Lab #02 POOB

David Rodríguez

Sergio Peña

**Conociendo el proyecto**

1. Con respecto a un vistazo inicial, Se puede apreciar que hay cuatro archivos principales las cuales son: Angulo, calculadora, AnguloTest y vector. Cada uno tiene tres archivos con archivos diferentes las cuales son \*.ctx \*.class y un java source.
   1. Tiene 4 clases.
   2. La clase Calculadora tiene tres relaciones las cuales son ángulo, vector y una a sí misma la cual es calculadora. Vector tiene solo una, la cual es a ángulo. Angulo no tiene relaciones con nada. Y el ultimo AnguloTest tiene relación con ángulo.
   3. La clase calculadora es la principal. Se reconoce ya que por pensamiento y razonamiento lógico de las relaciones según los conceptos pre dispuestos. También se podría decir que el proceso de inicialicen empieza con él.
   4. La clase “diferente” que se puede decir que es AnguloTest. Y como anteriormente dicho, por razón lógica, son pruebas dl código.
2. A todas las clases la falta documentación, sin mencionar el hecho de que es escasa y muy ambiguo la documentación que hay.
   1. Calculadora:
      1. Dice que representa una calculadora de vectores
      2. El constructor no dice nada
      3. Adicione solo le entran dos variables doubles.
      4. Consulte solo devuelve un String
      5. Duplique no se sabe nada, solo que es void.
      6. Elimine no se sabe nada, también void.
      7. Ok devuelve un boolean
      8. Opere recibe un char.
   2. Vector:
      1. Dice en el constructor que son vectores en coordenadas polares
      2. En ángulo dice que retorna el ángulo del vector.
      3. coordenadaX retorna la coordenada x del vector.
      4. CoordenadaY lo mismo que coordenadaX, pero en y.
      5. Distancia recibe un vector y retorna la distancia entre uno y otro.
      6. Equals recibe un objeto y declara si un vector es igual al parámetro.
      7. Devuelve la longitud del vector el método longitud.
      8. Multiplique recibe un vector, pero no retorna nada.
      9. productoEscalar recibe un doublé, y lo multiplica el vector por su producto escalar.
      10. Reste no dice que hace pero recibe un vector
      11. Rote recibe un ángulo y rota el vector
      12. Sume recibe un vector y no retorna nada.
      13. toString describe el vector en cadena
      14. traslade recibe dos doubles y desplaza a esas posiciones descritas
   3. Angulo:
      1. El constructor recibe un doublé y un int crea un angulo
      2. Coseno calcula el coseno
      3. Divida recibe un ángulo y divide ese ángulo
      4. Equals compara dos ángulos
      5. Equals compara el objeto
      6. Gradientes devuelve el valor en gradianes
      7. Grados valor del ángulo en grados
      8. Multiplique multiplica el ángulo con un real
      9. Radianes valor del ángulo en radianes
      10. Reste, resta el ángulo con otro
      11. Seno calcula le seno del ángulo
      12. Sume calcula la suma de este ángulo con otro
      13. toString devuelve el valor en grados
3. Está incompleta por lo anterior dicho y la lista representativa de los métodos.

**Conociendo Pruebas en BlueJ**

1. Las condiciones son que en ángulo recibe como parámetros un doublé llamado valor y un entero llamado tipo. Los diferentes tipos son grados, radianes y gradianes. Entre ellas existe una formula o un tipo de proceso en la cual hace que se conviertan entre ellas. Es un objeto que cuando se crea el estado de aquel no puede ser modificado. Tiene las ventajas de asegurar que los objetos no se modifiquen y así todos sus atributos están asegurados.
2. Tiene 9 pruebas. Solo una no está implementada que es tearDown
3. Todos se ejecutan, pero ninguno pasa la prueba. Pasan porque todo el código se ejecuta correctamente.
4. El error es cuando hay algo más escrito en el código y el fallo es cuando no pasa la prueba.
5. assertTrue verifica que lo devuelva el código true, assertEquals compara los objetos que se ingresan y si son iguales la prueba es exitosa. Y fail tiene la propiedad de que cuando tiene un error, no aparece el porqué de ese error.
6. Está en Angulo test
7. Está en Angulo test
   1. deberiaCrearBienLosAngulo() esta prueba incluye el método equals() y grados()
   2. deberiaDarElValorEnGrados() esta prueba incluye el método grados()
   3. deberiaDarElValorEnRadianes() esta prueba incluye el método radianes()
   4. deberiaDarElValorEnGradianes() esta prueba incluye el método gradianes()
   5. deberiaSumar() esta prueba incluye el método sume()
   6. deberiaRestar() esta prueba incluye el método reste()
   7. deberiaMultiplicar() esta prueba incluye el método multiplique()
   8. deberiaCalcularElSeno() esta prueba incluye el método seno()
   9. deberiaCalcularElCoseno() esta prueba incluye el método coseno()
   10. deberiaPoderExpresarUnAnguloComoCadena() esta prueba incluye el método toString()
   11. deberiaDividir() esta prueba incluye el método divida()
   12. deberiaMultiplicarAnguloConAngulo() esta prueba incluye el método multiplique()
8. Implementamos el método para saber en que cuadrante está el ángulo.

**Desarrollando**

Ciclo 1:

* adicione()
* consulte()
* ok()

Ciclo 2:

* elimine()
* duplique()

Ciclo 3:

* opere()

Ciclo 4:

* opere()

**TABLA:**

Calculadora:

1. adicione()
2. elimine()
3. duplique()
4. consulte()
5. opere()
6. ok()

CalculadoraTest:

1. deberiaAdicionar()
2. deberiaEliminar()
3. deberiaDuplicar()
4. deberiaOperar()
5. deberiaConsultar()

Vector:

1. coordenadaX()
2. coordenadaY()
3. angulo()
4. longitud()
5. distancia()
6. equals ()
7. equals(Object o)
8. traslade ()
9. productoEscalar()
10. rote()
11. sume()
12. reste()
13. multiplique()
14. toString()

VectorTest

1. deberiaCrearBienLosVectores()
2. deberiaDarLasCoordenadasEnX()
3. deberiaDarLasCoordenadasEnY()
4. deberiaDevolverElAngulo()
5. deberiaDevolverLaLongitud()
6. deberiaDecirLaDistancia()
7. deberiaTrasladar()
8. deberiaMultiplicarEscalar()
9. deberiaRotar()
10. deberiaSumar()
11. deberiaRestar()
12. deberiaMultiplicarVectores()
13. deberiaConvertirloString()

Angulo:

1. grados()
2. radianes()
3. gradianes()
4. sume()
5. reste()
6. multiplique()
7. divida()
8. multiplique(double r)
9. equals()
10. equals(Object o)
11. seno()
12. coseno()
13. cuadrante()
14. toString()

AnguloTest:

1. deberiaCrearBienLosAngulo()
2. deberiaDarElValorEnGrados()
3. deberiaDarElValorEnRadianes()
4. deberiaDarElValorEnGradianes()
5. deberiaSumar()
6. deberiaRestar()
7. deberiaMultiplicar()
8. deberiaCalcularElSeno()
9. deberiaCalcularElCoseno()
10. deberiaSerNulo()
11. deberiaFallar()
12. deberiaPoderExpresarUnAnguloComoCadena()
13. deberiaDividir()
14. deberiaMultiplicarAnguloConAngulo()
15. deberiaSaberCuadrante()

**RETROSPECTIVA:**

1. 26 horas por cada uno.
2. El estado del laboratorio es completo, no hace falta nada.
3. La práctica xp más útil fue la de las primeras primero ya que era la guía de este laboratorio.
4. El mayor logro fue en si todo el laboratorio, nos sentimos a gusto con nuestro trabajo.
5. El mayor problema técnico fue en la suma de vectores porque no la podíamos hacer muy bien, pero después de preguntarle a la profesora todo se arregló.
6. Hicimos bien como equipo la repartición de tareas y el tiempo ya que siempre tratamos de adelantar lo más posible para acabar bien y rápido.