



1. Un contador digital es un contador acotado que comienza de nuevo cuando se alcanza un valor máximo. Ejemplos incluyen los números en un reloj digital (el mínimo aquí es el valor 0 y el máximo es el valor 59) y el odómetro de un coche (el valor mínimo es 0 y el máximo 9 para cada dígito). Implementa una clase para representar este tipo de contadores. Se debe poder establecer el valor mínimo y el valor máximo, se debe permitir incrementar el contador y se debe permitir obtener el valor actual del contador.

NOTA:

Para obtener el resto de la división de un número por otro se puede utilizar el método:

```
public static double IEEERemainder(float f1, float f2)  
de la clase Math.
```

2. Escribe una clase para trabajar con números racionales (descritos mediante dos enteros, el numerador y el denominador). Proporciona métodos para sumar, restar, multiplicar y dividir fracciones. También se debe proporcionar un método para mostrar la fracción como un *float*.
3. Realizar una clase Punto2D para representar un punto del plano 2-D. Realizar una clase Rotacion para realizar rotaciones en el plano 2-D.

La rotación de un punto $\begin{bmatrix} x \\ y \end{bmatrix}$ un ángulo θ proporciona un nuevo punto que se puede obtener como:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta)x - \sin(\theta)y \\ \sin(\theta)x + \cos(\theta)y \end{bmatrix}$$

El constructor de la clase Rotacion debe recibir el ángulo de rotación. Esta clase de proporcionar un método para rotar un punto 2-D y obtener otro punto 2-D:

```
public Punto2D rota(Punto2D p)
```

NOTA:

La clase Math tiene una serie de métodos estáticos para trabajar con funciones trigonométricas:

```
public static double cos(double radianes)  
public static double sin(double radianes)  
además define la constante  $\pi$ :  
public static final double PI
```

En todos los casos hay que prestar atención a los siguientes aspectos:

- Cuales son los atributos que debe poseer cada clase.
- Cuales son los argumentos que se deben proporcionar a cada método.
- Qué debe devolver cada método.
- Decidir los modificadores de acceso para los atributos y métodos.