

Autor: Andrés Castillero Moriana

Nota aclaratoria:

Estás realizando un examen, y como tal no puedes manejar ningún material de referencia adicional, ni comunicarte de ninguna forma con nadie. Estás sólo tú con estas instrucciones y las herramientas imprescindibles para llevarlas a cabo (Eclipse y Writer). Si se detecta cualquier amago de comunicación, automáticamente suspenderás.

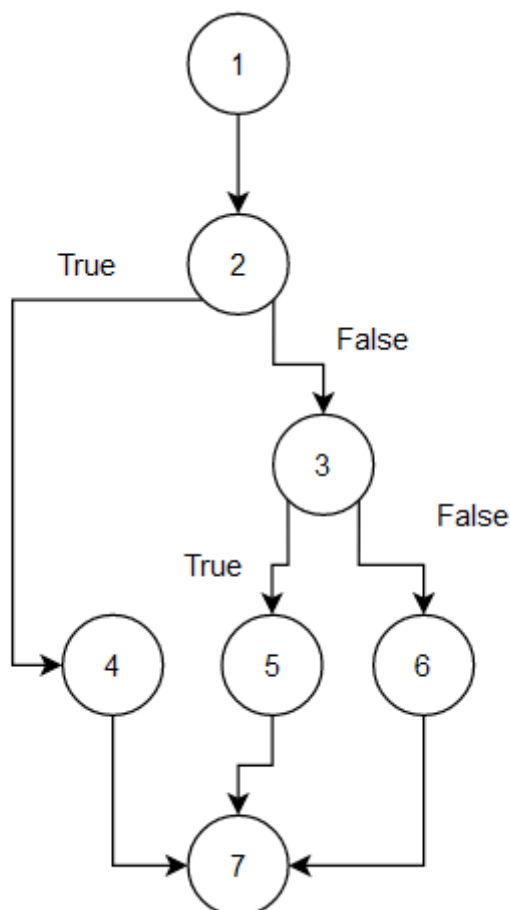
Utiliza el código MayorIgual que se adjunta. Muestra el mayor de dos números, indicando también si son iguales. Sigue las siguientes instrucciones.

Al final deberás entregar un fichero comprimido "NombreApellido1Apellido2ExEclEmma.rar" con dos ficheros:

1. NombreApellido1Apellido2ExEclEmma.pdf (Exportado a pdf desde OpenWriter)
2. NombreApellido1Apellido2ExEclEmmaProyecto. (Eclipse)

Vamos a diseñar un caso de pruebas para la clase MayorIgual. Realiza los siguientes pasos:

- Abre Eclipse y crea un proyecto "NombreApellido1Apellido2ExEclEmma".
- Añade el paquete mayorIgual que contiene MayorIgual.java.
- Crea el árbol de flujo del programa



- Indica la complejidad ciclomática y la fórmula con la que la has hallado

$$M = E - N + 2$$

$$8 - 7 + 2 = 3$$

- Indica también los valores necesarios para recorrer cada uno de los caminos básicos

$$a = 0 \quad b = 0.$$

$$a = 1 \quad b = 2.$$

$$a = 2 \quad b = 1.$$

- Añade al main() o mediante junit las invocaciones necesarias para la cobertura de dichos caminos básicos
- Entrega el proyecto, así como los siguientes pantallazos:

- Código de cada uno de los métodos con la cobertura/cubrimiento al 100%

The screenshot shows the Eclipse IDE with a Java project named 'exunit'. The main editor displays the code for 'Test.java' in the 'mayorIgual' package. The code defines a 'Test' class with a 'main' method that calls 'MayorIgual.mayorIgual' with various inputs. The 'Coverage' tab at the bottom shows the following data:

Element	Coverage	Verified Instruction	Missed Instruction	Total Instruction
exunit	14.2 %	54	325	379
src	14.2 %	54	325	379
(default package)	0.0 %	0	314	314
mayorIgual	83.1 %	54	11	65
Test.java	76.5 %	26	8	34
MayorIgual.java	90.3 %	28	3	31
MayorIgualTest.java	90.3 %	28	3	31
MayorIgualTest.java	100.0 %	28	0	28