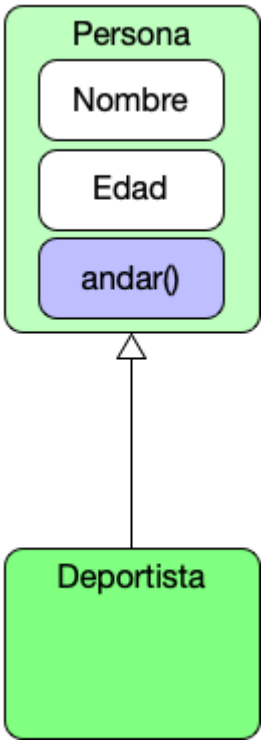
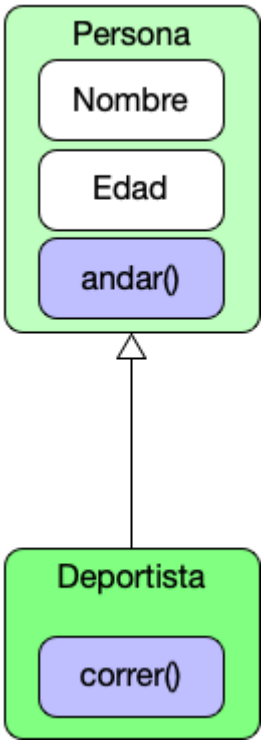
Tabla de Contenidos


- Java Herencia Multiple y Jerarquias de clase
- Java Herencia Multiple e interfaces
- Java Herencia Multiple con Java 8 Default Methods
- Conclusiones
- Otros artículos relacionados


¿Java Herencia Multiple? . Esta es una pregunta muy muy habitual en programación orientada a objeto . ¿Soporta Java Herencia Multiple? . La respuesta de cualquier desarrollador con experiencia es que no. Java no soporta la herencia multiple ya que una clase concreta solo puede tener una clase padre. Lamentablemente esta no es una respuesta totalmente correcta o final. Vamos a verlo más a detalle ya que se trata de un tema bastante interesante. Supongamos que tenemos la clase Persona con nombre y edad y el método andar .



De esta clase podemos heredar la clase de Deportista que añadirá el método correr y si nos apetece sobreescribirá el método andar para que el deportista ande más rápido hasta aquí todo es muy muy clásico.




Vamos a verlo en código:

```
package com.arquitecturajava;

public class Persona {

	private String nombre;
	private int edad;
	public String getNombre() {
		return nombre;
	}
	public void setNombre(String nombre) {
```

```
                this.nombre = nombre;
        }
        public int getEdad() {
                return edad;
        }
        public void setEdad(int edad) {
                this.edad = edad;
        }
        public void andar() {
                System.out.println("la persona " +nombre+ "anda");
        }
}
```

```
package com.arquitecturajava;

public class Deportista extends Persona {

        public void correr() {
                System.out.println("el deportista" + getNombre()+
"corre");
        }
}
```

# Java Herencia Multiple y Jerarquias de clase

Hasta aquí todo correcto ya que el deportista tiene el método andar y el método correr es cuestión de crear un método main y probarlos.

```
package com.arquitecturajava;

public class Principal {
```

```
        public static void main(String[] args) {
                Deportista d= new Deportista();
                d.setNombre("pedro");
                d.andar();
                d.correr();

        }

}
```

Al ejecutarlo vemos salir los mensajes por la consola:

```
la persona pedro anda
el deportista pedro corre
```

Acabamos de construir el ejemplo de herencia más clásico. Java no soporta a nivel de clases usar la palabra extends y aplicarla dos veces por ejemplo a la clase Deportista. Por lo tanto hasta aquí podemos decir que Java no soporta herencia múltiple. Sin embargo hay algunas cosas que revisar

# Java Herencia Multiple e interfaces

Muchas veces se nos olvida que Java si soporta herencia multiple a nivel de interfaces . Es decir un interface puede heredar de multiples interfaces . Vamos a verlo en código :

```
package com.arquitecturajava;

public interface Becable {

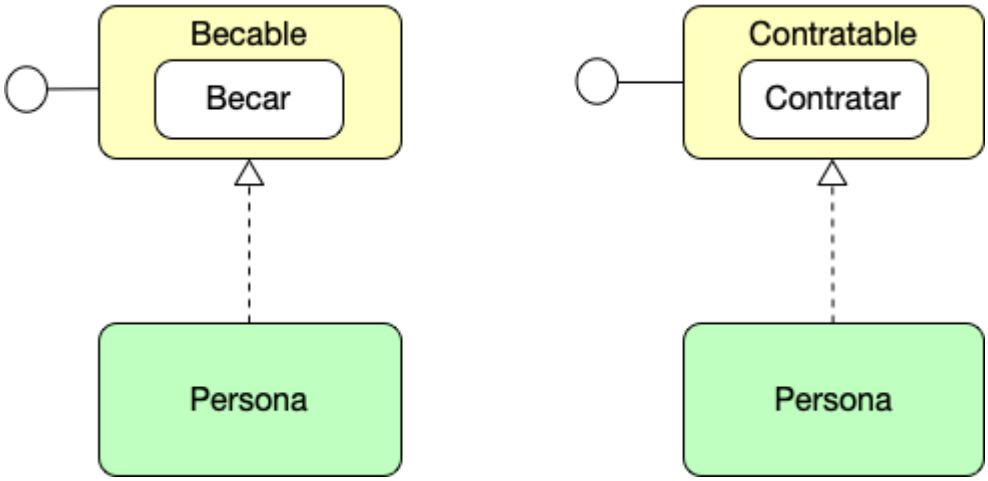
        public void becar() ;
}
```

```
package com.arquitecturajava;
```

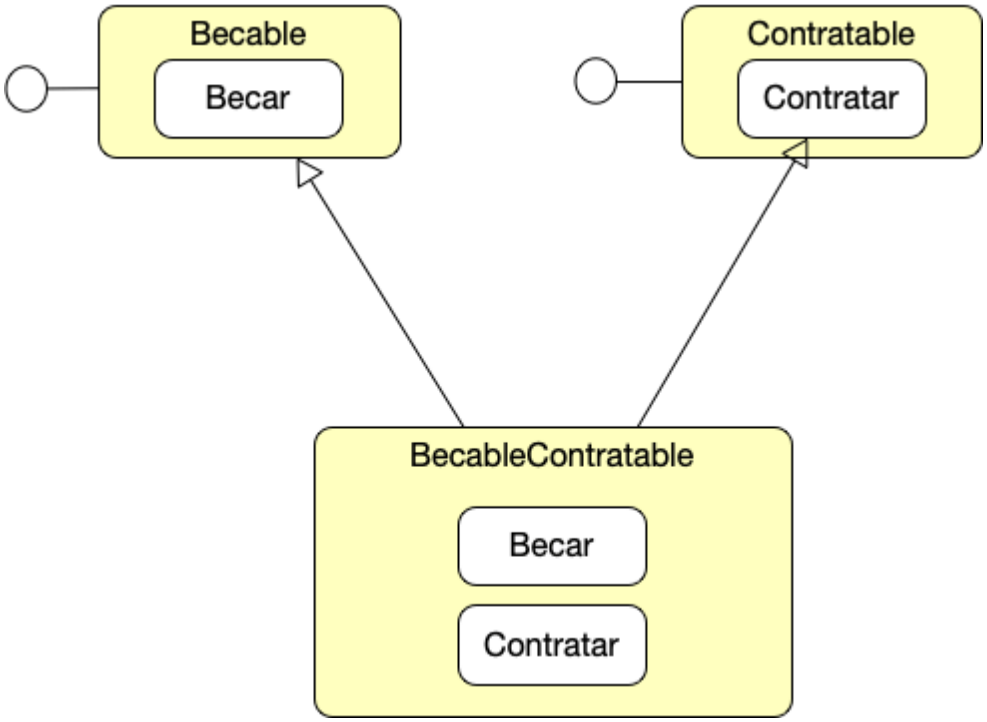
```
public interface Contratable {

        public void contratar();
}
```

En este caso tenemos dos interfaces Becable y Contratable que se pueden aplicar a una Persona ya que una persona se la puede contratar directamente o se le puede asignar una Beca para empezar a trabajar.




Hay gente que solo es Becable y hay gente que solo es Contratable . Unos porque les falta mucho para acabar la formación y solo pueden acceder a una Beca y otros porque han terminado ya sus estudios hace ya bastante tiempo. Ahora bien hay a veces gente que esta terminando los estudios y puede ser tanto Becable como Contratable en este caso el interface puede tener métodos de los dos.




Veamoslo en cóidigo:

```
package com.arquitecturajava;

public interface BecableContratable extends Contratable, Becable {

}
```

En este caso podemos diseñar la clase IngenieroJunior que extiende de Persona e implementa el interface BecableContratable.

```
package com.arquitecturajava;

public class IngenieroJunior extends Persona implements
BecableContratable {

        @Override
```

```
	public void contratar() {
		System.out.println("contratamos a la persona");

	}

	@Override
	public void becar() {
		System.out.println("becamos a la persona");

	}

}
```

# Java Herencia Multiple con Java 8 Default Methods

No solo eso sino que podríamos tener también a traves de Java 8 una implementación de estos métodos a nivel del propio interface de tal forma que la clase al implementar el interface incluya la implementación de estos métodos sin que tengamos que hacer nada nosotros a nivel de la clase . El código modificado sería:

```
package com.arquitecturajava;

public interface BecableContratable extends Contratable, Becable {
	@Override
	public  default void contratar() {
		System.out.println("contratamos a la persona");

	}

	@Override
```

```
        public  default void becar() {
                System.out.println("becamos a la persona");

        }


}
```

La clase IngenieroJunior implementaría el interface combinado y no le haría falta añadir ninguna implementación propia :

```
public class IngenieroJunior extends Persona implements
BecableContratable {


}
```

# Conclusiones

Acabamos de aplicar conceptos de herencia multiple en Java extendiendo interfaces e implementando Default methods. Estas son las posibilidades que de Java Herencia Multiple tanto a nivel de interfaces como de clases.

# Otros artículos relacionados

- Java Polimorfismo, Herencia y simplicidad
- Java Strategy Pattern herencia vs composición
- Default methods flexibilidad vs solidez
- Java Interface