Laboratorio 2

Juan Cancelado

Santiago Cordoba

Programacion orientada a objetos

Maria Irma Diaz

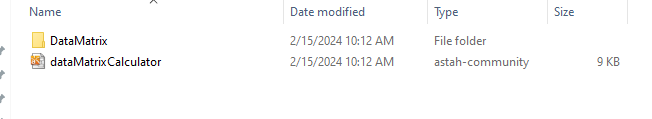
Escuela Colombiana De ingeniería Julio Garavito

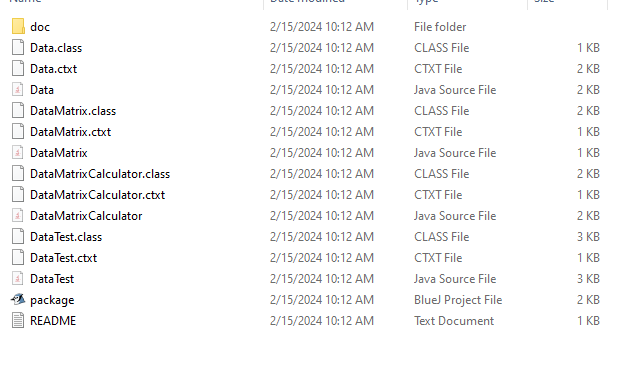
Bogotá D.C

2024

**Conociendo el proyecto**

1)El proyecto “dataMatrixCalculator” 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

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

El Proyecto contiene 3 clases y sus relaciones son 2, las cuales datamatrixcalculator tiene una datamatrix y la otra seria la relacion de pruebas con la data.

**b) ¿Cuál es la clase principal de la aplicación? ¿Cómo la reconocen?**

La principal clase de la aplicacion es DataMatrixCalculator.

La reconocemos que es la principal porque sin esta no se podria hacer la calculadora y Tambien esta hace todos las operaciones.

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

la clase diferente es DataTest y el proposito de esta clase es generar las pruebas necesarias para verificar su funcionamiento.

**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)**

a) No esta completa la documentacion ni en el cabezado, ni en los metodos.

Por ejemplo en Data temenos complete el en cabezado y temenos documentado casi todos los metodos.

Pero en los otros dos que son DataMatrix y DataMatrixCalculator no esta ni el en cabezado ni la documentacion de los metodos.

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

a) en Data temenos el encabezado esta complete y practicamente todos los metodos exceptuando uno que estan documentados, algunos metodos estan comentados para que nosotros completemos el metodo para que realicemos dichas instrucciones.

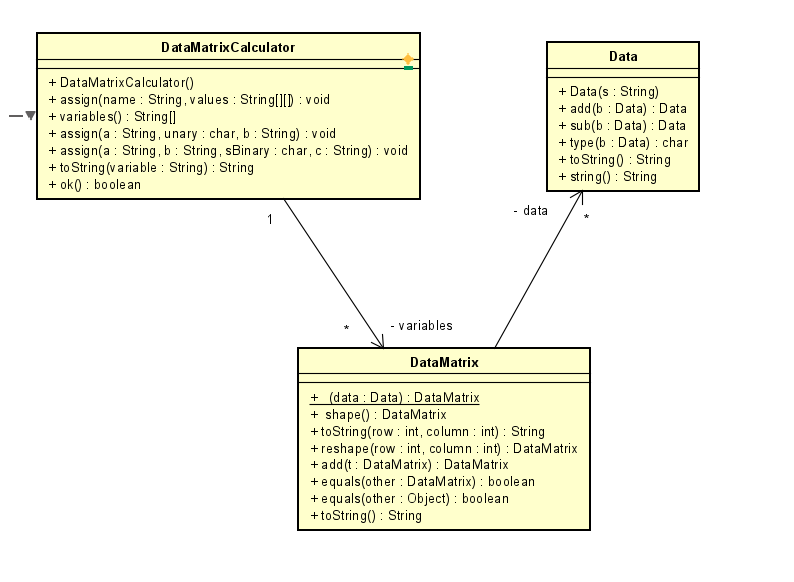
En DataMatrixCalculator no estan documentadas ni los metodos, ni el encabezado, pero Tambien aparecen los comentarios para realizer dicho metodo.

En DataMatrix solo hay una cosa documentada es el constructor y el resto de metodos no esta documentado, y tampoco hay comentarios para completer los metodos.

Dentro de los metodos de las clases se puede detallar los atributos mediante los argumentos que entran a estos.

**Ingeniería reversa**

**1)**

****

2)

**¿Cuáles contenedores están definido? ¿Qué diferencias hay entre el nuevo contenedor, el ArrayList y el vector [] que conocemos? Consulte el API de java. ¿Cómo adicionamos un elemento? ¿Cómo lo consultamos? ¿Cómo lo eliminamos?**

Los contenedores que estan definidos son Hashmap.

El Hashmap: es una implementación de map el cual almacena llave y valor.

No siempre están ordenadas.

Permite las llave y valores nulos.

Para eliminar un elemento utilizamos remove(key)

ArrayList: es una implementación de list que almacena elementos en un arreglo dinámico.

Mantiene un orden.

Permite elementos repetidos y nulos.

Para eliminar un elemento utilizamos remove(element)

Vector: es una estructura de datos similar a ArrayList, pero es sincronizada.

Para eliminar un elemento utilizamos remove(índice) o remove(element)

**//Toda esta información fue sacada de chatgpt.**

**Conociendo Pruebas en BlueJ**

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

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

Tenemos 7 etiquetas.

7 métodos.

Los métodos de prueba son: shouldKnowWhenTwoCharactersAreEquals, shouldKnowWhenTwoNumbersAreEquals, shouldKnowWhenTwoBooleansAreEquals, shouldRepresentADataAsAString y shouldCreateData.

Los reconozco porque en la etiqueta sale @Test y también dentro del método esta haciendo pruebas de la funcionalidad.

1. **Ejecuten los tests de la clase DataTest. (click derecho sobre la clase, Test All) ¿cuántas pruebas se ejecutan? ¿cuántas pasan? ¿por qué? Capturen la pantalla.**
2. **Estudie las etiquetas encontradas en 1 (marcadas con @). Expliquen en sus palabras su significado.**

La etiqueta @BeforeClass significa que se va a ejecutar una sola vez antes de que se ejecuten. todas las pruebas en la clase.

La etiqueta @Before seria prácticamente lo mismo que @BeforeClass.

La etiqueta @Test significa que es una prueba acerca de la funcionalidad especifica de la clase.

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

//Información sacada de chatgpt

assertTrue: este método verifica que la condición proporcionada sea verdadera y si es falsa el método fallara y enviara el mensaje de error.

assertFalse: este método verifica que la condición proporcionada sea falsa, y si es verdadera el método fallara y enviara el mensaje de error.

**assertEquals: permite comparar dos valores y si son iguales la prueba funciona y si no falla.**

**assertArrayEquals: compara dos matrices para verificar si son iguales.**

**assertNull: verifica que el objeto proporcionado sea nulo**

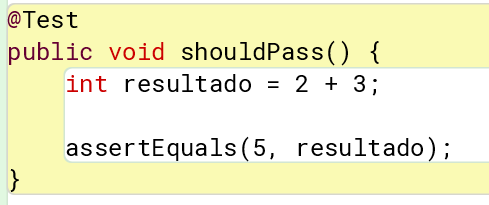
**fail: este método hace que falle siempre.**

1. **En JUnit, se distingue entre un fallo y un error:**

//Realizado con ayuda de chatgpt

**Fallo (Failure): Sucede cuando una aserción en una prueba unitaria no se verifica. En resumen, un fallo indica que la prueba unitaria no genera el resultado esperado según lo especificado por las aserciones en el código de prueba.**

**Error: Se produce cuando una excepción no controlada surge durante la ejecución de una prueba unitaria. Esto puede ser resultado de errores en el código de la prueba o de condiciones inesperadas durante su ejecución.**

****

**Texto

Descripción generada automáticamente con confianza media**

**Texto

Descripción generada automáticamente con confianza baja**

**Prácticando Pruebas en BlueJ**

1. **Determinen los atributos de la clase Data. Justifique la selección.**

Los atributos que consideramos en la clase Data son: un booleano llamado bool, un entero llamado n y un String llamado s.

1. Ni idea.
2. L**os métodos que implementamos para que se puedan realizar las pruebas son los siguientes:**

* El constructor

Imagen que contiene Tabla

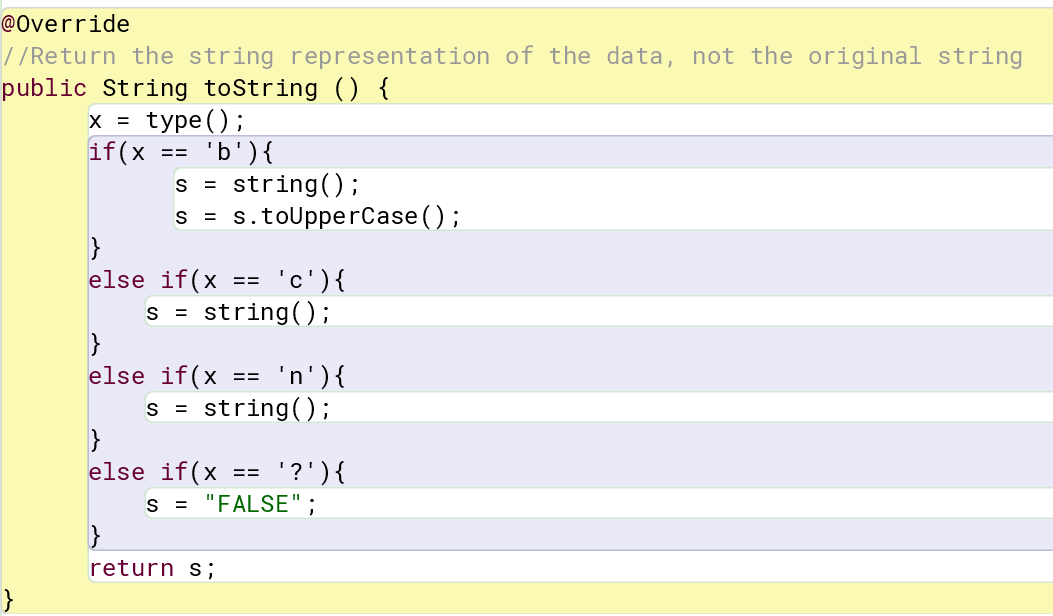
Descripción generada automáticamente

* Type

**Aplicación

Descripción generada automáticamente**

* toString

****

* string

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media**

* equals

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

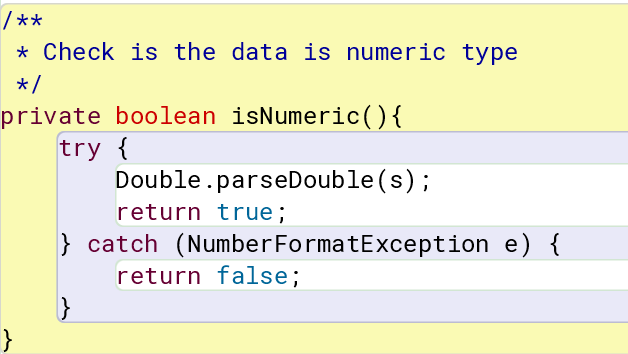
Descripción generada automáticamente

* isBoolean

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* isNumeric



* isChar

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Capturen los resultados de las pruebas de unidad.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Desarrollando DataMatrixCalculator**

**Ciclo 1**

1. Métodos definidos:

En DataMatrix

* Constructor
* toString(int row, int column)
* toString

En DataMatrixCalculator

* Constructor
* Assign(String name, String values[][])
* Variables
* toString

**Ciclo 2**

1. Métodos definidos:

En DataMatrix

* Shape
* Transpose

En DataMatrixCalculator

* assign (String a, char unary, String b)

1. Métodos definidos:

En DataMatrix

- add

- sub

-equals

En DataMatrixCalculator

* Assign (String a, String b, char sBinary, String c)

|  |  |  |
| --- | --- | --- |
| Mini-ciclo | DataMatrixCalculator | DataMatrixCalculatorTest |
| 1 | 4(y 3 de DataMatrix) |  |
| 2 | 1(y 2 de DataMatrix) |  |
| 3 | 1(y 3 de DataMatrix) |  |

**RETROSPECTIVA**

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

10/Santiago Cordoba y 10/Juan Cancelado.

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

Casi completo porque nos quedamos cortos de tiempo.

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

Si, porque nos ayudamos a ver errores escribiendo código juntos uno codificaba y el otro miraba si la lógica y todo estaba correcto y también hicimos mini ciclos para cada parte del laboratorio.

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

poder entender como se comportan las clases con su test porque siempre nos saltaba error hasta que logramos arreglar las clases

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

El mayor logro es que pasaran todas las pruebas pero lo logramos conociendo cada método a la perfección de la clase.

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

Nos comunicamos muy bien, y cada uno confía en el otro. Hacer el laboratorio con mas tiempo de antelación.