Tatiana Medina – Jose Pérez

POOB - 1

2020 - 2

Laboratorio 2/6 Diseño y pruebas

**Conociendo el proyecto**

1. El proyecto BlueJ “calConjuntos” 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.

El proyecto CalConjuntos está constituido por 3 clases:

* Conjunto: Se asume que esta clase pretende simular un set
* CalConjuntos: Se asume que esta clase es la implementa los métodos para la calculadora de pila con sets.
* ConjuntoTest: Se asume esta clase contiene las pruebas unitarias para los métodos de la clase conjunto.

1. Exploren el proyecto en BlueJ
   * ¿Cuántas clases tiene?

Tiene 3 clases: Conjunto, CalConjunto, ConjuntoTest

* + ¿Cuál es la relación entre ellas?

CalConjunto implementa conjunto para las operaciones de la calculadora, ConjuntoTest son las pruebas unitarias para la clase Conjunto

* + ¿Cuál es la clase principal?

La clase principal es CalConjunto

* + ¿Cómo la reconocen?

Basándonos en las líneas del diagrama mostrado en BlueJ

* + ¿Cuáles son las clases “diferentes”?

La clase diferente seria ConjuntoTest

* + ¿Cuál es su propósito?

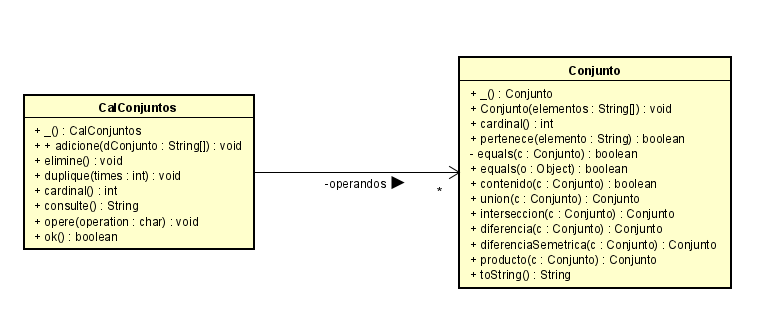
Comprobar que cada uno de los métodos de la clase Conjunto funciona de manera correcta.

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? (Detallen el estado de documentación de cada clase: encabezado y métodos)
     1. Conjunto: Esta clase pretende simular un set, la documentación está completa.
     2. CalConjuntos: Esta clase es la implementa los métodos para la calculadora de pila con sets), la clase no está documentada.
     3. ConjuntoTest: Esta clase continen las pruebas unitarias para los métodos de la clase conjunto, la clase no está documentada
2. Revisen el código del proyecto:
   * ¿en qué estado está cada clase? (Detallen el estado de las fuentes considerando: código, documentación y comentarios)
     1. Conjunto: Esta clase pretende simular un set, los métodos solo están definidos (no hacen nada).
     2. CalConjuntos: Esta clase es la implementa los métodos para la calculadora de pila con sets, los métodos solo están definidos (no hacen nada).
     3. ConjuntoTest: Esta clase contiene las pruebas unitarias para los métodos de la clase conjunto, los métodos están implementados.

**Ingeniería Reversa**

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



1. ¿Qué tipo de contenedor está definido? Consulte la especificación y el API Java

El contenedor definido es Stack.

public class **Stack<E>**

extends [Vector](https://docs.oracle.com/javase/8/docs/api/java/util/Vector.html)<E>

La clase Stack representa una estructura last-in-first-out (LIFO) de almacenamiento de objetos

1. ¿Qué diferencias hay entre éste y el ArrayList?

En un ArrayList se tiene una lista de elementos y se puede acceder a cualquiera de ellos en cualquier momento y todo está referenciado al inicio del ArrayList. En un Stack puro no se puede acceder a cualquier elemento aleatorio del Stack, en general en un Stack solo se puede adicionar/remover sobre el final del Stack, se tendrá acceso de lectura a cualquier elemento, pero esta referenciado sobre el final.

**Conociendo Pruebas en BlueJ**

1. Revisen el código de la clase ConjuntoTest.
   * ¿cuáles etiquetas tiene (componentes con símbolo @)?

Tiene las etiquetas @Before, @Test y @After

* + ¿cuántos métodos tiene?

Posee 12 métodos

* + ¿cuántos métodos son de prueba?

Hay 10 métodos de prueba

* + ¿cómo los reconocen?

Se reconocen por la etiqueta @Test que acompaña al método.

1. Ejecuten los tests de la clase ConjuntoTest. (click derecho sobre la clase, Test All)
   * ¿cuántas pruebas se ejecutan?

Se ejecutan 10 pruebas

* + ¿cuántos pasan las pruebas?

Pasan 3 pruebas

* + ¿por qué?

Las tres pruebas que pasan son *DeberiaFallar, DeberiaErrar, DeberiaPasar*; puesto que no contienen ningún *assert* en su cuerpo

1. Estudie las etiquetas encontradas en 1. Expliqen en sus palabras su significado
   * @Before: Se utiliza para marcar el método que se ejecuta antes de cada prueba
   * @Test: Se utiliza para marcar el método cada prueba
   * @After: Se utiliza para marcar el método que se ejecuta al finalizar cada prueba
2. Estudie los métodos assertTrue, assertFalse, assertEquals, assertNull y fail de la clase assert del API JUnit

* *assertTrue:* Está compuesto por un mensaje y una condición, si la condición es verdadera se pasará la prueba
* *assertFalse:* Está compuesto por un mensaje y una condición, si la condición es falsa se pasará la prueba
* *assertEquals:* Está compuesto por una serie de argumentos, si estos son iguales se pasará la prueba
* *assertNull:* Está compuesto de un argumento, este será el que se quiere checkear su valor inicial, esperando que sea nulo
* *Fail:* Está compuesto por un mensaje, siempre que se alcanza esta sentencia se lanzara un *Exception* con ese mensaje

1. Investiguen la diferencia 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.*

Se produce un fallo cuando la prueba no se comporta como se espera i.e . los *assert* son incorrectos. Y se produce un error cuando en el cuerpo de la prueba ocurre un fallo en la ejecución ej. *Exception*

**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.

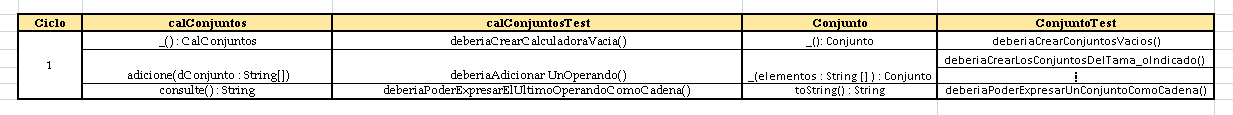
Se van a almacenar los elementos del conjunto en un ArrayList de tipo String. Ya que este nos deja almacenar una variedad de elementos de todo tipo, haciendo así que la aplicación funcione con un rango de elementos bastante amplio.

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

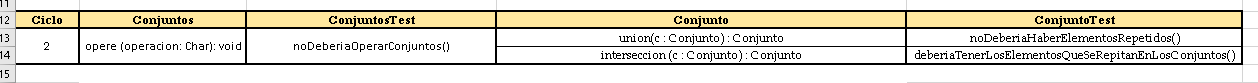
* \_ (): Conjunto
* \_ (elementos: String []): Conjunto
* cardinal (): int
* pertenece (elemento: String): boolean
* toString (): String

**Desarrollando CalConjuntos**

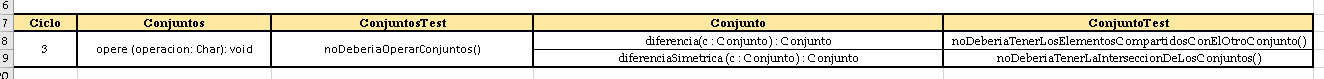
* Ciclo 1: Operaciones básicas de pila: creación, adicione, consulte
  1. Definir los métodos base correspondientes al ciclo
* \_ (): CalConjuntos
* adicione (dConjunto: String []): void
* consulte (): String
  1. Completar la tabla de clases y métodos



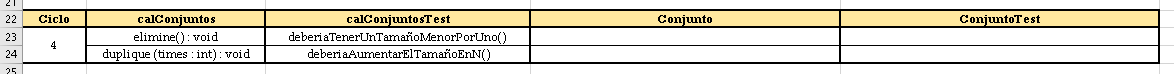
* **Ciclo 2: Operaciones básicas de conjuntos: unión e intersección**
  1. Definir los métodos base correspondientes al ciclo
     + opere (operacion: Char): void
* union (c: Conjunto): Conjunto
* intersección (c: Conjunto): Conjunto
  1. Completar la tabla de clases y métodos



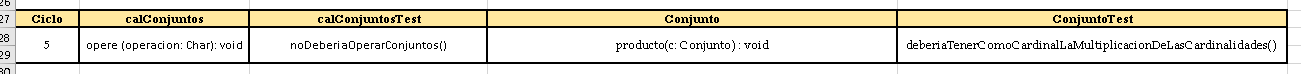
* **Ciclo 3: Operaciones adicionales de conjuntos: diferencia y diferencia simétrica**
  1. Definir los métodos base correspondientes al ciclo
     + opere (operacion: Char) : void
* diferencia (c: Conjunto): Conjunto
* diferenciaSimetrica (c: Conjunto): Conjunto
  1. Completar la tabla de clases y métodos



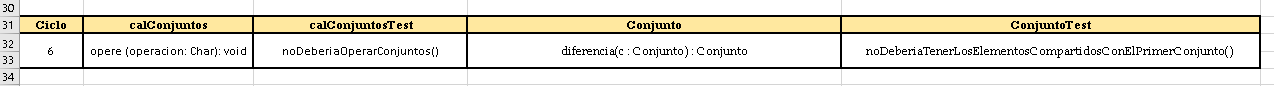
* **Ciclo 4: Operaciones adicionales de pila: Elimine, duplique**
  1. Definir los métodos base correspondientes al ciclo
* *elimine (c: Conjunto): Conjunto*
* *duplique (c: Conjunto): Conjunto*
  1. Completar la tabla de clases y métodos



* **Ciclo 5: Operación avanzada de conjuntos: Producto**
  1. Definir los métodos base correspondientes al ciclo
     + opere (operacion: Char) : void
* producto ()
  1. Completar la tabla de clases y métodos



* **Ciclo 6: Defina una nueva funcionalidad**
  1. Definir los métodos base correspondientes al ciclo
     + opere (operacion: Char) : void
* diferencia (c: Conjunto): Conjunto
  1. Completar la tabla de clases y métodos



**RETROSPECTIVA**

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Hombre) 8 horas Angie Medina / 8 horas Jose Perez.
2. ¿Cuál es el estado actual del laboratorio? ¿Por qué?

Se hizo aproximadamente un 100% del laboratorio debido a que no tuvimos grandes dificultades con respecto a este.

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

Consideramos que de las practicas que se vieron en este laboratorio la más útil fue el *Unit Test First* ya que nos permitió saber con más claridad que se estaba haciendo mal al momento de codificar y así mismo saber cuándo se había terminado de codificar dicho comportamiento.

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

Terminar completamente el laboratorio porque el tema de pruebas unitarias es nuevo para nosotros

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

Los computadores y el internet a veces representaban problemas técnicos, nos tocaba esperar un tiempo para volver a trabajar o reiniciar los computadores.

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

Consideramos que nos comunicamos muy bien como equipo, pero igual nos comprometemos a implementar de mejor manera las practicas xp vistas.