

Ejercicio:

Un Array se caracteriza por:

- ▶ vector: array de enteros.

Únicamente se podrá acceder a las características de Array a través de métodos de la propia clase.

Las acciones que se pueden realizar con un array son:

- ▶ el constructor al que se le pasa por parámetro un entero que indica la cantidad de enteros que puede almacenar. Por defecto será 0.
- ▶ modifica: se le pasan por parámetro dos enteros; el primero indica la posición del array que se quiere modificar y el segundo el valor con el que se modifica dicha posición. Si la posición no pertenece al rango del array se lanza y propaga una excepción *RangoNoValido* pasándole al constructor de la excepción la posición no válida.

- consulta: se le pasa por parámetro un entero que es la posición del array que se quiere consultar, de manera que devuelve su contenido. Si la posición no pertenece al rango del array se lanza y propaga una excepción *RangoNoValido* pasándole al constructor de la excepción la posición no válida.

Un *ArrayRango* es un *Array* que se define por un rango, posición inicial de acceso y posición final de acceso, de manera que la posición inicial de acceso puede ser distinta de la posición 0 habitual. Es decir, un *ArrayRango* se puede crear con un rango $[3, 6]$, de manera que sus posiciones válidas de acceso serían la posición 3 (correspondiente a la posición 0 habitual), la 4 (correspondiente a la posición 1 habitual), la 5 (correspondiente a la posición 2 habitual) y la 6 (correspondiente a la posición 3 habitual).

Se pide:

- ▶ Implementa las clases `Array` y `ArrayRango`;
- ▶ Implementa la clase `RangoNoValido` de manera que a su constructor se le pase por parámetro un entero y que su mensaje asociado sea dicho entero.
- ▶ Implementa una clase `Uso` con un `main` en el que:
 - ▶ se declara una variable de tipo `Array` y se crea un objeto de tipo `ArrayRango` para el rango `[-1,7]` en esta variable;
 - ▶ se modifican las posiciones 1 con el valor -2, 6 con el valor 10, -1 con el valor 7 y 8 con el valor 1;
 - ▶ se consultan y muestran por pantalla las posiciones 6, 1, -1 y -2.
- ▶ Implementa el manejador de la excepción donde sea necesario de manera que muestre por pantalla la excepción.
- ▶ Indica lo que se muestra por pantalla al ejecutar la clase `Uso`.