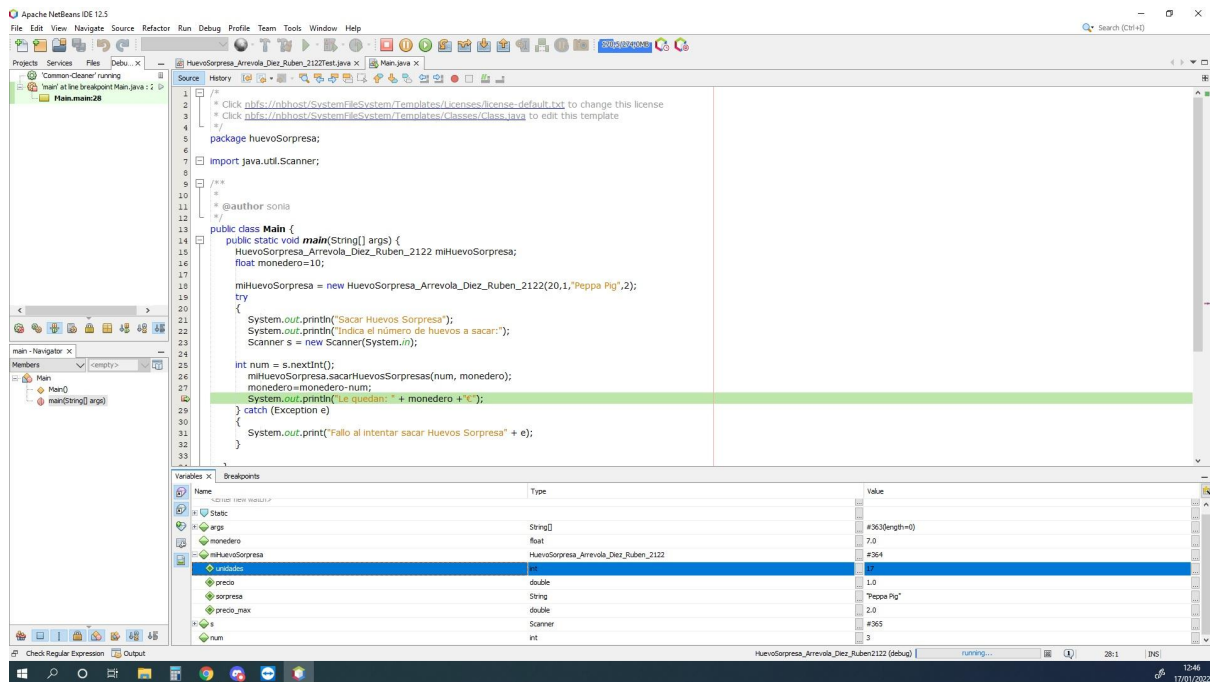


Tarea 3.- Entornos de Desarrollo

Autor: Rubén Arrevola Diez.

Apartado 2.- Captura de pantalla de variable “unidades” en punto de ruptura en la línea 28



Apartado 3.- Casos de prueba con valores válidos.

```
/**
```

- * Test sacarHuevosSorpresas con valores válidos. Intentamos sacar 3 huevos
- * con 20€ de dinero. Al ser el valor de los huevos 2€ debería funcionar
- * bien y dejar el número de unidades en 17.

```
*/
```

```
@Test
```

```
public void testSacarHuevosSorpresasValidoV1() throws Exception {
    System.out.println("Test SacarHuevoSorpresas con valores válidos.");
    int unidades = 3;
    double dinero = 20;
    try {
        HuevoSorpresas_Arrevola_Diez_Ruben_2122 miHuevo = new
        HuevoSorpresas_Arrevola_Diez_Ruben_2122(20, 2, "SORPRESA", 3);
        miHuevo.sacarHuevosSorpresas(unidades, dinero);
        assertEquals(17, miHuevo.obtenerUnidades());
    } catch (Exception e) {
        /* no debería saltar ninguna excepción en este caso,
        por lo que sí lo hace es porque algo no está bien y el test debería fallar */
    }
}
```

```

        fail("Se ha producido una excepción no esperada: " + e);
    }

}

/**
 * Test sacarHuevosSorpresas con valores válidos. Intentamos sacar 3 huevos
 * con 6€ de dinero. Al ser el valor de los huevos 2€ debería funcionar bien
 * y dejar el número de unidades en 17.
 */
@Test
public void testSacarHuevosSorpresasValidoV2() throws Exception {
    System.out.println("Test SacarHuevoSorpresa con valores válidos.");
    int unidades = 3;
    double dinero = 6;
    try {
        HuevoSorpresa_Arrevola_Diez_Ruben_2122 miHuevo = new
HuevoSorpresa_Arrevola_Diez_Ruben_2122(20, 2, "SORPRESA", 3);
        miHuevo.sacarHuevosSorpresas(unidades, dinero);
        assertEquals(17, miHuevo.obtenerUnidades());
    } catch (Exception e) {
        /* no debería saltar ninguna excepción en este caso,
        por lo que sí lo hace es porque algo no está bien y el test debería fallar */
        fail("Se ha producido una excepción no esperada: " + e);
    }

}

/**
 * Test sacarHuevosSorpresas con valores válidos. Intentamos sacar 20 huevos
 * con 50€ de dinero. Al ser el valor de los huevos 2€ debería funcionar
 * bien y dejar el número de unidades en 0.
 */
@Test
public void testSacarHuevosSorpresasValidoV3() throws Exception {
    System.out.println("Test SacarHuevoSorpresa con valores válidos.");
    int unidades = 20;
    double dinero = 50;
    try {
        HuevoSorpresa_Arrevola_Diez_Ruben_2122 miHuevo = new
HuevoSorpresa_Arrevola_Diez_Ruben_2122(20, 2, "SORPRESA", 3);
        miHuevo.sacarHuevosSorpresas(unidades, dinero);
        assertEquals(0, miHuevo.obtenerUnidades());
    } catch (Exception e) {
        /* no debería saltar ninguna excepción en este caso,
        por lo que sí lo hace es porque algo no está bien y el test debería fallar */
        fail("Se ha producido una excepción no esperada: " + e);
    }

}

```

```
}
```

Apartado 4.- Casos de prueba con valores inválidos.

```
/**
 * Test sacarHuevosSorpresas con valores inválidos. Dinero introducido
 * negativo o 0. Intentamos sacar 3 huevos con -20€ de dinero. Debe saltar
 * la excepción. Con esta prueba comprobamos que el método
 * sacarHuevoSorpresa no acepta números negativos ni 0.
 */
@Test
public void testSacarHuevosSorpresasDineroNegativo() throws Exception {
    System.out.println("Test SacarHuevoSorpresa con dinero negativo.");
    int unidades = 3;
    double dinero = -20;
    HuevoSorpresa_Arrevola_Diez_Ruben_2122 miHuevo = new
HuevoSorpresa_Arrevola_Diez_Ruben_2122(20, 2, "SORPRESA", 3);
    try {
        miHuevo.sacarHuevosSorpresas(unidades, dinero);
        fail("Intento de sacar huevo sorpresa con dinero negativo");
    } catch (Exception e) {
        System.out.println(e);
        assertEquals(20, miHuevo.obtenerUnidades());
    }
}

/**
 * Test sacarHuevosSorpresas con valores inválidos. Dinero introducido
 * Insuficiente. Intentamos sacar 3 huevos con 2€ de dinero. Debe saltar la
 * excepción. Con esta prueba comprobamos que el método sacarHuevoSorpresa
 * no acepta comprar con menos dinero del necesario.
 */
@Test
public void testSacarHuevosSorpresasDineroInsuficiente() throws Exception {
    System.out.println("Test SacarHuevoSorpresa con dinero insuficiente.");
    int unidades = 3;
    double dinero = 2;
    HuevoSorpresa_Arrevola_Diez_Ruben_2122 miHuevo = new
HuevoSorpresa_Arrevola_Diez_Ruben_2122(20, 2, "SORPRESA", 3);
    try {
        miHuevo.sacarHuevosSorpresas(unidades, dinero);
        fail("Intento de sacar huevo sorpresa con dinero insuficiente.");
    } catch (Exception e) {
        System.out.println(e);
        assertEquals(20, miHuevo.obtenerUnidades());
    }
}
```

```
}
```

```
/**
```

```
* Test sacarHuevosSorpresas con valores inválidos. Número de Huevos  
* negativo o 0. Intentamos sacar -3 huevos con 20€ de dinero. Debe saltar  
* la excepción. Con esta prueba comprobamos que el método  
* sacarHuevoSorpresa no acepta comprar un número negativo de huevos ni 0.  
*/
```

```
@Test
```

```
public void testSacarHuevosSorpresasHuevosNegativos() throws Exception {  
    System.out.println("Test SacarHuevoSorpresa con número de huevos negativo.");  
    int unidades = -3;  
    double dinero = 20;  
    HuevoSorpresa_Arrevola_Diez_Ruben_2122 miHuevo = new  
HuevoSorpresa_Arrevola_Diez_Ruben_2122(20, 2, "SORPRESA", 3);  
    try {  
        miHuevo.sacarHuevosSorpresas(unidades, dinero);  
        fail("Intento de sacar un número negativo de huevo sorpresa.");  
    } catch (Exception e) {  
        System.out.println(e);  
        assertEquals(20, miHuevo.obtenerUnidades());  
    }  
}
```

```
}
```

```
/**
```

```
* Test sacarHuevosSorpresas con valores invalidos. Numero de Huevos  
* superior al número de huevos guardados. Intentamos sacar 30 huevos con 100€ de  
dinero, cuando solo hay 20 huevois en stock.  
* Debe saltar la excepción. Con esta prueba comprobamos que el método  
* sacarHuevoSorpresa no acepta comprar un número mayor de huevos que hay en  
stock.
```

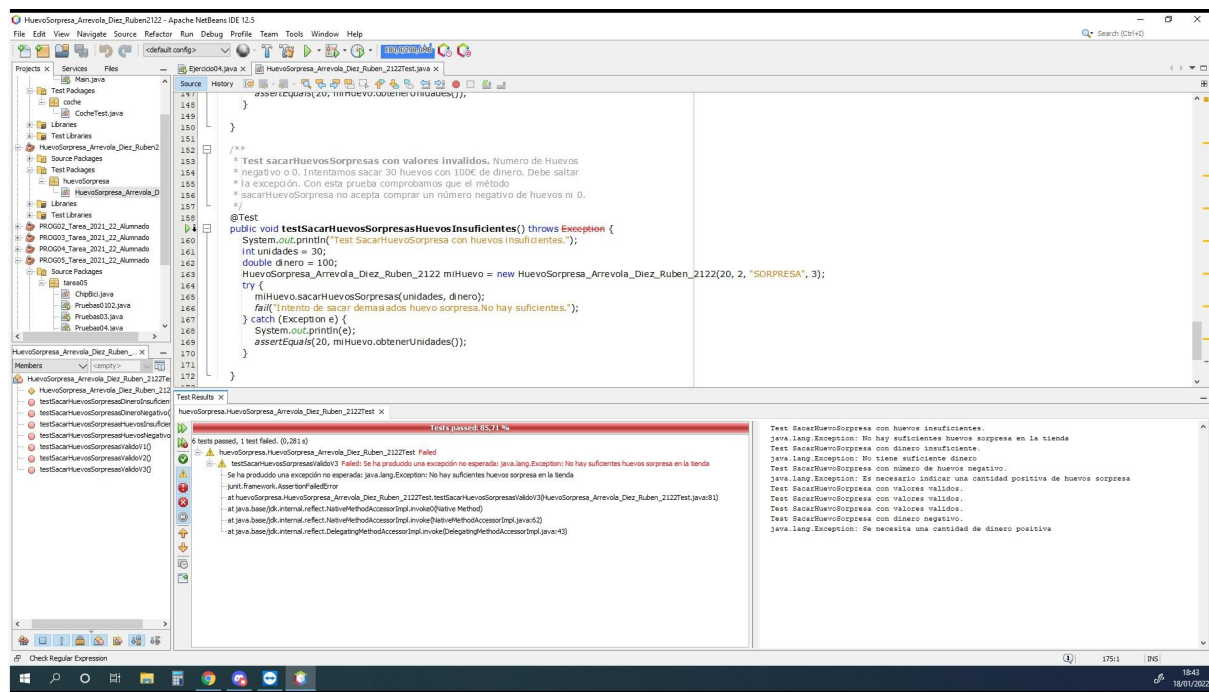
```
*/
```

```
@Test
```

```
public void testSacarHuevosSorpresasHuevosInsuficientes() throws Exception {  
    System.out.println("Test SacarHuevoSorpresa con huevos insuficientes.");  
    int unidades = 30;  
    double dinero = 100;  
    HuevoSorpresa_Arrevola_Diez_Ruben_2122 miHuevo = new  
HuevoSorpresa_Arrevola_Diez_Ruben_2122(20, 2, "SORPRESA", 3);  
    try {  
        miHuevo.sacarHuevosSorpresas(unidades, dinero);  
        fail("Intento de sacar demasiados huevo sorpresa.No hay suficientes.");  
    } catch (Exception e) {  
        System.out.println(e);  
        assertEquals(20, miHuevo.obtenerUnidades());  
    }  
}
```

```
}
}
```

Apartado 5.- Captura de ejecución de pruebas.



Todas las pruebas de ejecución de valores inválidos son correctas, lo que quiere decir que el método sacarHuevoSorpresa sólo admite:

- Dinero introducido mayor que 0.
- Que el dinero introducido sea mayor que la cantidad de huevos a sacar por el precio de los huevos.
- Que el número de huevos a sacar sea mayor que 0.
- Que el número de huevos a sacar sea mayor que el número de huevos de la tienda.

La prueba con valores correctos testSacarHuevosSorpresasValidoV3() intenta comprar 20 huevos con 50€. La cantidad de huevos en la tienda es de 20 huevos y el precio de cada huevo es de 2€ por lo que la compra debería ser correcta dejando un stock de 0 huevos en la tienda.

Esta prueba da error por lo que tenemos dos opciones:

- El programa no admite comprar todos los huevos de la tienda y dejar el stock a 0 (no lo especifica en la documentación).
- El programa tiene un fallo y lo hemos detectado con esta prueba.