Examina la clase *Bicho* y su clase hija *BichoDormilon:*

```
class Bicho {
	public int hambre;
	private int peso;
	Bicho () { hambre =50; peso=50; }
	Bicho (int h, int p) { hambre =h; peso=p;}
	public void come () { hambre -=5; peso++; }
}

class BichoDormilon extends Bicho {
	public int sueño;
	BichoDormilon () { sueño = hambre * 2; }
	BichoDormilon (int i) { super(i,0); sueño = i+20; }

	@Override

	public void come () {hambre -=10; sueño +=5; }
	public void aDormir () {super.come(); sueño =0; }
}
```

a) ¿Qué mostraría el siguiente código?

```
Bicho bi = new Bicho();
BichoDormilon bd = new BichoDormilon();

System.out.println("bi -> hambre: " + bi.hambre);
```

bi → hambre: 50
coje el dato del hambre directamente sin problemas

```
System.out.println("bd -> hambre: " + bd.hambre);
```

bd → hambre: 50
coje el dato heredado del padre sin problemas

```
System.out.println("bd -> sueño: " + bd.sueño);
```

bd → sueño: 100
coje el dato de la clase hijo sin problemas

```
System.out.println("bd -> peso: " + bd.peso);
```

Error de compilación por que el atributo peso de la clase padre está privado.

b) Si a continuación se ejecutan estas instrucciones ¿qué mostraría?

```
bi.come();
bd.come();
como cada metodo come esta sobrescrito cada objeto esta ejecutando el suyo

System.out.println("bi -> hambre: " + bi.hambre); hambre 45

System.out.println("bd -> hambre: " + bd.hambre); hambre 40

bd.aDormir();

System.out.println("bd -> hambre: " + bd.hambre); hambre 35 ( con el dato del
hijo , internetmente , bd.aDormir llama al metodo del padre con lo cual se resta 5)

System.out.println("bd -> sueño: " + bd.sueño); sueño 0
```

c) Sobre las mismas clases, vamos a crear un nuevo *BichoDormilon* ¿qué mostrarían estas sentencias?

```
BichoDormilon bd2 = new BichoDormilon (5);
bd2.come(10); Error a come no se le pasa nada
System.out.println("bd2 -> hambre: " + bd2.hambre);
```

d) Vamos a hacer algo parecido otra vez, vamos a crear otro *BichoDormilon* ¿qué mostrarían estas sentencias?

```
BichoDormilon bd2 = new BichoDormilon (5);
bd2.come();
System.out.println("bd2 -> hambre: " + bd2.hambre); hambre -5
System.out.println("bd2 -> sueño: " + bd2.sueño); sueño 30
```
