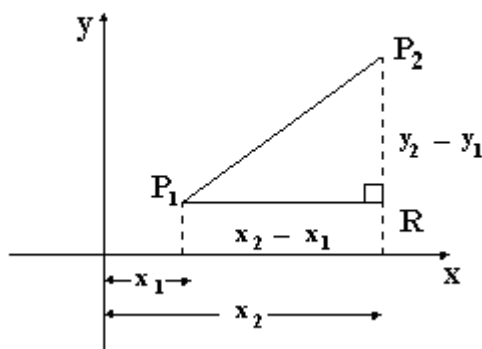


EJERCICIOS PRÁCTICOS TEMA 4

1. Queremos crear objetos que representen puntos. De estos objetos, necesitamos:
 - a. Que tengan dos propiedades **x** e **y**. Las coordenadas del punto. Estas coordenadas se pasan por parámetro y si no se pasa nada por defecto será 0.
 - b. Un método llamado **cambiar** al que le pasemos 2 números y nos permite modificar las coordenadas del punto.
 - c. Un método llamado **copia** que devuelve una copia del objeto.
 - d. Un método llamado **iguales** que recibe otro punto como parámetro y nos dice si son iguales o no.
 - e. Un método llamado **suma** que recibe un segundo punto y devuelve un tercer punto cuyas coordenadas son el resultado de sumar las dos anteriores.
 - f. Un método llamado **obtenerDistancia** que también recibe un punto y nos devuelve la distancia entre ambos, teniendo en cuenta lo siguiente:



- g. Un método **toString** que devuelve un texto con las coordenadas del punto. Por ejemplo (-8, 3).
2. Añadir un método al prototipo de los arrays, que devuelva la media aritmética de los elementos de un array. Pruébalo con un par de ejemplos.
3. Crear un objeto para fracciones y añadir métodos en la clase Math para calcular el **mcm** y el **MCD**. Para ello debemos tener en cuenta lo siguiente:
 - a. El máximo común divisor de 2 números naturales es el mayor número por el que podemos dividir de forma exacta ambos números. El mínimo común múltiplo es el número múltiplo más pequeño de ambos números a la vez. Para ambos cálculos podemos usar estos detalles que nos van a facilitar la implementación de ambos métodos:
 - i. $mcm(a,b)=a*b/mcd(a,b)$
 - ii. $mcd(a,b)=b$ si el resto de a entre b es 0
 - iii. $mdc(a,b)=mcd(b,a\%b)$ si el resto de a entre b no es 0
 - b. Una fracción consta de un **numerador** y un **denominador**, ambos son números enteros. La idea es que cuando se usen estos objetos no se permita modificar directamente el numerador y el denominador sino que podremos hacerlo mediante métodos.
 - c. El método **getNumerador** obtendrá el valor del numerador, **getDenominador** obtendrá el denominador.
 - d. Los métodos **setNumerador** y **setDenominador** permitirán cambiar tanto el numerador como el denominador. Ambos requieren del nuevo valor.
 - e. El método **cambiarFraccion** permitirá cambiar a la vez numerador y denominador.
 - f. Implementar un método llamado **toString** que permita retornar la fracción en formato numerador/denominador (ejemplo 5/6).

Desarrollo Web en Entorno Cliente

- g. implementar un método llamado **simplificar** que permita simplificar la función. Eso se consigue calculando el mcd del numerador y el denominador, y dividiendo el numerador y el denominador por ese mcd.
 - h. implementar un método para la **suma** de fracciones que requiere del cálculo del mínimo común múltiplo de los denominadores.
 - i. Crear un método para la **resta** de fracciones. Se hace igual que la suma pero restando los numeradores.
 - j. Crear un método para la **multiplicación** de fracciones. Se multiplican los numeradores y los denominadores entre sí.
 - k. Crear un método para la **división** de fracciones. Se multiplica de forma cruzada el numerador y el denominador.
 - l. En todas las operaciones matemáticas, se debe de **simplificar** la fracción resultante.
4. Crea un tipo de objetos que sirvan para representar **Cartas**. estos objetos tendrán dos propiedades:
- a. **palo**: que será un número de 1 a cuatro (donde 1 significa oros, 2 espadas, 3 Bastos y 4 copas)
 - b. **valor**: un número del 1 al 10 (1=as, 8=sota, 9=caballo, 10=rey)

Los objetos de este tipo se construyen indicando el palo y el valor. Si hay fallos en los datos, se devuelve un objeto nulo en la creación.

Las cartas tendrán estos métodos:

- a. **darValor**: Que recibe un número de palo y un valor para la carta para, con ellos, modificar la carta. ante datos incorrectos no cambia nada en la carta.
- b. **toString**: Método habitual (y estándar) para devolver en forma de texto entendible el valor de la carta. Por ejemplo: As de Oros.

Además, habrá otro tipo de objeto: **Baraja**. La idea es que represente una baraja de cartas españolas.

Tendrá los siguientes detalles:

- a. La baraja la formarán 40 cartas. Para ello tendrá la propiedad **cartas** que será un array de 40 cartas.
- b. Al construir la baraja se rellenan las cartas en el siguiente orden: por palos y cada palo con las cartas del 1 al 10. No se podrá acceder directamente al array fuera de la función constructora.
- c. El método **barajar** permite barajar las cartas. Es decir, desordenarlas de forma aleatoria.
- d. El método **toString** permite obtener la baraja en forma de texto para saber cómo están ordenadas las cartas.