PRÁCTICA PRUEBAS IDM

Ejercicio 1 - Cálculo de años bisiestos

- **Apartado 1.** Se analizará la clase `Bisiestos`, donde uno de sus métodos llamado `esBisiesto`, devuelve true o false si el año que le proporcionamos es o no bisiesto.
- **Apartado 2.** Tenemos 1 parámetro:
  - Tenemos un único Parámetro `año`, de tipo *int*. Este será el año que queremos comprobar si es bisiesto o no.
- **Apartado 3.** Tendremos 3 caracterizaciones:
  - C1: año respecto del 0.
    - b1: año negativo
    - b2: año 0
    - b3: año positivo
  - C2: año múltiplo de 4.
    - b1: True
    - b2: False
  - C3: año múltiplo de 100.
    - b1: True
    - b2: False
  - C4: año múltiplo de 400.
    - b1: True
    - b2: False
- **Apartado 4.** Valores adecuados a cada bloque según criterios de cobertura:
  - C1: -200 | 0 | 2018
  - C2: 4 | 5
  - C3: 100 | 101
  - C4: 400 | 401

Ejercicio 2 - Conversión de números romanos a base 10

- **Apartado 1.** Se analizará la clase `RomanNumeral`, la cual contiene un método `convierte` que se encarga de transformar el *String* proporcionado en un número en base 10 (int).
- **Apartado 2.** Tenemos un único parámetro `s` de tipo *String*, que será el número romano que queremos convertir a base diez.
- **Apartado 3.** Tendremos 3 caracterizaciones:

    - C1: null (string vacío).
    - C2: string que sea romano.
    - C3: string que NO sea romano.
- **Apartado 4.** Valores adecuados a cada bloque según criterios de cobertura:
    - C1: string vacío | XVII | HJK / MMJ / IIIII

Ejercicio 3 - Embotelladora

- **Apartado 1.** Se analizará la clase `Embotelladora`, la cual contiene un método llamado `calculaBotellasPequenas`, que debe devolver el número de botellas pequeñas, medianas y grandes necesarias para embotellar un número total de litros, minimizando el número de botellas grandes.
- **Apartado 2.** Hay 3 parámetros:
    - `pequeñas`: tipo int. Será el número de botellas de 1 litro disponibles en el almacén.
    - `grande`: tipo int. Será el número de botellas de 5 litros disponibles en el almacén.
    - `total`: tipo int. Número total de litros que queremos embotellar.
- **Apartado 3.** Tendremos en esta ocasión 3 caracterizaciones:
    - C1: Cantidad de botellas.
        - b1: ninguna (pequeñas y grandes son 0).
        - b2: solo pequeñas disponibles.
        - b3: solo grandes disponibles.
        - b4: ambas disponibles.
        - b5: pequeñas negativas.
        - b6: grandes negativas.
    - C2: Mayor cantidad de botellas.
        - b1: misma cantidad pequeñas que grandes
        - b2: mayor cantidad de pequeñas que de grandes
        - b3: mayor cantidad de grandes que de pequeñas
    - C3: Litros totales con respecto al 0.
        - b1: litros negativos
        - b2: litros 0
        - b3: litros positivos
    - C4: Abastecimiento dado litros totales y botellas
        - b1: abastecemos con botellas justas
        - b2: no abastecemos
        - b3: abastecemos y nos sobran botellas
- **Apartado 4.**
    - C1: 0,0,10 | 10,0,10 | 0,10,10 | 1,2,10 | -1,2,10 | 1,-2,10
    - C2: 3,3,8 | 3,1,8 | 1,3,8
    - C3: 10,0,-10 | 10,0,0 | 10,10,10
    - C4: 0,2,10 | 2,0,10 | 5,2,13

Ejercicio 4 - Descuento Black Friday

- **Apartado 1.** Se analizará la clase `DescuentoBlackFriday`, la cual contiene un método llamado `PrecioFinal`, que devuelve un precio final un 30% menor que el precio original proporcionado, en el caso de que el día actual sea 23 de noviembre.
- **Apartado 2.** Un único parámetro `precioOriginal` de tipo *double*. Será el precio original de un producto marcado en su etiqueta, sin descuento.
- **Apartado 3.** Hay 1 caracterización:
  - C1: precio original respecto del 0:
    - b1: precio negativo
    - b2: precio 0
    - b3: precio positivo
- **Apartado 4.** Da valores adecuados a cada bloque según criterios de cobertura:
  - C1: -2.0 | 0.0 | 25.50