**class** Prenda {

**const** tipo; //basta con que sea un string, pues cada tipo de prenda no tiene comportamiento propio

**var** estadoPrenda; //puede ser instancia de EstadoNuevo, EstadoPromocion o EstadoLiquidacion

**var** precioPropio;

**method** precioFinal() {

**return** estadoPrenda.afectarPrecio(precioPropio);

}

}

**class** EstadoNuevo{

**method** afectarPrecio(unPrecio) {

**return** unPrecio;

}

}

**class** EstadoPromocion{

**var** valorPromocion;

**method** afectarPrecio(unPrecio) {

**return** unPrecio - valorPromocion;

}

}

**class** EstadoLiquidacion{

**method** afectarPrecio(unPrecio) {

**return** unPrecio / 2;

}

}

**class** Venta {

**const** prendasVendidas;

**const** cantidadPrendas;

**const** fecha;

**method** precioVenta();

**method** esDelDia(unaFecha) {

return unaFecha == fecha;

}

}

**class** VentaEfectivo **inherits** Venta {

**override method** precioVenta() {

**return** prendasVendidas.sum({prenda => prenda.precioFinal()});

}

}

**class** VentaTarjeta **inherits** Venta {

**const** cantidadCuotas;

**const** coeficienteFijo;

**method** valorRecargo() {

**return** cantidadCuotas \* coeficienteFijo + prendasVendidas.sum({prenda => prenda.precioFinal() \* 0.01})

}

**override method** precioVenta() {

**return** prendasVendidas.sum(prenda => prenda.precioFinal()) + **self.**valorRecargo();

}

}

**class** Macowins {

**var** ventas;

**var** prendas;

**method** gananciasDelDia(unaFecha) {

**var** ventasDelDia = ventas.filter({venta => venta.esDelDia(unaFecha)})

**return** ventasDelDia.sum({venta => venta.precioVenta()})

}

}