objetos. Si no tienes esto claro, repasa los contenidos del curso “Aprender programación Java desde cero” disponibles en la web aprenderaprogramar.com.

EJERCICIO

Define una clase que tenga cuatro atributos de tipo String cada uno de los cuales podrá ser exclusivamente una cadena de 12 ceros o de 12 unos. Es decir, cada atributo puede valer o bien "000000000000" ó bien "111111111111". La clase debe implementar la interface Cloneable y disponer de un método que permita ver el resultado de concatenar sus atributos.

Crea otra clase con el main donde crees un objeto cuyos atributos serán: "000000000000", "000000000000", "000000000000", "000000000000".

A partir de este primer objeto, crea cuatro copias usando el método clone() y modifícalas para llegar a tener cinco objetos (el inicial más los otros cuatro copias del original y luego modificados) cuyo contenido mostrarás por pantalla, debiendo obtener el siguiente resultado:

000000000000000000000000000000000000000000000000
111111111111000000000000000000000000000000000000
000000000000111111111111000000000000000000000000
000000000000000000000000111111111111000000000000
000000000000000000000000000000000000111111111111

Para comprobar si tu solución es correcta puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00913C

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=58&Itemid=180