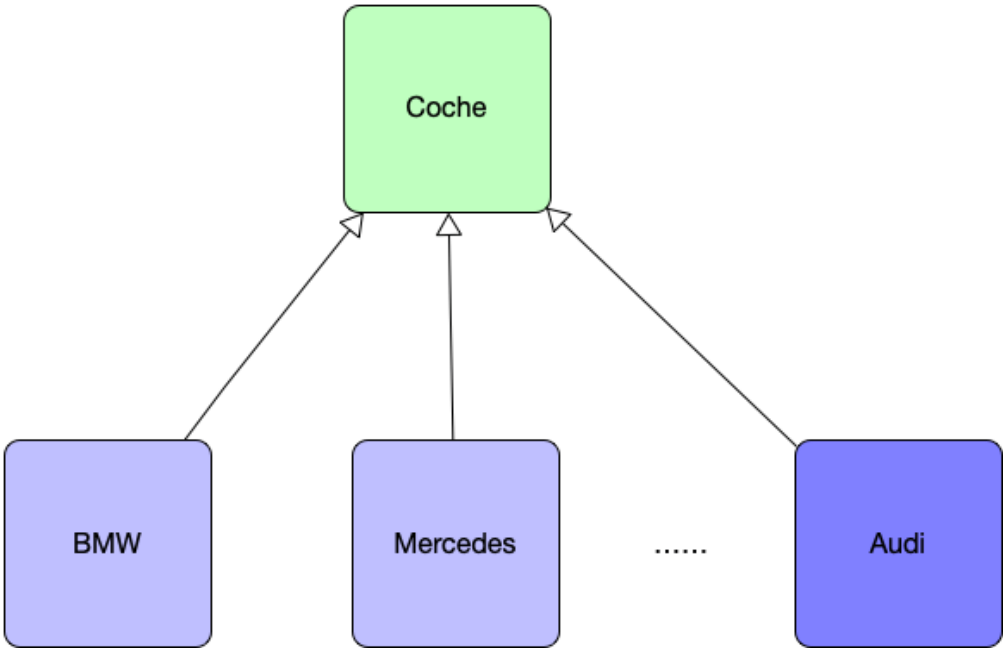
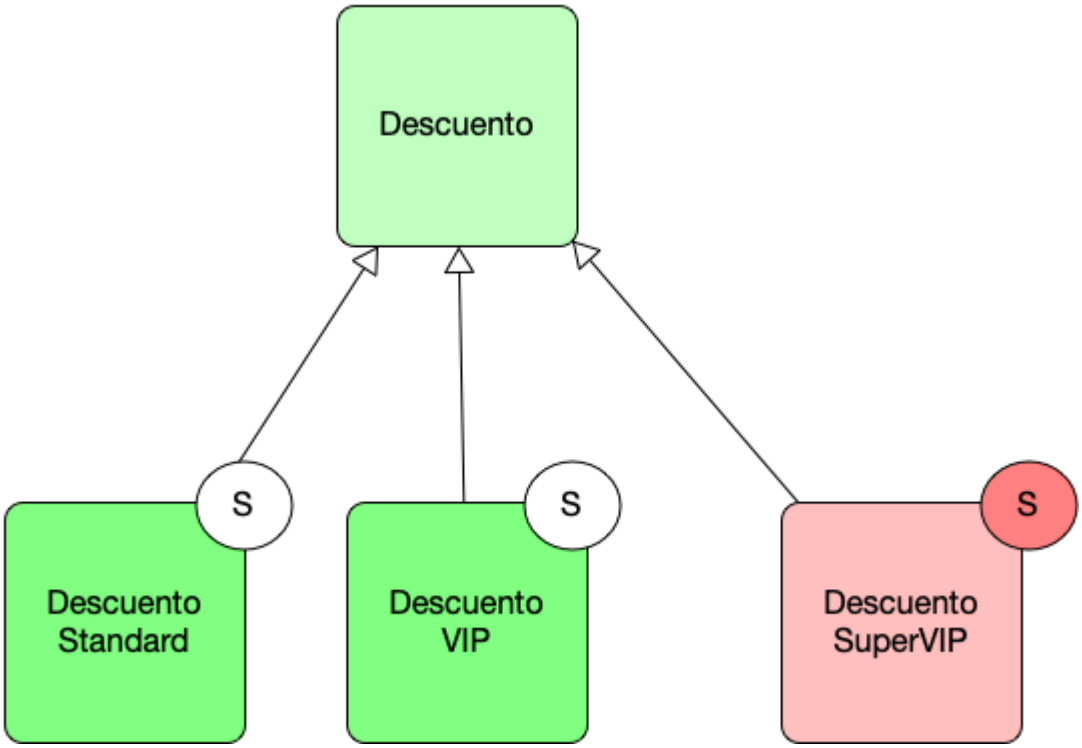
El concepto de Java Sealed Classes es un concepto relativamente nuevo y hace referencia a clases Selladas es decir a clases que nosotros permitimos su modificación de una forma muy concreta. Por ejemplo imaginemonos que tenemos una jerarquía de clases como Coche BWM , Mercedes . Se trata de una situación de herencia sencilla.



En este caso no hay mucho problema porque la jerarquía de clases puede ser muy muy grande y se admiten muchas marcas.

Java Sealed Classes

Ahora bien existen en otras condiciones en las cuales la jerarquia puede ser limitada este es el caso de por ejemplo la clase Descuento que puede tener dos o tres clases hijas especializadas pero no queremos que exista cualquier tipo de descuento en este caso la clase se puede marcar como clase sealed o sellada .



De esta manera admitimos el descuento Standard y el DescuentoVIP pero no admitimos el descuento SuperVIP ya que este no esta permitido al ser la clase sellada. Vamos a ver el código para entender el ejemplo de forma sencilla:

```
package com.arquitecturajava;


public sealed abstract class Descuento permits DescuentoStandard ,
DescuentoVIP {

        public abstract  double calcularDescuento(double importe);
}
```

```
package com.arquitecturajava;


public final   class DescuentoStandard extends Descuento {
```

```
	@Override
	public double calcularDescuento(double importe) {
		// TODO Auto-generated method stub
		return importe *0.90;
	}

}
```

```
package com.arquitecturajava;


public final  class DescuentoVIP extends Descuento {

	@Override
	public double calcularDescuento(double importe) {
		// TODO Auto-generated method stub
		return importe *0.80;
	}

}
```

De esta manera nosotros extendemos la clase principal de descuento generando dos clases hijas que han de ser finales y nos permiten personalizar el descuento aplicado. Ahora bien ¿Que pasaría si intentamos crear una nueva clase DescuentoSuperVIP?.

```
package com.arquitecturajava;


public final  class DescuentoSuperVIP extends Descuento {

	@Override
	public double calcularDescuento(double importe) {
```

```
                // TODO Auto-generated method stub
                return importe *0.70;
        }

}
```

Esta clase no podría compilar .¿Porque? muy sencillo porque hemos declarado su clase padre sellada. Al declarar la clase sellada solo aquellas que estén en el bloque de permits son admitidas como clases Hijas. Es así como funciona los mecanismos de Java Sealed Classes en JDK y su sellado. Estas capacidades se pueden usar también en interfaces.

Otros artículos relacionados

- Ofertas Cursos 2021 y COVID hasta 70%
- Navidad 2020 Curso React y ofertas hasta el 70%
- Cupón 60% de descuento en mis Cursos de Java Clásicos
- Navidad 2021 70 % de Descuento
- Black Friday 2021 75 % de Descuento