

Ejercicio 1

Diseña la clase **Calendario** que representa una fecha concreta (año, mes y día). La clase debe disponer de los métodos:

- **Calendario(int año, int mes, int dia)**: que crea un objeto con los datos pasados como parámetros, siempre y cuando, la fecha que representen sea correcta.
- **void incrementarDia()**: que incrementa en un día la fecha del calendario.
- **void incrementarMes()**: que incrementa en un mes la fecha del calendario.
- **void incrementarAnio(int cantidad)**: que incrementa la fecha del calendario en el número de años especificados. Ten en cuenta que el año 0 no existió.
- **void mostrar()**: muestra la fecha por consola.
- **boolean iguales(Calendario otraFecha)**: que determina si la fecha invocante y la que se pasa como parámetro son iguales o distintas.

Por simplicidad, solo tendremos en consideración que existen meses con distinto número de días, pero no tendremos en cuenta los años bisiestos.

Ejercicio 2

Escribe la clase Punto que representa un punto en el plano (con un componente x y un componente y), con los métodos:

- **Punto(double x, double y)**: construye un objeto con los datos pasados como parámetros.
- **void desplazaX(double dx)**: incrementa el componente x en la cantidad **dx**.
- **void desplazaY(double dy)**: incrementa el componente y en la cantidad **dy**.
- **void desplaza(double dx, double dy)**: desplaza ambos componentes según las cantidades **dx** (en el eje x) y **dy** (en el componente y).
- **double distanciaEuclidea(Punto otro)**: calcula y devuelve la distancia euclídea entre el punto invocante y el punto **otro**.
- **void muestra()**: muestra por consola la información relativa al punto.

Ejercicio 3

Un conjunto es una estructura dinámica de datos como la lista, con dos diferencias: en primer lugar, en una lista puede haber elementos repetidos, mientras que, en un conjunto, no. Además, en una lista el orden de inserción de los elementos puede ser relevante y debemos tenerlo en cuenta, mientras que en un conjunto solo interesa si un elemento pertenece o no al conjunto y no el lugar que ocupa. Se pide implementar la clase Conjunto utilizando una lista para almacenar números de tipo **Integer**. Implementa los siguientes métodos:

- Un constructor sin parámetros.
- **int numeroElementos()**: devuelve el número de elementos del conjunto.
- **boolean insertar(Integer nuevo)**: inserta un nuevo elemento en el conjunto.
- **boolean insertar(Conjunto otroConjunto)**: añade al conjunto los elementos del conjunto **otroConjunto**.

- **boolean eliminarElemento(Integer elemento)**: en caso de pertenecer al conjunto, elimina **elemento**.
- **boolean eliminarConjunto(Conjunto otroConjunto)**: elimina del conjunto invocante los elementos del conjunto que se pasa como parámetro.
- **boolean pertenece(Integer elemento)**: indica si el elemento que se le pasa como parámetro pertenece o no al conjunto.
- **muestra()**: muestra el conjunto por consola.

De forma general, los métodos que devuelven un booleano indican con él si el conjunto se ha modificado.

Ejercicio 4

Añade a la clase **Conjunto** los siguientes métodos estáticos:

- **static boolean incluido(Conjunto c1, Conjunto c2)**: que devuelve **true** si **c1** está incluido en **c2**, es decir, si todos los elementos de **c1** están también en **c2**.
- **static Conjunto unión(Conjunto c1, Conjunto c2)**: devuelve un nuevo conjunto con todos los elementos que están en **c1**, en **c2** o en ambos (elementos comunes y no comunes).
- **static interseccion(Conjunto c1, Conjunto c2)**: que devuelve un nuevo conjunto con todos los elementos que están en **c1** y en **c2** a la vez (elementos comunes).
- **static diferencia(Conjunto c1, Conjunto c2)**: que devuelve un nuevo conjunto con todos los elementos que están en **c1**, pero no en **c2**.