Innovación en Formación Profesional

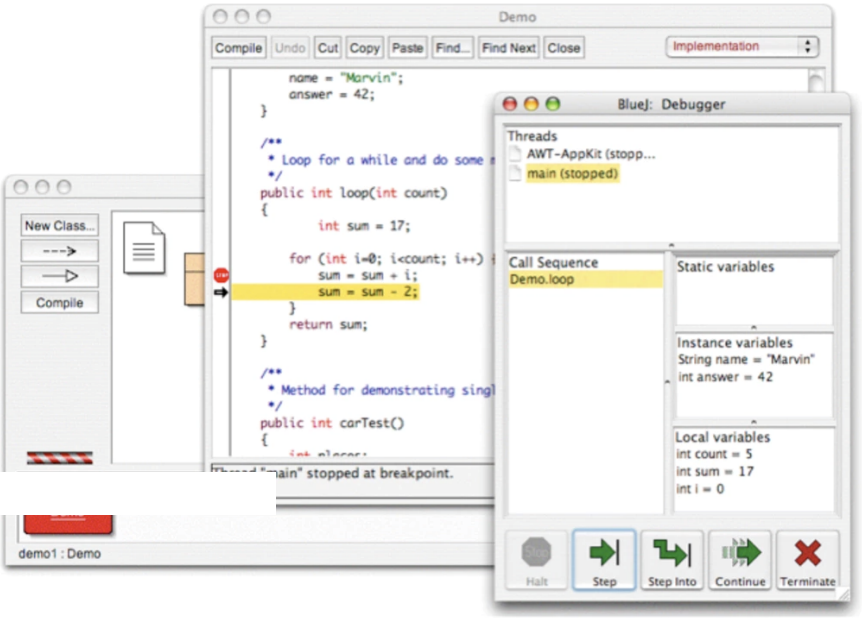
Módulo Profesional 05:
ENTORNOS DE DESARROLLO
Actividad UF2

Nombre del Estudiante: Enrique Verea Reimpell

CICLO FORMATIVO DE GRADO SUPERIOR EN

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

MODALIDAD ONLINE



Desarrollo de la actividad

Parte 1: Corrección de errores en la fase de testeo (4 puntos)

Resultado del banco de pruebas unitario de caja negra:

```
Calificaciones iFP

____________________

JUAN: El/La estudiante no ha presentado todos los ejercicios, ha
suspendido.

MARIA El/La estudiante ha presentado todos los ejercicios, ha aprobado
con una nota final de 7.2000003 puntos.

ANA: El/La estudiante ha presentado todos los ejercicios, ha suspendido
con una nota final de 4.9333334 puntos.
```

Recuerda que debes identificar los siguientes 6 errores (no importa el orden):

- 2 errores de sintaxis
- 1 error de conversión de datos
- 1 error de referencia o símbolo no válido
- 2 errores algorítmicos (de lógica o de codificación)

Utiliza la siguiente plantilla para describir cada error:

Error 1 (Error de sintaxis)	
Nombre del fichero	Modulo.java
Número de línea	3
Descripción del error	El constructor de la clase *Estudiante* espera 3 argumentos, pero sólo 2 han sido proporcionados
Solución aplicada	Añadir el argumento que falta. Se ha escogido el valor '0', en línea con el estilo visto en las construcciones de los siguientes *Estudiantes*, Maria y Ana

Error 2 (Error de sintaxis)	
Nombre del fichero	Modulo.java
Número de línea	19
Descripción del error	Se intenta invocar el método *println* de la salida estándar de la clase *System*, pero la palabra 'System' está mal escrita ('Syste')

Solución aplicada	Corregir el error tipográfico, sustituyendo la palabra 'Syste' por 'System'

Error 3 (Error de conversión de datos)	
Nombre del fichero	Modulo.java
Número de línea	7
Descripción del error	El método *setNotaUF1* de la clase *Estudiante* espera un argumento de tipo *float*, pero se le pasa un argumento tipo *double*.
Solución aplicada	Especificar que el número es un *float*, añadiendo una 'f' al final del mismo

Error 4 (Error de referencia o símbolo no válido)	
Nombre del fichero	Modulo.java
Número de línea	20
Descripción del error	Se intenta imprimir el estudiante 'Mario', pero no existe ninguna instancia o símbolo con ese nombre
Solución aplicada	Se cambia el nombre 'Mario' por 'Maria', instancia de *Estudiante* creada anteriormente

Error 5 (Error algorítmico)	
Nombre del fichero	Estudiante.java
Número de línea	46
Descripción del error	Se instancia el valor de la notaUF3, pero se instancia el valor de presentaUF3 como *false*, rompiendo la lógica de que si una UF tiene un valor de nota, ésta debió ser presentada
Solución aplicada	Se establece el valor de presentaUF3 a *true* siempre que se establezca un valor a notaUF3

Error 6 (Error algorítmico)	
Nombre del fichero	Estudiante.java
Número de línea	81
Descripción del error	Se pretende producir un *booleano* que indique si el *Estudiante* ha entregado todas las UF, sin embargo, se evalúa que el *Estudiante* haya presentado la UF1 y la UF2, o la UF3, cuando la lógica requiere la intersección del estado de entrega de las tres UF
Solución aplicada	Se cambia el operador '||' (OR) por el operador '&&' (AND), con lo que se indica que un *Estudiante* ha entregado todas las UF si y sólo si el estado de cada UF ha sido establecido como entregado

(**Adjuntar en la entrega una carpeta llamada "Parte 1" con el código fuente sin errores de esta primera parte del enunciado**)

Parte 2: Ejecución de un banco de pruebas unitario automatizado (3 puntos)

(**Adjuntar en la entrega una carpeta llamada "Parte 2" con la implementación de la prueba unitaria de caja blanca automatizada**)

Responder a la siguiente pregunta: ¿La aplicación supera con éxito TODAS las pruebas unitarias? En caso negativo, indicar qué prueba presenta fallos y el nivel de criticidad de dicho fallo.

No todas las pruebas unitarias son superadas con éxito:

En la prueba *registrarCalificacionesJuan* la afirmación *Juan.getNotaFinal == 6.6* falla. Esto es porque el método *getNotaFinal* produce un número *float* con un mayor número de decimales de los que se espera en la afirmación y cuyo valor supera el valor *delta* de 0.05 indicado. Sin embargo, este fallo de prueba no parece traducirse a un fallo en la aplicación, que no hace conversiones de los valores *float* a valores con un número determinado de decimales como se hace en la afirmación. En la aplicación, el resultado de *getNotaFinal* es utilizado por el método *toString*, que lo imprime en consola con todos sus decimales, y por el método *estaAprobado*, que produce una evaluación *mayor o igual que* 5, teniendo en cuenta todos sus decimales y sin realizar ningún redondeo.

En la prueba *copiarCalificaciones* la afirmación *Ana.equals(Juan) == true* falla. Esto se debe a que el método *equals* no ha sido implementado en la clase *Estudiante*, de manera que esta invocación de *equals* sólo compara si los dos objetos son la misma instancia, es decir, que las referencias apuntan al mismo objeto en memoria. Este fallo tampoco tiene ningún impacto en la aplicación, que no utiliza el método *equals* de ninguna manera.

Parte 3: Refactorización (3 puntos)

(**Adjuntar en la entrega una carpeta llamada "Parte3" con el código optimizado siguiendo las pautas de refactorización del enunciado**)