Tarea -Testing

Nombre: Gerardo Torres
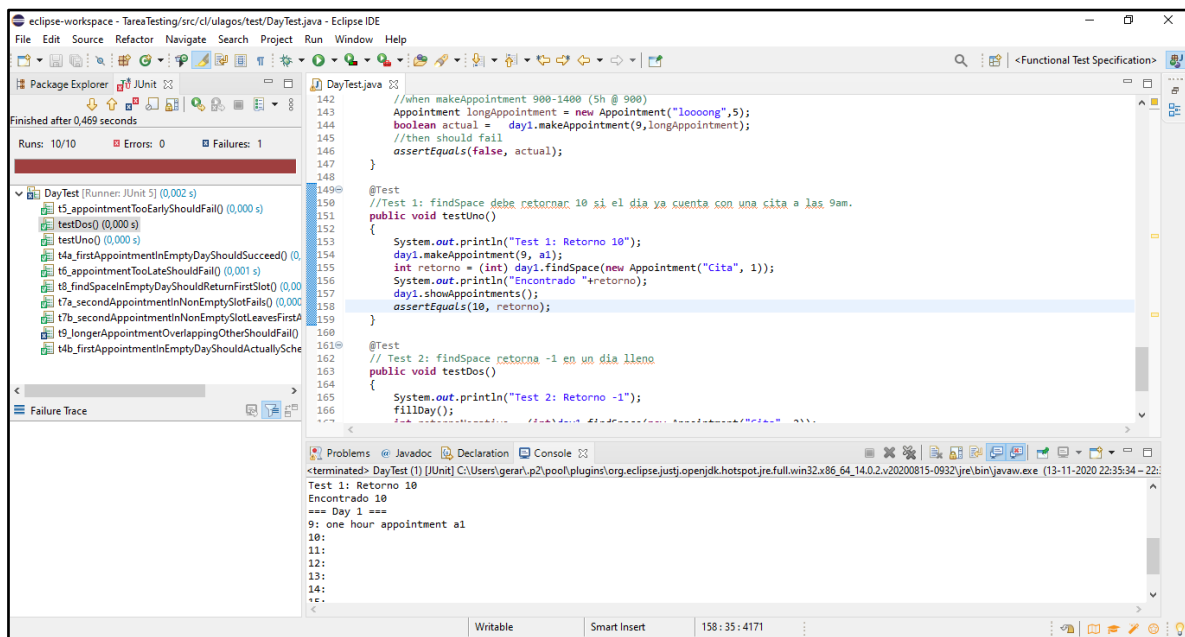Cátedra: Taller de Ingeniería de Software

- **Enunciado:**
    1. Crear un método de testing en la clase DayTest para verificar que findSpace retorne el valor 10 para una cita de una hora, si en el día ya se cuenta con una cita a las 9am.
    2. Crear un test para verificar que el método findSpace retorna el valor de -1, si se hace un intento de agendar una cita en un día que esté lleno.
    3. Crear un test negativo (usar la instrucción assertFalse) para el siguiente caso. Intentar inicializar una cita de un ahora y luego poner una cita de dos horas a la misma hora. Que sucede cuando corre el test? Incluir un screenshot en el reporte.
    4. Hacer cambios al programa para que acepte citas de media hora. Testear.

- <u>**Test n°1:**</u>

    Para este primer test se llamará a la función *makeAppointment()* para generar la primera cita con la hora y el nombre de la cita. Después se ocupará *findSpace()* la que buscará alguna cita que esté vacía en la hora más cercana para así que la agenda.

```java
@Test
    //Test 1: findSpace debe retornar 10 si el día ya cuenta con una cita a
las 9am.
    public void testUno()
    {
        System.out.println("Test 1: Retorno 10");
        day1.makeAppointment(9, a1);
        int retorno = (int) day1.findSpace(new Appointment("Cita", 1));
        System.out.println("Encontrado "+retorno);
        day1.showAppointments();
        assertEquals(10, retorno);

    }
```

- **Test n°2:**

  Para que la función *findSpace()* retorne -1, se tiene que hacer primero que llenar un día, así que se llama a la función *fillDay()*. Con la variable creada *retornoNegativo* se va a poder calcular si alguna hora del día creado se encontrará vacío y de no ser así imprimirá en pantalla el -1 tal y como se pide en el enunciado.

```java
@Test
// Test 2: findSpace retorna -1 en un dia lleno
public void testDos()
{
        System.out.println("Test 2: Retorno -1");
        fillDay();
        int retornoNegativo = (int)day1.findSpace(new Appointment("Cita", 2));
        day1.showAppointments();
        assertEquals(-1, retornoNegativo);
        System.out.println("---------------------------");
}
```

File   Edit   Source   Refactor   Navigate   Search   Project   Run   Window   Help

Package Explorer   JUnit

Finished after 0,469 seconds

Runs: 10/10   Errors: 0   Failures: 1

DayTest [Runner: JUnit 5] (0,002 s)
  t5_appointmentTooEarlyShouldFail() (0,000 s)
  testDos() (0,000 s)
  testUno() (0,000 s)
  t4a_firstAppointmentInEmptyDayShouldSucceed() (0,...
  t6_appointmentTooLateShouldFail() (0,001 s)
  t8_findSpaceInEmptyDayShouldReturnFirstSlot() (0,00
  t7a_secondAppointmentInNonEmptySlotFails() (0,000
  t7b_secondAppointmentInNonEmptySlotLeavesFirstA
  t9_longerAppointmentOverlappingOtherShouldFail()
  t4b_firstAppointmentInEmptyDayShouldActuallySche

Failure Trace

DayTest.java

```
152     {
153         System.out.println("Test 1: Retorno 10");
154         day1.makeAppointment(9, a1);
155         int retorno = (int) day1.findSpace(new Appointment("Cita", 1));
156         System.out.println("Encontrado "+retorno);
157         day1.showAppointments();
158         assertEquals(10, retorno);
159     }
160
161     @Test
162     // Test 2: findSpace retorna -1 en un dia lleno
163     public void testDos()
164     {
165         System.out.println("Test 2: Retorno -1");
166         fillDay();
167         int retornoNegativo = (int)day1.findSpace(new Appointment("Cita", 2));
168         day1.showAppointments();
169         assertEquals(-1, retornoNegativo);
170         System.out.println("----------------------------");
171     }
172 }
173
174
```

Problems   @ Javadoc   Declaration   Console

<terminated> DayTest (1) [JUnit] C:\Users\gerar\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_14.0.2.v20200815-0932\jre\bin\javaw.exe (13-11-2020 22:35:34 – 22:

```
Test 2: Retorno -1
=== Day 1 ===
9: one hour appointment a1
10: a2
11: a3
12: a4
13: a5
14: a6
15: a7
16: a8
17: a9
----------------------------
```