  
ESCUELA SUPERIOR   
PÓLITECNICA DEL LITORAL

**WORKSHOP ABOUT EMPIRICAL SOFTWARE TESTING**

**AUTORES:**

* AGUILAR MORA OSWALDO
* BERMUDEZ MOREIRA KAREN
* BERNAL MOREIRA GUILLERMO
* ORTIZ HOLGUIN EDUARDO
* WONG PAVON HUGO

**MATERIA:** ING. SOFTWARE II  
**TUTOR:** DR. MERA CARLOS  
**ENTREGA:** 2020/06/11

1. **Resumen**

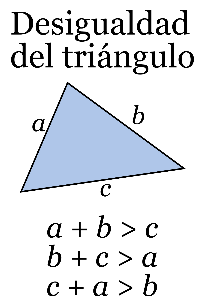
El presente documento contiene el reporte técnico correspondiente al primer taller grupal de nombre **“WORKSHOP ABOUT EMPIRICAL SOFTWARE TESTING”** del **GRUPO#4** perteneciente al curso de **INGENIERIA DE SOFTWARE II** del **PAO I**.

El reporte contiene una descripción del taller, el pseudocódigo, el planteamiento de las pruebas, la implementación en JAVA y el resultado de las pruebas.

1. **Descripción**

Se presenta la problemática de triángulo junto a un pseudocódigo que da solución a la identificación del tipo de triangulo, o si los valores ingresados no forman dicha figura, con sus respectivas validación, se nos pide en este taller implementar el pseudocódigo, diseñar los casos de pruebas que se consideren necesarios y finalmente ejecutar dichos casos de prueba.

* 1. **Problema del triángulo**

Es uno de los problemas más usados para la enseñanza de **pruebas de software**, consiste en un programa que debe tomar como entrada tres valores numéricos enteros que corresponden a los lados de un triángulo, debe evaluar las condiciones necesarias y finalmente determinar si los valores ingresados corresponden a un triángulo **escaleno**, a uno **equilátero**, uno **isósceles** o en última instancia los valores no forman un triángulo.

Para determinarlo se tienen las siguientes condiciones:

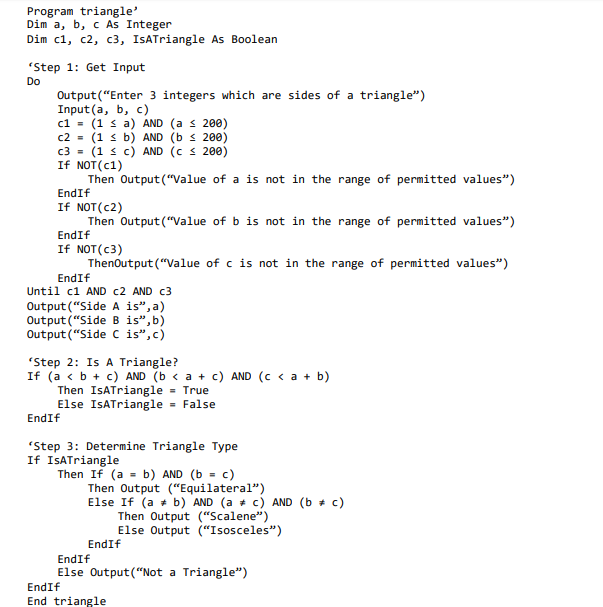
* Los valores ingresados deben estar en el rango de [1,200]
* Los valores deben cumplir con la **desigualdad triangular**.
* La igualdad de los lados determina si son equiláteros, escalenos o isósceles.

**Gráfico 1.- Desiguldad**

* 1. **Implementación**

Se pide realizarla implementación usando como lenguaje **JAVA** adicionalmente emplear alguna herramienta para realizar los test y la herramienta de desarrollo colaborativo **Git**.

1. **Pseudocódigo**



**Gráfico 2.- Pseudocódigo Problematica Triangulo**

1. **Diseño de Casos de Pruebas**

Se ha decidido dividir las pruebas en las siguientes categorías:

* Pruebas de Tipo de Datos
* Pruebas de Rango de Valores
* Pruebas de Resultados
  1. **Pruebas de Tipo de Datos**

En estas pruebas se evaluará la **robustez** del programa, es decir si no se cae debido a los valores ingresados y si notifica al usuario del error. Se probarán con valores no enteros ya sean de punto flotante o cadenas de texto.

* 1. **Pruebas de Rango de Valores**

En estas pruebas se evaluará la correcta validación de las variables, si se encuentran en el rango designado entre **1 a 200**, si muestran los mensajes correctos.

* 1. **Pruebas de Resultados**

En estas pruebas se evaluará los resultados finales luego de pasar las validaciones previas, se usarán valores correctos y se verificarán los resultados entre los 4 casos posibles **(Equilátero, Escaleno, Isósceles, No-Triangulo)**.

* 1. **Detalle de Casos de pruebas planteados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **# Caso** | **Comentario** | **a** | **b** | **c** | **Resultado** |
| **Pruebas de Tipo de Datos** | | | | | |
| **1** | String | 30 | b | 2 | **Error 1** |
| **2** | String | sk | 1 | Sd | **Error 1** |
| **3** | Decimales | 21 | 5 | 3,4 | **Error 1** |
| **4** | Decimales | 5,4 | 2,1 | 199,2 | **Error 1** |
| **5** | Mezcla | 4 | 7,5 | abc | **Error 1** |
| **Pruebas de Rango de Valores** | | | | | |
| **6** | Excedente a | 274 | 12 | 97 | **Error 2** |
| **7** | Excedente b | 76 | 599 | 3 | **Error 2** |
| **8** | Excedente c | 55 | 55 | 201 | **Error 2** |
| **9** | Insuficiente a | 0 | 2 | 2 | **Error 2** |
| **10** | Insuficiente b | 12 | -45 | 90 | **Error 2** |
| **11** | Insuficiente c | 124 | 125 | -1 | **Error 2** |
| **12** | Mezcla 1 | 156 | 500 | -12 | **Error 2** |
| **13** | Mezcla 2 | 0 | -999 | 999 | **Error 2** |
| **Pruebas de Resultados** | | | | | |
| **14** | Equilátero 1 | 5 | 5 | 5 | Equilátero |
| **15** | Equilátero 2 | 200 | 200 | 200 | Equilátero |
| **16** | Escaleno 1 | 5 | 3 | 7 | Escaleno |
| **17** | Escaleno 2 | 70 | 120 | 170 | Escaleno |
| **18** | Isósceles 1 | 6 | 3 | 6 | Isósceles |
| **19** | Isósceles 2 | 132 | 132 | 140 | Isósceles |
| **20** | No triangulo 1 | 6 | 13 | 6 | No triangulo |
| **21** | No triangulo 2 | 40 | 24 | 199 | No triangulo |
| **22** | No triangulo 3 | 1 | 2 | 3 | No triangulo |

1. **Código Fuente**
   1. **Repositorio**

Se uso **Github** como herramienta de colaboración para el desarrollo del taller, el vínculo es el siguiente:

* <https://github.com/kbermude/Taller1_Software>
  1. **Consideraciones de Desarrollo**

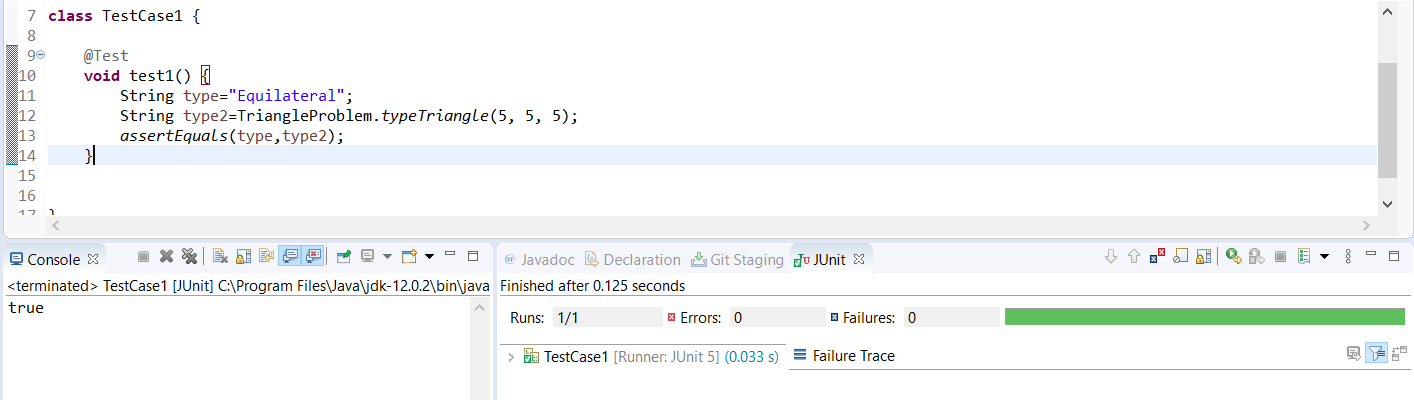
Para el desarrollo de la actividad y su implementación se agregaron los siguientes puntos:

* **Eclipse** se usó como IDE de Java para esta actividad.
* La herramienta **JUnit** se usó para realizar las pruebas.

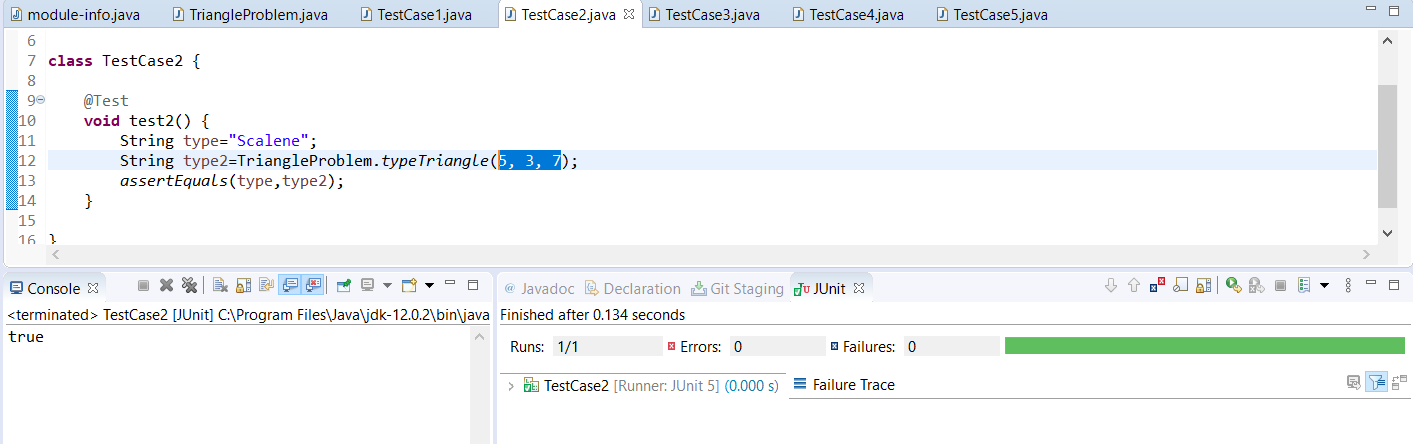
1. **Testeo y Resultados**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **# Caso** | **Comentario** | **a** | **b** | **c** | **Resultado Teórico** | **Resultado de prueba** |
| **Pruebas de Tipo de Datos** | | | | | |  |
| **1** | String | 30 | b | 2 | **Error 1** | **Caída** |
| **2** | String | sk | 1 | Sd | **Error 1** | **Caída** |
| **3** | Decimales | 21 | 5 | 3,4 | **Error 1** | **Caída** |
| **4** | Decimales | 5,4 | 2,1 | 199,2 | **Error 1** | **Caída** |
| **5** | Mezcla | 4 | 7,5 | abc | **Error 1** | **Caída** |
| **Pruebas de Rango de Valores** | | | | | |  |
| **6** | Excedente a | 274 | 12 | 97 | **Error 2** | **Error 2** |
| **7** | Excedente b | 76 | 599 | 3 | **Error 2** | **Error 2** |
| **8** | Excedente c | 55 | 55 | 201 | **Error 2** | **Error 2** |
| **9** | Insuficiente a | 0 | 2 | 2 | **Error 2** | **Error 2** |
| **10** | Insuficiente b | 12 | -45 | 90 | **Error 2** | **Error 2** |
| **11** | Insuficiente c | 124 | 125 | -1 | **Error 2** | **Error 2** |
| **12** | Mezcla 1 | 156 | 500 | -12 | **Error 2** | **Error 2** |
| **13** | Mezcla 2 | 0 | -999 | 999 | **Error 2** | **Error 2** |
| **Pruebas de Resultados** | | | | | |  |
| **14** | Equilátero 1 | 5 | 5 | 5 | Equilátero | Equilátero |
| **15** | Equilátero 2 | 200 | 200 | 200 | Equilátero | Equilátero |
| **16** | Escaleno 1 | 5 | 3 | 7 | Escaleno | Escaleno |
| **17** | Escaleno 2 | 70 | 120 | 170 | Escaleno | Escaleno |
| **18** | Isósceles 1 | 6 | 3 | 6 | Isósceles | Isósceles |
| **19** | Isósceles 2 | 132 | 132 | 140 | Isósceles | Isósceles |
| **20** | No triangulo 1 | 6 | 13 | 6 | No triangulo | No triangulo |
| **21** | No triangulo 2 | 40 | 24 | 199 | No triangulo | No triangulo |
| **22** | No triangulo 3 | 1 | 2 | 3 | No triangulo | No triangulo |

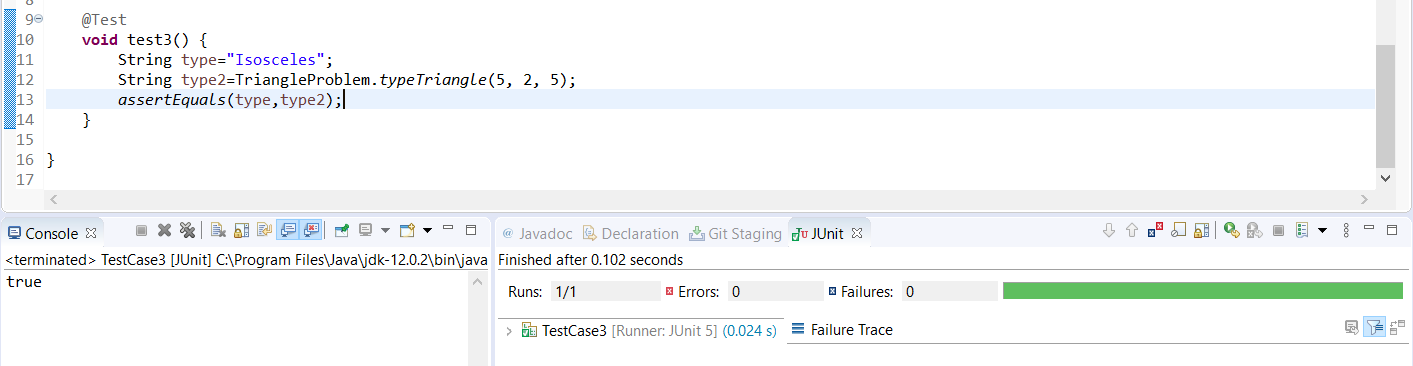
1. **Evidencia**



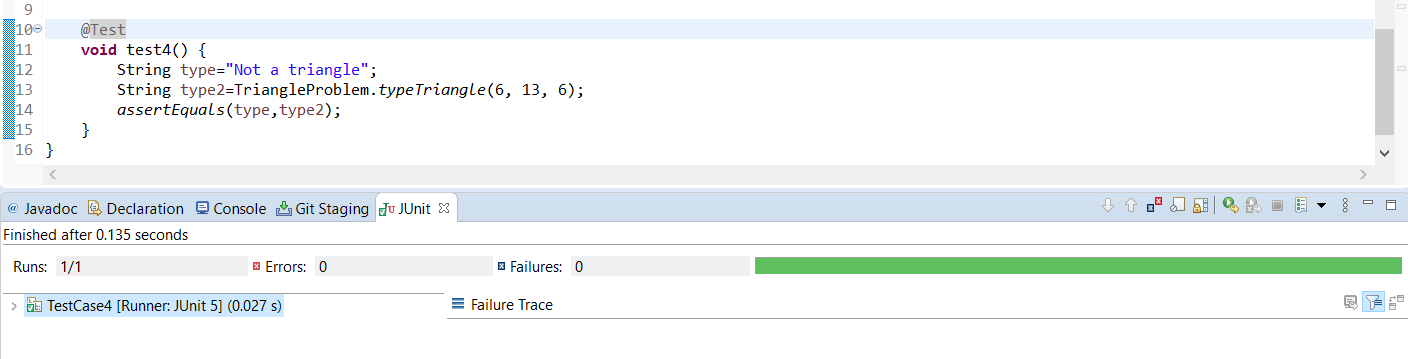
**Gráfico 3.- Test #14 (5,5,5)**



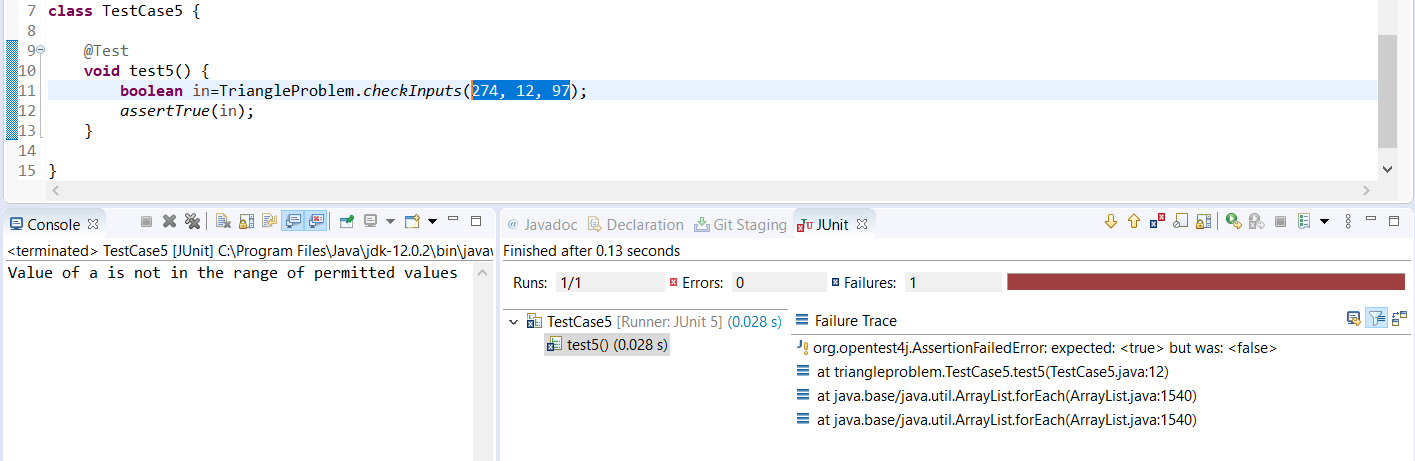
**Gráfico 4.- Test #16 (5,3,7)**



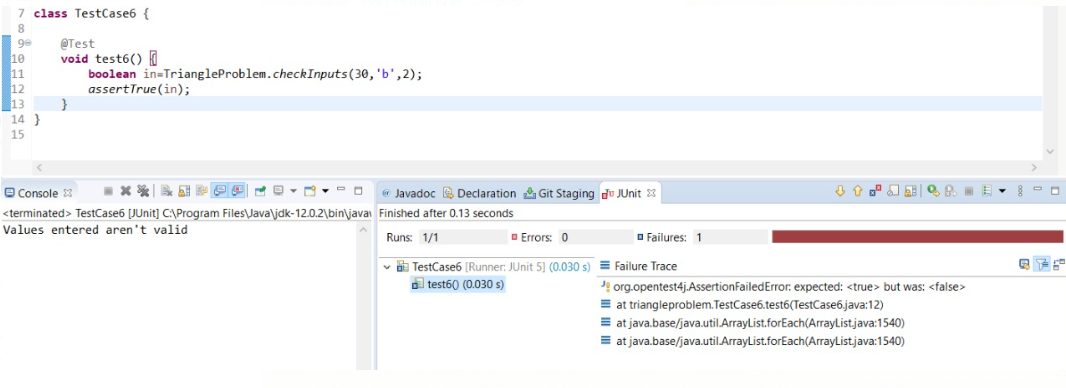
**Gráfico 5.- Test #18 (5,2,5)**



**Gráfico 6.- Test #20 (6,13,6)**



**Gráfico 7.- Test #6 (274,12,97)**



**Gráfico 8.- Test #1 (30,'b',2)**

1. **Conclusiones**

* El programa pasó **17/22** pruebas.
* Las **5** **fallas** son a pruebas del primer grupo **“Data type Tests”**, las entradas del programa están definidas a ser enteros, por lo que al recibir algo que nos e entero ocasiona una falla.
* El programa pasa perfectamente las pruebas de rango, valida correctamente si los valores exceden o son inferiores al rango preestablecido.
* El programa pasa perfectamente las pruebas de resultado, da resultados correctos si la figura es un triángulo equilátero, escaleno o isósceles.
* El programa pasa perfectamente las pruebas de resultado, muestra el mensaje correcto en caso de que los valores no formen un triángulo.
* **JUnit** es un herramienta muy útil para las pruebas de software.