

```
/*
 * Material de apoyo Curso 2013 Juan Manuel Fernández
 * Proyecto Comercial
 * Paquete Tienda
 * Clase Brocha: representa una brocha específica
 *     Versión 1.0: Brocha individual que se vende por pieza, muy concreto
 */
package Tienda;

public class Brocha {
    private String nombre= "Brocha";
    private String material;
    private double ancho; //pulgadas
    private int cantidad;
    private String marca;
    private double precio;

    // constructor
    public Brocha(String mat, double aa, int k, String ma, double pp){
        material = mat;
        ancho = aa;
        cantidad = k;
        marca = ma;
        precio = pp;
    }

    //métodos de acceso

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getMaterial() {
        return material;
    }

    public void setMaterial(String material) {
        this.material = material;
    }

    public double getAncho() {
        return ancho;
    }
}
```

```

    public void setAncho(double ancho) {
        this.ancho = ancho;
    }

    public int getCantidad() {
        return cantidad;
    }

    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public double getPrecio() {
        return precio;
    }

    public void setPrecio(double precio) {
        this.precio = precio;
    }
}

```

```

/*
 * Material de apoyo Curso 2013 Juan Manuel Fernández
 * Proyecto Comercial
 * Paquete Tienda
 * Clase BrochaTest: representa una clase para probar objetos Brocha
 *     Versión 1.0: Prueba Brocha individual que se vende por pieza, muy concreto
 *
 */
package Tienda;

import static org.junit.Assert.assertEquals;

import org.junit.Before;
import org.junit.Test;

public class BrochaTest {
    //declaramos objetos para realizar pruebas
    private Brocha b1, brochita, brochota;

    @Before
    public void setUp() throws Exception {

```

```

// creamos los objetos para trabajar con ellos
b1 = new Brocha("pelo camello",2.0,10,"Pintafácil",11.50);
brochita = new Brocha("vinil",0.5, 22,"Colorex", 6.80);
brochota = new Brocha("pelo equino", 11.0,7,"Colorex", 23.00);
}

//Realizamos pruebas
@Test
public void testUno(){
    System.out.println("Características de b1: ancho= "+b1.getAncho()+
        " material= "+b1.getMaterial()+" marca= "+b1.getMarca()+
        " precio= "+b1.getPrecio());
    assertEquals(b1.getCantidad(),10);
}

@Test
public void testDos(){
    System.out.println("Características de brochita: ancho= "+b1.getAncho()+
        " material= "+brochita.getMaterial()+" marca= "+brochita.getMarca()+
        " cantidad= "+brochita.getCantidad());
    assertEquals(b1.getPrecio(),6.8,0.01); //el último parámetro es la tolerancia
}
}
}

```

Procedimiento para crear clase Brocha:

1. Abrir el paquete Eclipse indicando un directorio al que ustedes tengan derecho. No lo marque para recordarlo.
2. Crear un proyecto Java y darle el nombre que quieran. Yo lo llamé "Comercial"
3. Den dos clics en el nombre del proyecto.
4. Con botón derecho sobre el proyecto, con menú File crear nuevo Paquete y darle el nombre "Tienda"
5. Con File crear nueva Clase (o usar icono) y llamarla Brocha (cuide mayúsculas y minúsculas)
6. Copie todo excepto los métodos llamados getX o setX
7. Cuando termine, posicione el cursor en la penúltima línea (antes de la llave que cierra (}); oprima botón derecha, seleccione "Source" y luego "Generate getters and setters" y marque todas las casillas verificadoras.

Procedimiento para Prueba

1. Teniendo abierta la clase Brocha, usando menú File seleccione Save.
2. Con el icono de nueva clase, elegir JUnit test Case; verifique que aparezca el nombre "BrochaTest" y seleccione junit 4.. Marque setUp. De Next
3. Marque las pruebas deseadas. Por el momento marque dos getX.
4. Si aparece una ventana preguntando si anota junit en el path diga que sí.
5. Modifique los métodos "testwww" según el ejemplo anterior. Observe que los métodos traen una función "fail", que debe ser eliminada y sustituida por un assertEquals u otro assert.