

- 7.11. Escribe la clase `MarcaPagina`, que ayuda a llevar el control de la lectura de un libro. Deberá disponer de métodos para incrementar la página leída, para obtener información de la última página que se ha leído y para comenzar desde el principio una nueva lectura del mismo libro.
- 7.12. Implementa una clase que permita resolver ecuaciones de segundo grado. Los coeficientes pueden indicarse en el constructor y modificarse a *posteriori*. Es fundamental que la clase disponga de un método que devuelva las distintas soluciones y de un método que nos informe si el discriminante es positivo.
- 7.13. En el momento de decorar una casa, una habitación o cualquier objeto, se plantea el problema de elegir la paleta de colores que vamos a utilizar en nuestra decoración. Existe una solución, algo atrevida, que consiste en utilizar colores al azar.
Diseña la clase `Colores`, que alberga por defecto una serie de colores (mediante una cadena), aunque es posible añadir tantos como necesitemos. La clase tendrá un método que devuelve una tabla con los n colores que necesitemos elegidos al azar sin repeticiones.
- 7.14. Crea una clase que sea capaz de mostrar el importe de un cambio, por ejemplo, al realizar una compra, con el menor número de monedas y billetes posibles.
- 7.15. 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 incrementares ()`: que incrementa en un mes la fecha del calendario.
 - `void incrementarAño (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.
- 7.16. 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 despiazaX (double dx)`: incrementa el componente x en la cantidad dx .
 - `void despiazaY (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 distanciaEuciidea (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.
- El cifrado César es una forma sencilla de modificar un texto para que no sea entendiéndolo a quienes no conocen el código. Este cifrado consiste en modificar cada letra de un texto por otra que se encuentra en el alfabeto n posiciones detrás.

- 7.17. Por ejemplo, para un valor de n igual a 3, la letra a se codifica con la d , y la letra q se codifica con la x . En el caso de que una letra exceda a la z , seguiremos de forma circular utilizando la a . Solo se cifrarán las letras, mayúsculas o minúsculas.
- Realiza una clase que, mediante un método estático, devuelva cifrado el texto que se le pasa con un paso de n letras.
- 7.18. Una cola es otra estructura dinámica como la pila, donde los elementos, en vez de apilarse y desapilarse, se encolan y desencolan. La diferencia con las pilas es que se desencola el primer elemento encolado, ya que así es como funcionan las colas del autobús o del cine. El primero que llega es el primero que sale de la cola (vamos a suponer que nadie se cuela). Por tanto, los elementos se encolan y desencolan en extremos opuestos de la estructura, llamados *primero* (el que está primero y será el próximo en abandonar la cola) y *último* (el que llegó último). Implementa la clase Cola donde los elementos `integer` encolados se guardan en una tabla.
- 7.19. Implementa la clase **Pila** para números **integer**, usando directamente una tabla para guardar los elementos apilados.
- 7.20. Repite la Actividad de aplicación 7.18, usando una `Lista` para guardar los elementos encolados.
- 7.21. 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 **IOS 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.