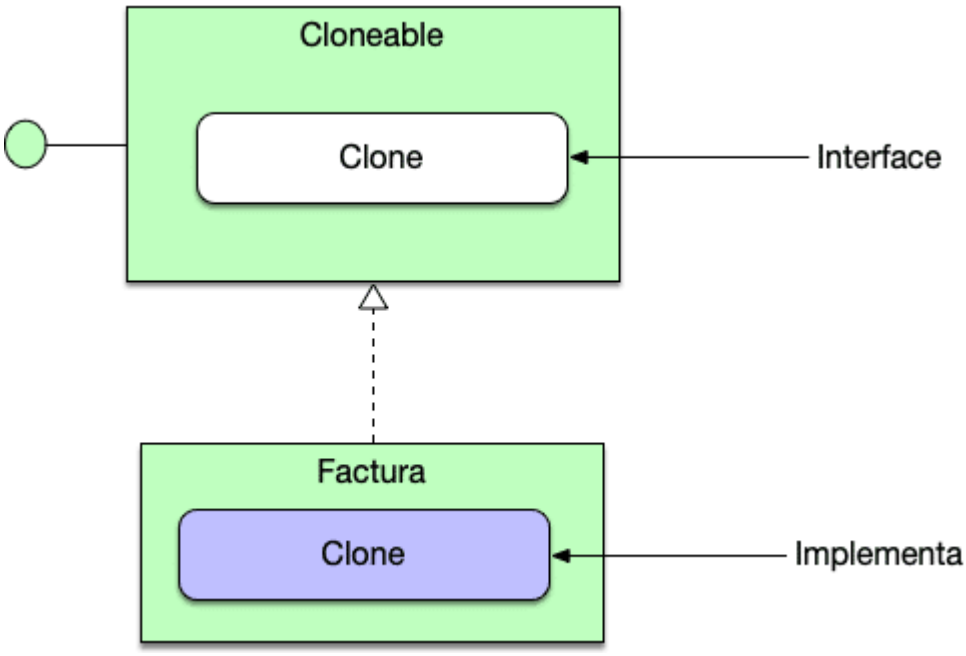
Tabla de Contenidos

- Java Clone
- Java Deep Clone con Cloning
  - Conclusiones
- Otros artículos relacionados

La operación de Java Clone es una de las operaciones más habituales cuanto trabajamos con el lenguaje siempre hay alguna situación en la que necesitamos clonar objetos . Vamos a ver un par de opciones habituales a la hora de trabajar en nuestro día a día . En primer lugar necesitamos construir una clase Factura. Esta clase necesitará para poder clonar objetos de ella implementar el interface Cloneable.



Este interface solo dispone de un método que es el método Clone que se encarga de implementar el sistema de clonación vamos a verlo en acción:

```
package com.arquitecturajava;
```

```
public class Factura implements Cloneable {

	private int numero;
	private String concepto;
	private double importe;
	public int getNumero() {
		return numero;
	}
	public void setNumero(int numero) {
		this.numero = numero;
	}
	public String getConcepto() {
		return concepto;
	}
	public void setConcepto(String concepto) {
		this.concepto = concepto;
	}
	public double getImporte() {
		return importe;
	}
	public void setImporte(double importe) {
		this.importe = importe;
	}
	public Factura(int numero, String concepto, double importe) {
		super();
		this.numero = numero;
		this.concepto = concepto;
		this.importe = importe;
	}
	@Override
	protected Object clone() throws CloneNotSupportedException {
```

```
                Factura nueva= new Factura
(this.numero,this.concepto,this.importe);
                return nueva;
        }
}
```

Java Clone

Es momento de una vez creada la clase construir el programa principal que se encarga de clonar la Factura y devolvernos una nueva copia .

```
package com.arquitecturajava;

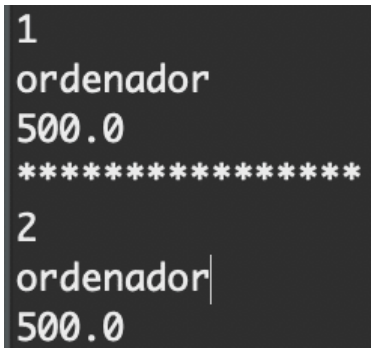
public class Principal {

        public static void main(String[] args) {

                Factura f= new Factura(1,"ordenador",500);
                try {
                        Factura nueva=(Factura)f.clone();
                        nueva.setNumero(2);
                        System.out.println(f.getNumero());
                        System.out.println(f.getConcepto());
                        System.out.println(f.getImporte());
                        System.out.println("****************");
                        System.out.println(nueva.getNumero());
                        System.out.println(nueva.getConcepto());
                        System.out.println(nueva.getImporte());
                } catch (CloneNotSupportedException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
```

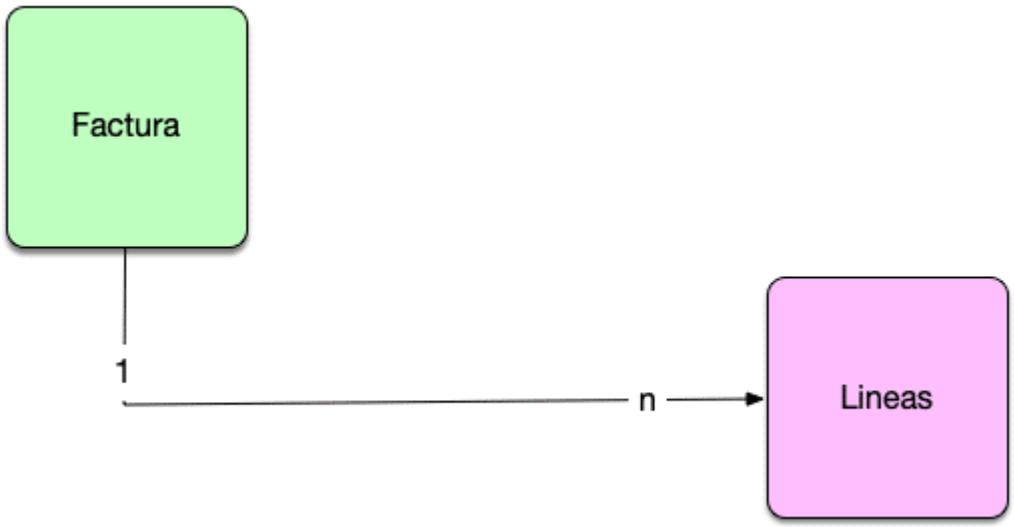
```
        }
}
```

Como vemos es tan sencillo como realizar un casting una vez invocado el método clone . Variamos alguno de los datos de la factura e imprimimos el resultado.



Java Deep Clone con Cloning

En muchas ocasiones las situaciones no son tan sencillas como la que acabamos de abordar y surge la necesidad de clonar estructuras de objetos más complejas como podría ser que tuvieras una Factura con sus lineas



La nueva estructura de clases quedaría así:

```
package com.arquitecturajava;
```

```
import java.util.ArrayList;
import java.util.List;

public class Factura implements Cloneable {

        private int numero;
        private String concepto;
        private double importe;
        private List<LineaFactura> lineas = new
ArrayList<LineaFactura>();
        public int getNumero() {
                return numero;
        }
        public void setNumero(int numero) {
                this.numero = numero;
        }
        public String getConcepto() {
                return concepto;
        }
        public void setConcepto(String concepto) {
                this.concepto = concepto;
        }
        public double getImporte() {
                return importe;
        }
        public void setImporte(double importe) {
                this.importe = importe;
        }
        public List<LineaFactura> getLineas() {
                return lineas;
        }
```

```
	public void setLineas(List<LineaFactura> lineas) {
		this.lineas = lineas;
	}
	public Factura(int numero, String concepto, double importe) {
		super();
		this.numero = numero;
		this.concepto = concepto;
		this.importe = importe;
	}
	@Override
	protected Object clone() throws CloneNotSupportedException {
		Factura nueva= new Factura
(this.numero,this.concepto,this.importe);
		return nueva;
	}
	public void addLinea(LineaFactura linea) {
		this.lineas.add(linea);
	}
}
```

```
package com.arquitecturajava;

public class LineaFactura {
	private int numero;
	private double importe;
	private String concepto;

	public int getNumero() {
		return numero;
	}
	public void setNumero(int numero) {
		this.numero = numero;
```

```
	}
	public double getImporte() {
		return importe;
	}
	public void setImporte(double importe) {
		this.importe = importe;
	}
	public String getConcepto() {
		return concepto;
	}
	public void setConcepto(String concepto) {
		this.concepto = concepto;
	}
	public LineaFactura(int numero, double importe, String
concepto) {
		super();
		this.numero = numero;
		this.importe = importe;
		this.concepto = concepto;
	}
}
```

En este caso para clonarlos vamos a usar Cloning una librería que permite clonar estructuras de objetos de forma sencilla

```
 package com.arquitecturajava;

import com.rits.cloning.Cloner;

public class Principal2 {

	public static void main(String[] args) {
```

```
                Factura f= new Factura(1,"ordenador",500);
                LineaFactura linea1 = new LineaFactura(1,200,"cpu");
                LineaFactura linea2 = new LineaFactura(12,300,"cpu");
                f.addLinea(linea1);
                f.addLinea(linea2);
                Cloner cloner=new Cloner();

                Factura copia=cloner.deepClone(f);
                System.out.println(copia.getNumero());
                System.out.println(copia.getConcepto());
                System.out.println(copia.getImporte());
                for (LineaFactura lf:f.getLineas()) {
                        System.out.println(lf.getNumero());
                        System.out.println(lf.getConcepto());
                        System.out.println(lf.getImporte());
                }
        }
}
```

Hemos clonado el objeto de forma completa con sus lineas.

```
1
ordenador
500.0
1
cpu
200.0
12
cpu
300.0
```

Conclusiones

El uso del interface Cloneable es una opción para situaciones sencillas . Ahora bien cuando necesitamos algo más potente las librerías siempre ayudan.

CURSO Java Herencia
GRATIS
APUNTATE!!

Otros artículos relacionados

- Java Interfaces y la simplicidad
- Java Herencia
- Java Polimorfismo