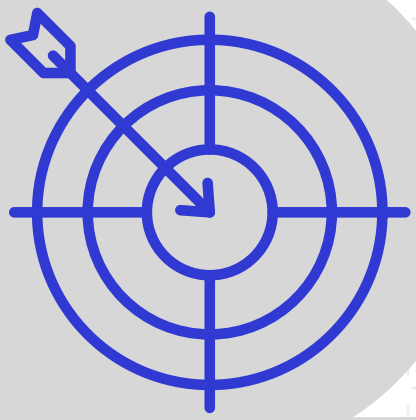
El futuro digital es de todos

MinTIC

Cubriendo el código con JUnit

Universidad de Caldas



Objetivo

Al finalizar la semana el estudiante debe estar en la capacidad de comprender y complementar un proyecto Java para integrar las pruebas unitarias con JUnit.

Problema

El reto de esta semana se enfoca en la práctica de la lógica para realizar pruebas unitarias en el código java de la aplicación que se esté desarrollando. Adjunto a este documento se encuentra un proyecto que cuenta con dos paquetes, el primero de ellos son las Clases y el segundo es el de Inicio. En las clases encontrarán dos clases, una clase dirección con sus atributos y un método “getAddress()” que retorna la información completa de la dirección de una persona. En la segunda clase llamada “clsPersona” se encuentran definidos todos los atributos y 3 métodos, el primero es el constructor que recibe cierta cantidad de atributos para asignar, y dentro del constructor se inicializan otros con valores predeterminados.

Por otra parte, en la misma clase clsPersona se cuenta con el método WalkAround que recibe los metros a caminar y retorna la sumatoria de todos los metros caminados durante diferentes ejecuciones del método. También se cuenta con el método AddRelative que permite agregar un familiar a la lista de familiares y retorna la posición de la lista en la cual quedó dicho familiar. Cabe aclarar que los familiares son objetos de la misma clase persona.

Para el desarrollo del reto se solicita complementar la clase clsPersona con los métodos de updatePerson para modificar los datos de la persona retornando un entero con el resultado de la operación (1: no se encuentra la persona, 2: error actualizando los datos, 3: persona actualizada correctamente). Además del método deleteRelative para eliminar un familiar de la lista de familiares, en este se podrá retornar una excepción si la lista de familiares está vacía, 1 en caso de que la lista no esté vacía pero tampoco exista el familiar en esta, o un 2 en caso de que se hubiese podido remover.

Posteriormente se deben generar las pruebas unitarias para todos los métodos de las clases presentadas anteriormente. Cabe aclarar que se debe utilizar un objeto Mock donde lo considere necesario para simular los objetos de los cuales se depende.

Universidad de Caldas