

Ejercicio 2

Para el siguiente código, realice las siguientes tareas:

- (i) indique qué mal olor presenta
- (ii) indique el refactoring que lo corrige
- (iii) aplique el refactoring (modifique el código)

Si vuelve a encontrar un mal olor, retorne al paso (i).

```
class SuperMarioBros extends Game
```

```
{  
    public initialize()  
    { ... }  
    public startPlay()  
    { ... }  
    public endPlay()  
    { ... }  
    public play()  
    {  
        initialize();  
        startPlay();  
        endPlay();  
    }  
}
```

```
class DonkeyKong extends Game
```

```
{  
    public init()  
    { ... }  
    public startPlay()  
    { ... }  
    public endPlay()  
    { ... }  
    public jugar()  
    {  
        init();  
        startPlay();  
        endPlay();  
    }  
}
```

Ejercicio 2 - Refactoring

En forma iterativa, realice los siguientes pasos:

- (i) indique el mal olor,
 - (ii) indique el refactoring que lo corrige,
 - (iii) aplique el refactoring (modifique el código).
 - (iv) asegúrese de que los tests previos corran exitosamente.
- Si vuelve a encontrar un mal olor, retorne al paso (i).



```
public class Factura {
    private List<LineaDeFactura> items;

    public void imprimirDetalleConTotales() {

        String message = "Detalle de los items facturados";

        Iterator<LineaDeFactura> lineaDeFacturaIterator = items.iterator();
        while (lineaDeFacturaIterator.hasNext()) {
            LineaDeFactura lineaDeFactura = lineaDeFacturaIterator.next();
            message = message + String.format("Producto: %s",
                lineaDeFactura.getProducto().getNombre());
            + String.format(" Cantidad: %s", lineaDeFactura.getCantidad());
            + String.format("Precio unitario: %s", lineaDeFactura.getProducto().getPrecio());
            + String.format("Total por producto: %s", lineaDeFactura.getCantidad() *
                lineaDeFactura.getProducto().getPrecio());

            message = message + "Total sin impuestos";
            double totalSinImpuestos = items.stream().mapToDouble(lineaDeFactura ->
                lineaDeFactura.getProducto().getPrecio() * lineaDeFactura .getCantidad()).sum();
            message = message + String.format("Total: %s", totalSinImpuestos);

            message = message + "Total con IVA";
            message = message + String.format("Total: %s", totalSinImpuestos*1.21);

            System.out.println(message);
        }
    }

    public class LineaDeFactura {
        private Producto producto;
        private int cantidad;

        public Producto getProducto() {
            return this.producto;
        }

        public int getCantidad() {
            return this.cantidad;
        }
    }

    public class Producto {
        private String nombre;
        private double precio;

        public double getPrecio() {
            return this.precio;
        }

        public String getNombre() {
            return this.nombre;
        }
    }
}
```