**ESCUELA COLOMBIANA DE INGENIERÍA**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**2020-1**

**Laboratorio 2/6**

**Conociendo el Proyecto**

1. El proyecto BlueJ “calmatfra” 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.

Import.java.util.HashMap

Import static org.junit.Assert.\*

Import org.junit.before

Import org.junit.test

2. Exploren el proyecto en BlueJ

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

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

¿Cuáles son las clases “diferentes”? ¿Cuál es su propósito?

El Proyecto calmatfra tiene cuatro distintas clases, la clase principal es la clase calmatfra, que es una calculadora de matrices de fraccionarios,es la principal por que le pregunta la siguiente clase que se llama matriz si tiene una matriz y matriz le pregunta a fraccionario si tiene un numero fraccionario,todo esto lo hace para hacer operaciones en la calculadora con fraccionarios.

La cuarta clase es la clase diferente,se llama fraccionario test, y su proposito es validar los metodos incluidos en la clase fraccionario

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

3. 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)

La clase fraccionarios esta completa en su documentacion.

La clase calmatfra no esta completamente documentada,tiene el metodo asigne sin documentar el metodo consulta Tambien y el metodo ok Tambien esta sin documentar.

La clase Matriz esa mal documentada en la informacion de la clase, Tambien para sus metodos el metodo matriz que ingresa una matriz cubica esta sin documentar,el metodo,faltan muchas descripciones de los parametros

La clase de prueba esta bien documentada.

**Ingeniería reversa**

1. Genere el diagrama de clases correspondiente a calmatfra con todos sus elementos.

(No incluya la clase de pruebas)

2. ¿Qué tipos de contenedores tienen sus colecciones? Consulte la especificación y el API Java

1¿Qué diferencias hay entre ellos?

Tiene dos tipos de contedenedores uno hashMap y otro que es de la coleccion matriz [][]

La diferencia es que una e una hastable y la otra es una coleccion vectorial de un arreglo dentro de otro arreglo.

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

codificación)

La mayoria de metodos se encuentran vacios,se tienen pocas variables de clase,la mayoria de los constructors se encuentra vacio por lo tanto se tiene poca informacion de lo que realizan exactamente los metodos

**Conociendo Pruebas en BlueJ**

1. Revisen el código de la clase FraccionarioTest. ¿cuáles etiquetas tiene (componentes

con símbolo @)? ¿cuántos métodos tiene? ¿cuantos métodos son de prueba? ¿cómo los

reconocen?

1 etiqueta @test,16 metodos,todos son metodos de prueba y se puede comprobar por que en sus metodos utilizan assertequals.

2. Ejecuten los tests de la clase FraccionarioTest. (click derecho sobre la clase, Test

All) ¿cuántos tests se ejecutan? ¿cuántos pasan las pruebas? ¿por qué?

Se ejecutan los 16 test,de estos 3 pasan las pruebas, 13 las fallan.

Solo las 3 pasan por que deberiaPoderCrearFraccionariosNumeradorCero llama a numerador y este retorna cero se cumple la condicion.

mcdDeberiaSerConmutativo se cumple por que retorna 0 y 0=0 siempre se cumple

mcdDeberiaSerElMismoParaNegativos el mismo que el anterior por lo tanto se cumple.

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

@Test se colocara en los metodos que queremos que JUnit ejecute. Estos serán los métodos en los que implementemos nuestras pruebas.

4. Estudie los métodos assertTrue, assertFalse, assertEquals, assertNull y fail de

la clase assert del API JUnit 2

. Explique en sus palabras que hace cada uno de ellos.

**assertTrue:**Es un metodo que afirma que lo que esta incluido en el es cierto.

**assertFalse:** Es un metodo que afirma que lo que esta incluido no es cierto.

**assertEquals:** Es un metodo que afirma que lo que esta incluido en la comparacion es igual.

**assertNull: :** Es un metodo que afirma que lo que esta incluido es nulo.

**Fail:**Falla una prueba sin enviar un mensaje

5. 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.

La diferencia es que un fallo es la invalidacion del resultado de una prueba y el error es una mala ejecucion de la prueba

**Prácticando Pruebas en BlueJ**

1. Determinen las estructuras de datos necesarias para almacenar los elementos de un fraccionario. Justifique la selección.

Un arraylist que contenga el numerador y el denominador porque es la manera mas eficiente para revisar los numeros cuando se tengan que hacer operaciones.

RETROSPECTIVA

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes?

(Horas/Hombre)

Javier lopez-13 horas

Andres Davila-13 horas

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

Esta terminado con las pruebas de unidad perfecta

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

La mas util fue la de first, por que nos enseña la importancia de realizer una prueba de unidad y la manera en como estas se deben realizar.

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

Esta vez si trabajamos en equipo y logramos comprender la importancia de hacer unos Buenos test de unidad y llegar temprano

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

Ninguno todo fue muy facil

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

resultados?

Aplicamos muy muy bien los principios de pair programming, nos comprometemos a seguir manteniendo la calidad de nuestros trabajos.