

Guía Práctica No. 1: Unit Testing

Ej. 1. Considere la clase `SampleStaticRoutines` provista en el moodle de la asignatura, y realice lo siguiente:

- Provea varios tests *positivos* para `originalMax`.
- Provea varios tests *positivos* y *negativos* para `max`.
- Implemente `max` para que sea genérica.
- Corra todos los tests producidos, y depure sus implementaciones.

Ej. 2. Considere la clase `Point` provista en el moodle de la asignatura, y realice lo siguiente:

- Construya tests para la clase (al menos un par por cada método).
- Implemente el método `equals` de la clase.
- En caso de haber detectado bugs, corrijalos y vuelva a ejecutar los tests, hasta que todos pasen.

Ej. 3. Considere nuevamente la clase `SampleStaticRoutines` provista en el moodle de la asignatura, y realice lo siguiente:

- Testee los métodos `largest` y `bubbleSort`, intentando proveer cierta variedad en las entradas posibles de cada método.
- En caso de haber detectado bugs, corrijalos y vuelva a ejecutar los tests, hasta que todos pasen.

Ej. 4. Considere las clases `Mine` y `Minefield`, provistas en el moodle de la asignatura. Realice lo siguiente:

- Provea tests para ambas clases, que evalúen todos los métodos.
- Para el caso particular de `minedNeighbours` de la clase `Minefield`, provea tests que describan el comportamiento esperado de esta rutina. Evalúe la posibilidad o necesidad de utilizar métodos de `setUp` y `tearDown` para estos tests.
- Implemente el método `minedNeighbours` de la clase `Minefield`.

Ej. 5. Considere nuevamente la clase `SampleStaticRoutines` provista en el moodle de la asignatura, y provea tests parametrizados para evaluar las rutinas `largest` y `bubbleSort`.