**Ejercicios Finales**

1. Escribe una clase, de la que no se podrán crear objetos ni puede ser heredada, llamada **Cadenas** que disponga de los siguientes métodos: (el control de excepciones se realizará en cada método)
   * **cadenaVector**. Recibe una cadena de caracteres y la devuelve como un array de caracteres. No utilices el método de la clase String que convierte una cadena a un vector.
   * **vectorCadena**. Recibe un array de caracteres y la devuelve como una cadena de caracteres.
   * **cantidadVocales**. Recibe una cadena de caracteres y devuelve el numero de vocales que tiene.
   * **subCadena**. Recibe dos valores enteros y una cadena de caracteres. Devuelve la sub-cadena desde la posición que indica el primer entero hasta la posición que indica el segundo entero. Mediante control de excepciones, si alguno de los enteros no pertenece al rango de la cadena, se devuelve un NULL.
   * **subVector**. Recibe dos valores enteros, un array de caracteres y un carácter. Devuelve el array como una cadena de caracteres en la que todas las posiciones que hay desde la posición indicada por el primer carácter hasta la posición indicada por el segundo carácter se han sustituido por el carácter que se pasa como parámetro.
   * Sobrecarga el método anterior. El método recibe un valor entero, una cadena de caracteres y un carácter, la sustitución se hará desde la posición que indica el valor entero hasta el final de la cadena.
   * **haySubcadena**. Recibe dos cadenas de caracteres. Si la segunda cadena es una sub-cadena de la primera devuelve true, en otro caso devuelve false.
   * **quitaSubcadena**. Recibe dos cadenas de caracteres. Si la segunda cadena es una sub-cadena de la primera, devuelve la primera cadena eliminando todas las sub-cadenas que sean como la segunda (la primera cadena se debe conservar).
2. Escribe un programa llamado **PruebaCadenas** que muestre un menú de opciones. Las opciones permitirán ejecutar cada uno de los métodos de la clase **Cadena**. El programa prueba debe estar en un fichero separado del fichero donde se encuentra la clase Cadena, pero en el mismo proyecro, y seguirá ejecutándose hasta que se indique que se quiere terminar. Realiza control de excepciones en los lugares que sea necesario.
3. Se quiere diseñar una clase llamada producto que represente los diferentes productos que se pueden encontrar en un centro comercial. Los productos tienen las siguientes características comunes:
   * **codigo.** De tipo entero. La clase dispone de un atributo que inicialmente tendrá el valor 00001 y se irá incrementando cada vez que se añada un producto. Cuando se crea un producto, su atributo nombre tomará el valor que tenga esa variable y después la incrementará.
   * **precio**. Número decimal
   * **nombre.** De tipo cadena de caracteres
   * Constructor que recibe el nombre y precio del producto
   * Un método que permite cambiar el nombre del producto
   * Un método que permite cambiar el precio del producto
   * Los métodos que permiten obtener el nombre y obtener el precio del producto
   * Un método que permite obtener el precio final del producto. Este método no se codifica en esta clase, se hace en las clases que heredan.
   * Un método que permite obtener la información del producto. Este método también se codificará en las clases que heredan de esta clase.
4. Implementa la clase **ProductoAlimentación**, que hereda de la clase **Producto**, y que tienen las siguientes características propias:
   * Una constante de tipo carácter que identifica el producto como de alimentación. Su valor es **A**
   * Una constante que identifica el IVA aplicado al producto. Su valor es 5%
   * Un atributo que indica el año de caducidad
   * Un atributo que indica el año de caducidad
   * Un atributo que indica el año de caducidad
   * Un constructor que recibe el código del producto, el nombre, el precio y la fecha de caducidad del producto
   * Los métodos que permiten cambiar la fecha de caducidad del producto
   * El método que permite consultar el año de caducidad del producto
   * Para calcular el precio final del producto, al precio base hay que añadir el IVA
5. Implementa la clase **ProductoElectrónica**, que hereda de la clase **Producto**, y que tienen las siguientes características propias:
   * Una constante de tipo carácter que identifica el producto como de electrónica. Su valor es **E**
   * Una constante que identifica el IVA aplicado al producto. Su valor es 15%
   * Un atributo que indica los días de garantía
   * Un método que reciba el código del producto, el nombre, el precio y los días de garantía del producto
   * Un método que permita cambiar los días de garantía del producto
   * Un método que permita consultar los días de garantía del producto
   * Para calcular el precio final del producto, al precio base hay que añadir el IVA
6. Implementa la clase **ProductoRopa**, que hereda de la clase **Producto**, y que tienen las siguientes características propias:
   * Una constante de tipo carácter que identifica el producto como de ropa. Su valor es **R**
   * Una constante que identifica el IVA aplicado al producto. Su valor es 15%
   * Un atributo que indica la composición de la prenda
   * Un método que reciba el código del producto, el nombre, el precio y la composición del producto
   * Un método que permita cambiar la composición del producto
   * Un método que permita consultar la composición del producto
   * Para calcular el precio final del producto, al precio base hay que añadir el IVA
7. Utilizando las clases de los ejercicios anteriores, crea una clase que permita crear una cesta de la compra e imprimir un ticket con los productos de la cesta, el precio de cada producto y el precio total de la compra. La clase tiene las siguientes características:
   * Un atributo llamado **cesta** dónde se irán almacenando los productos de la compra
   * **aniadeProducto(Producto).** Añade a la cesta el producto que se recibe como parámetro
   * **generaTicket().** Genera el ticket de la cesta. Por cada producto se imprimirá:
     1. El código del producto junto con la letra que identifica el tipo del producto.
     2. El nombre del producto
     3. El precio base
     4. El tipo de IVA aplicado
     5. El precio final
     6. El precio total de la cesta.
     7. Mostrar la información de los productos de la cesta
     8. Cada vez que se imprime un elemento, se elimina de la cesta
8. Escribe un programa que genere varios objetos de tipo Producto y los añada a la cesta. Después mostrará la información de todos los productos de la cesta y generará el ticket.