# Escuela Colombiana de Ingeniería PROGRAMACIÓN ORIENTADA A OBJETOS

## 2019-2

## Laboratorio 2/6

Integrantes:

Brayan Burgos

Daniel Alfonso

# Conociendo el proyecto

1. El proyecto BlueJ “calMat” contiene una construcción parcial del sistema. Revisen el directorio donde se encuentra el proyecto. Describan el contenido considerando los directorios y las extensiones de los archivos.  
   Contiene:
   1. Extensión .java que se refiere la construcción de las clases CalMat.java, MatrizTest.java, Matriz.java
   2. Extensión .class se refiere a la codificación para que la maquina pueda interpretar el código CalMat.class, Matriz.class, MatrizTest.class
   3. Extensión. ctxt se refiere a una documentación de los métodos de cada una de las clases CalMat.ctxt, Matriz.ctxt, MatrizTest.ctxt
   4. Directorio doc Contiene los recursos, paquetes que usaran en él proyecto y fallos en el archivo log
2. Exploren el proyecto en BlueJ

* ¿Cuántas clases tiene? ¿Cuál es la relación entre ellas?

El proyecto tiene dos clases CallMat y Matriz. La relación de CallMat a Matriz es un atributo de tipo Matriz en CallMat

* ¿Cuál es la clase principal? ¿Cómo la reconocen?

La clase principal es CalMat, porque en el diagrama que nos muestra BlueJ, las relaciones salen de CalMat

* ¿Cuáles son las clases “diferentes”? ¿Cuál es su propósito?  
  La clase “diferente” es MatrizTest y su propósito es realizar las pruebas de las diferentes operaciones entre matrices

Para las siguientes dos preguntas sólo consideren las clases “normales”:

1. Generen y revisen la documentación del proyecto; ¿está completa la documentación de

cada clase? (Detalle el estado de documentación de cada clase: encabezado y métodos)  
En la documentación esta el encabezado, el tipo de método con su respectivos parámetros y tipo de retorno, pero no describe que debe hacer cada método

1. Revisen el código del proyecto, ¿en qué estado está cada clase? (Detalle el estado de

codificación)

El código del proyecto solo contiene las clases, los encabezados y los métodos que se deben implementar.

# Ingeniería reversa

1. ¿Qué tipos de contenedores tienen sus colecciones? Consulte la especificación y el APIJava ¿Qué diferencias hay entre ellos?

<https://pediaa.com/what-is-the-difference-between-array-and-stack/>

# Conociendo Pruebas en BlueJ

Para poder cumplir con las prácticas XP vamos a aprender a realizar las pruebas de unidad usando las herramientas apropiadas. Para eso consideraremos implementaremos algunos métodos en la clase MatrizTest.

1. Revisen el código de la clase MatrizTest. ¿cuáles etiquetas tiene (componentes con símbolo @)? ¿cuántos métodos tiene? ¿cuántos métodos son de prueba? ¿cómo los reconocen?

@Before 1

@After 1

@Test 9

Total: 11

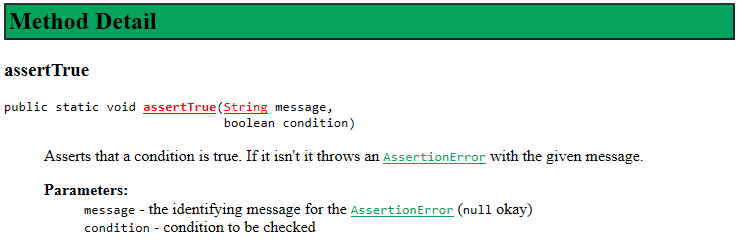
1. Ejecuten los tests de la clase MatrizTest. (click derecho sobre la clase, Test All) ¿cuántos tests se ejecutan? ¿cuántos pasan las pruebas? ¿por qué?

Se ejecutan 8 test, 3 pasan las pruebas porque no tienen ninguna implementación en su cuerpo de método, los demás fallan.

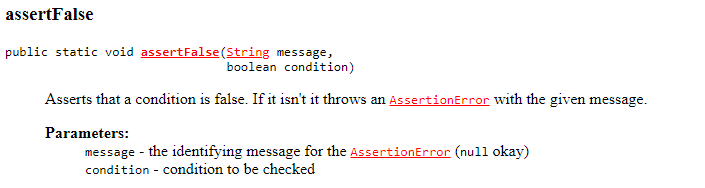
1. Estudie las etiquetas encontradas en 1. Expliquen en sus palabras su significado.

Significa que existen pruebas antes, durante y después de ejecutar el código

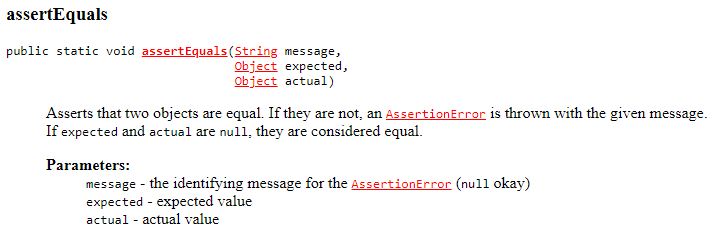
1. Estudie los métodos assertTrue, assertFalse, assertEquals, assertNull y fail de la clase assert del API JUnit. Explique en sus palabras que hace cada uno de ellos.



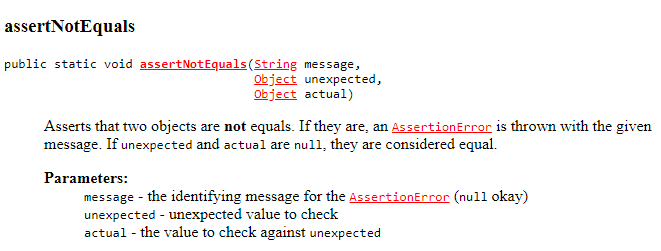
* El método necesita como parámetros un mensaje que indica si la prueba esta de manera correcta o no.
* En dado el caso de que el mensaje sea nulo, indicara el mensaje “[AssertionError](http://docs.oracle.com/javase/1.5.0/docs/api/java/lang/AssertionError.html?is-external=true" \o "class or interface in java.lang)”.
* el método necesita una condición, para validar su valor de verdad.



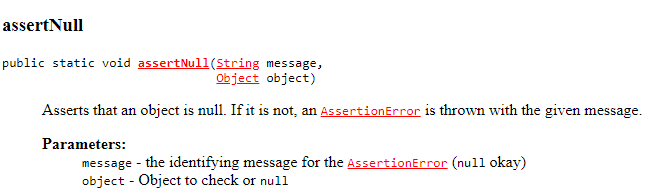
* El método necesita como parámetros un mensaje que indica si la prueba esta de manera incorrecta o no.
* En dado el caso de que el mensaje sea okay, indicara el mensaje “[AssertionError](http://docs.oracle.com/javase/1.5.0/docs/api/java/lang/AssertionError.html?is-external=true" \o "class or interface in java.lang)”.
* el método necesita una condición, para validar su valor de verdad, en este caso la idea es que sea False.



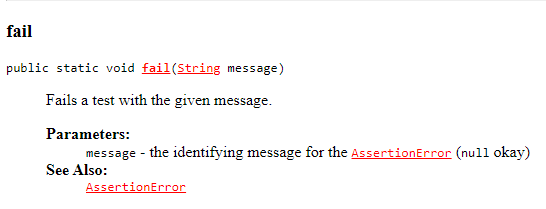
* El método necesita como parámetros un mensaje que indica si la prueba esta de manera correcta o no.
* Recibe dos objetos donde se comparan mediante características, datos y comportamientos.
* NOTA: existen pruebas para todo tipo de datos, se eligió arbitrariamente que la explicación fuese con objetos.



* El método necesita como parámetros un mensaje que indica si la prueba esta de manera correcta o no.
* Recibe dos objetos donde se comparan mediante características, datos y comportamientos, si el mensaje es ok, la prueba estaría correcta
* NOTA: existen pruebas para todo tipo de datos, se eligió arbitrariamente que la explicación fuese con objetos.



* El método necesita como parámetros un mensaje que indica si la prueba esta de manera correcta o no.
* Recibe dos objetos donde se comparan mediante características, datos y comportamientos, si el mensaje es ok, la prueba estaría correcta
* NOTA: existen pruebas para todo tipo de datos, se eligió arbitrariamente que la explicación fuese con objetos.



* El método necesita como parámetros un mensaje que indica si la prueba esta de manera correcta o no.
* Si el mensaje es nulo, indica un “[AssertionError](http://docs.oracle.com/javase/1.5.0/docs/api/java/lang/AssertionError.html?is-external=true" \o "class or interface in java.lang)”.

<https://junit.org/junit4/javadoc/latest/>

1. Investiguen la diferencia que entre un fallo y un error en Junit. Escriba código usando los métodos anteriores para lograr que los siguientes tres casos de prueba se comporten como lo prometen deberiaPasar, deberiaFallar, deberiaErrar.

Fallo es que la respuesta de la prueba esta mal pero el proceso se hizo.

# Practicando Pruebas en BlueJ

Ahora vamos escribir el código necesario para que las pruebas de MatrizTest.

1. Determinen las estructuras de datos necesarias para almacenar los elementos de una matriz. Justifique la selección.
2. Implementen los métodos necesarios para pasar todas las pruebas definidas. ¿Cuáles métodos implementaron?



***RETROSPECTIVA***

¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/

Hombre)

- 20 horas

¿Cuál es el estado actual del laboratorio? ¿Por qué?

-Incompleto, nos faltó dedicarle tiempo, además de la dificultad de no manejar la sintaxis de JAVA

Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?

-Programación por pares, ya que beneficia la productividad con el código

¿Cuál consideran fue el mayor logro? ¿Por qué?

-Darnos cuenta de nuestras debilidades con tiempo, para trabajar en ellas antes del parcial

¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

-La falta de conocimiento en el lenguaje JAVA. Consultar sobre la sintaxis, métodos, clases y demás en JAVA

¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los

resultados?

- Comunicar nuestras ideas, programar en pares. Dedicación a los procesos de desarrollo

disciplinados con los tiempos