LABORATORIO 2

PROGRAMACION ORIENTADA A OBJETOS

ELABORADO POR:

DIEGO FERNANDO RUIZ ROJAS

JUAN FELIPE AGUAS PULIDO

PRESENTADO A: MARIA IRMA DIAZ ROZO

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

PROGRAMACIÓN ORIENTADA A OBJETOS

PROGRAMA DE INGENIERÍA DE SISTEMAS

BOGOTÁ D. C.

12 DE FEBRERO DE 2021

**Conociendo el proyecto**

1. El proyecto BlueJ “relationalCalculator” 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.

2. Exploren el proyecto en BlueJ

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

El proyecto contiene 3 clases de las cuales dos están relacionadas a una misma clase Table. Las otras dos clases son la clase RelationalCalculator y la clase de pruebas TableTest

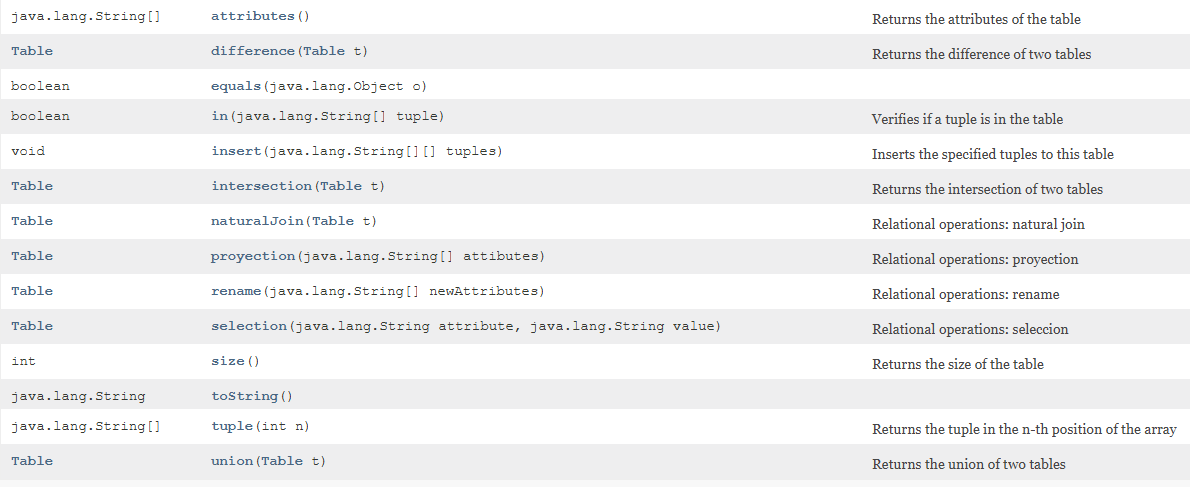
1. ¿Cuál es la clase principal? ¿Cómo la reconocen?

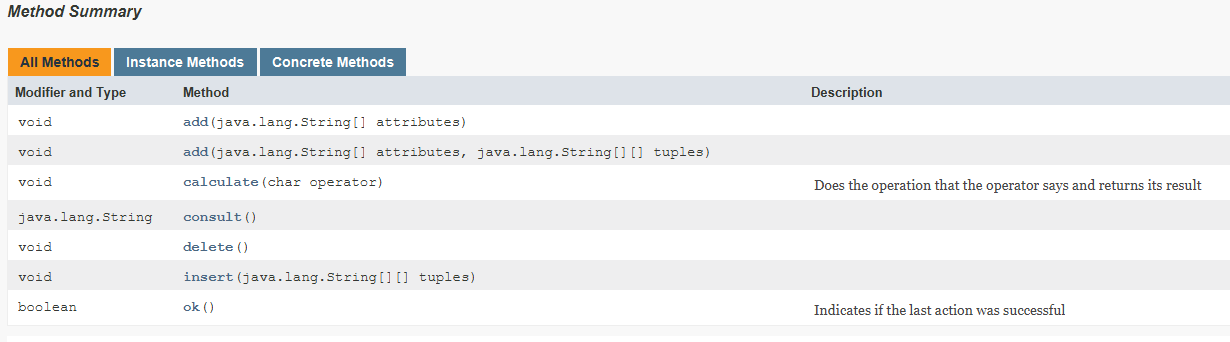
La clase principal es la clase Table puesto que es la que está relacionada a las otras dos clases.

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

**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? (Detallen el estado de documentación de cada clase: encabezado y métodos)





4. Revisen las fuentes del proyecto, ¿en qué estado está cada clase? (Detallen el estado de las fuentes considerando: código, documentación y comentarios) ¿Qué son el código, la documentación y los comentarios?

**Ingeniería reversa**

1.Genere el diagrama de clases correspondiente a relationalCalculator con todos sus elementos. (No incluya la clase de pruebas)

1. Se encuentra en el archivo relationalCalculator.asta

2. ¿Qué tipo de contenedor está definido? Consulte la especificación y el API Java 1¿Qué diferencias hay entre el Stack, ArrayList,[]?

1. Stack maneja una política de manejo de datos LIFO, ArrayList no tiene una política definida. Por otra parte, stack solo posee 5 métodos propios mientras que ArrayList ofrece una gran variedad de métodos para consultar, modificar o agregar datos.

[]

ArrayList también puede ser comparada con una LinkedList puesto que tiene algunas de sus características y su tamaño no debe ser definido puesto que esta es una estructura de datos dinámica.

**Conociendo pruebas en Bluej**

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

1. ¿cuáles etiquetas tiene (componentes con símbolo @)?

-@Before

-@Test

-@After

1. ¿cuántos métodos tiene?

TableTest cuenta con 9 metodos.

1. ¿cuántos métodos son de prueba?

Hay en total 7 metodos de prueba.

1. ¿cómo los reconocen?

Se reconocen con la etiqueta @Test

2.Ejecuten los tests de la clase TableTest. (click derecho sobre la clase, Test All)

1. ¿cuántas pruebas se ejecutan?

Se ejecutan en total 7 pruebas.

1. ¿cuántos pasan las pruebas? ¿por qué?

Pasan 4 pruebas porque cada prueba muestra que el código debería fallar y en este momento el código no ha sido desarrollado y terminado debidamente.

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

1. @Before

Esta etiqueta muestra aquellos métodos que organizan e inicializan los métodos de prueba

1. @Test

Esta etiqueta nos indica los métodos que ponen a prueba el código inicializadas con los métodos etiquetados con @Before.

1. @After

Esta etiqueta señala los métodos que acaban las pruebas y finaliza el proceso de pruebas mostrando los resultados de las pruebas inicializadas en @Before

4. Estudie los métodos assertTrue, assertFalse, assertEquals, assertArrayEquals, assertNull y fail de la clase assert del API JUnit 2 . Explique en sus palabras que hace cada uno de ellos.

* assertTrue

Verifica que el parámetro que ingresa sea un boolean verdadero

* assertFalse

Verifica que el parámetro que ingresa sea un boolean falso

* assertEquals

Verifica que los parámetros cumplan su atributo equals()

* assertArrayEquals

Verifica que los parámetros cumplan su atributo equals()

* assertNull

Verifica que el parámetro ingresado sea nulo

* fail

Hace fallar un test a propósito

5. Investiguen la diferencia que entre un fallo y un error en Junit. Escriba código, usando los métodos del punto 4., para lograr que los siguientes tres casos de prueba se comporten como lo prometen shouldPass, shouldFail, shouldErr.

La diferencia entre un error y un fallo es que el error se genera dentro del test en un problema del código mientras que un fallo se genera cuando el resultado del assert es false.

**Practicando Pruebas en Bluej**

1. Determinen la forma en que van a almacenar los elementos de un conjunto. Justifique la selección. Para esto revisen el conjunto de pruebas definido hasta el momento.

Serán almacenados por medio de arrays y ArrayLists debido a que a nuestro parecer estas son las mejores estructuras de datos para manejar las tablas y los conjuntos

1. Implementen únicamente los métodos de Table necesarios para pasar todas las pruebas definidas. ¿Cuáles métodos implementaron?