

Ayudantía 3

Alexander Inostroza – alexander.inostroza@usm.cl

1. Catálogo de productos.

Se desea implementar un catálogo para una tienda que vende todo tipo de productos. Todo producto debe ser un artículo.

```
public interface Artículo {  
    public int getPrecio();  
    public int getStock();  
    public String getCategoria();  
    public String getTipo();  
    public String getNombre();  
}
```

Los productos estarán separados en categorías (muebles, computación, línea blanca, etc). Cada categoría se separará en tipos de productos (computación: notebooks, mouse, teclados, monitores, etc).

Por ejemplo, para dos audífonos, se modelaría la información:

Audifono1	Audifono 2
Precio: \$ 239.990	Precio: \$ 39.990
Stock: 34	Stock: 75
Categoría: Sonid	Categoría: Sonido
Tipo: Audífonos	Tipo: Audífonos
Nombre: Apple AirPods Pro 2da Generación	Nombre: Audífonos Bluetooth On-Ear 520BT Black

Implemente dos categorías y dos tipos para cada categoría. Luego cree algunos artículos de los tipos creados en un programa main para comprobar la visualización de los datos.

2. Equipamiento en un RPG.

Bugisoft está planeando lanzar un nuevo RPG, están en una fase muy temprana del desarrollo y le encargan a usted desarrollar la lógica para una primera versión del equipamiento y un personaje genérico que se pueda equiparse este equipamiento. Para ello le entregan a usted varios códigos sin terminar.

```
public class Character {
    private int healthPoints;
    private int defensePoints;
    private int evasionPoints;
    private int attackPoints;
    private int accuracyPoints;
    private Helm helm;

    public Character(){
        healthPoints=defensePoints=evasionPoints=attackPoints=accuracyPoints=0;
        helm = null;
    }
    public void setHelm(Helm helm){}
    public void getHelm(){ }
    public void equip(Equipment equipment){}
    public void unequip(Equipment equipment){}
}
```

```
public interface Equipment {
    public void equip(Character character);
    public void unequip(Character character);
    public int getDefensePoints();
    public int getEvasionPoints();
    public int getAttackPoints();
    public int getAccuracyPoints();
}
```

```
public abstract class Weapon implements Equipment{
    public int getAttackPoints(){return 0;}
    public int getAccuracyPoints(){return 0;}
    public abstract boolean isTwoHanded();
}
```

```
public abstract class DefensivePiece implements Equipment {  
    public int getDefensePoints(){return 0;};  
    public int getEvasionPoints(){return 0;};  
}
```

El personaje debe equiparse: un yelmo, una armadura, guantes, botas, arma en mano principal y arma en mano secundaria. Para ellos implemente las clases **Helm**, **Armor**, **Gloves**, **Boots**, **GreatSword** (de dos manos), **Bow** (de dos manos), **Sword** (mano principal), **Shield** (mano secundaria), **Dagger** (mano secundaria); utilizando herencia, polimorfismo y encapsulamiento.

Nota: Si el personaje intenta equiparse un yelmo, y ya tiene uno equipado, debe desequiparlo y descontar las estadísticas del anterior, antes de equipar el nuevo.