

1. A partir del código generado en el ejemplo explicado en teoría para calcular la puntuación:

a. Crea un método llamado `mostrarPosicionRankinPuntuaciones`, debe tener dos parámetros uno será el nombre del jugador y un segundo parámetro que será la posición en la tabla de puntuaciones. El método debe mostrar por consola el nombre del jugador junto con el mensaje " consiguió llegar a la posición ", la posición que obtuvo y " en el ranking de puntuaciones."

b. Crea un segundo método llamado `calcularPosicionRankingPuntuaciones`, debe tener un argumento que será la puntuación del jugador y devolverá la posición del ranking. Debe devolver un tipo de dato entero atendiendo a los siguiente:

Si la puntuación del jugador es > 1000 \Rightarrow return 1;

Si la puntuación del jugador es > 500 y < 1000 \Rightarrow return 2;

Si la puntuación del jugador es > 100 y < 500 \Rightarrow return 3;

En cualquier otro caso \Rightarrow return 4;

c. Utiliza ambos métodos en el programa principal y muestra los resultados con las siguientes puntuaciones: 1500, 900, 400 y 30.

d. ¿Qué posición del ranking alcanzaría un jugador si obtuviera una puntuación de 1000? ¿Y si obtuviera 500? Corrige el programa.

e. Intenta mejorar tú código, en el método `calcularPosicionRankingPuntuaciones`, de forma que sea más eficiente, ayúdate de la corrección del apartado anterior. También puedes eliminar el último else que devuelve la posición 4 del ranking y devolver directamente un 4.

f. Ahora vamos a volver a modificar el método `calcularPosicionRankingPuntuaciones`, de forma que veamos que hay muchas posibilidades para resolver un problema. En vez de realizar cuatro return en nuestro código, vamos a crearnos una variable entera llamada `posicion` y la vamos a inicializar a 4. Dentro de cada if modificaremos el valor de esta variable y al final del método tendremos solo un return con la variable `posicion`.

```
public class PuntuacionesOne {
    public static void main(String[] args) {
        Scanner entrada= new Scanner(System.in);
        System.out.println("Introduce nombre jugador:");
        String nombreJugador= entrada.nextLine();
        System.out.println("Introduce puntuación jugador:");
        int puntuacionJugador= entrada.nextInt();

        int posicionTablapuntos = calcularPosicionRankingPuntuaciones(puntuacionJugador);
        String jugadorPosicion = mostrarPosicionRankinPuntuaciones (nombreJugador,
posicionTablapuntos);
        System.out.println(jugadorPosicion);
    }

    public static String mostrarPosicionRankinPuntuaciones (String nombreJugador, int
```

```

posicionTablaPuntos){
    return nombreJugador+" consiguió llegar a la posición " + posicionTablaPuntos + " en el ranking
de puntuaciones.";
}

public static int calcularPosicionRankingPuntuaciones (int puntuacionJugador){
    if(puntuacionJugador>1000){
        return 1;
    }else if(puntuacionJugador>500 && puntuacionJugador<1000){
        return 2;
    }else if(puntuacionJugador>100 && puntuacionJugador<500){
        return 3;
    }else {
        return 4;
    }
}
}
}

```

2. Vamos a crear un programa que realiza conversiones de velocidad.

Para ello, crea una nueva clase Java llamada SpeedConverter.java. Dentro de la clase escribe un método llamado toMilesPerHour que tenga 1 parámetro de tipo double con el nombre kilometersPerHour.

Este método debe devolver el valor calculado redondeado de tipo long.

Si el parámetro kilometersPerHour es menor que 0, el método toMilesPerHour debe devolver -1 para indicar un valor no válido.

De lo contrario, si es positivo, calcula el valor de millas por hora, redondea y devuélvelo. Para redondear utiliza el método round de la librería Math: Math.round(valor);

Escribe otro método llamado printConversion con 1 parámetro de tipo double con el nombre kilometersPerHour. Este método no debe devolver nada (void) y deberá calcular las millas por hora utilizando el parámetro de kilómetros por hora.

Deberá imprimir un mensaje con el formato "XX km/h = YY mi/h", donde XX representa el valor de kilometersPerHour y YY representa el valor redondeado de convertir kilómetros por hora a millas por hora.

Si el parámetro kilometersPerHour es <0 imprime el mensaje "Valor inválido".

Ayuda: 1 milla por hora son 1,609 km por hora.

En el método printConversion has de llamar al método toMilePerHour.

Ejemplos de input/output

toMilesPerHour(1.5); -> debe devolver el valor 1
toMilesPerHour(10.25); -> debe devolver el valor 6
toMilesPerHour(-5,6); -> debe devolver el valor -1

toMilesPerHour(75.114); -> debe devolver el valor 47

printConversion(1.5) \approx imprime en la consola: 1.5 km/h = 1 mi/h

printConversion(10.25) \approx imprime en la consola: 10.25 km/h = 6 mi/h

printConversion(-5.6) \approx imprime en la consola: Valor inválido

printConversion(75.114) \approx imprime en la consola: 75.114 km/h = 47 mi/h

Si queremos probar los métodos necesitaremos crear el método main y llamar a los métodos desde el main.

Ejemplo:

```
public static void main(String[] args) {  
    toMilesPerHour(10.5);  
    printConversion(10.5);  
}
```

```
public class SpeedConverter {  
    public static void main(String[] args) {  
        Scanner entrada=new Scanner(System.in);  
        System.out.println("Vamos a convertir KM/h a MI/h. Introduce valor para Kilometros por hora:");  
        double kilometerPerHour= entrada.nextDouble();  
        long milesPerHourSup = toMilesPerHour(kilometerPerHour);  
        System.out.println(milesPerHourSup);  
        printConversion(kilometerPerHour);  
    }  
  
    public static long toMilesPerHour(double kilometersPerHour) {  
        long milesPerHour = 0;  
        if (kilometersPerHour < 0) {  
            milesPerHour = -1;  
        } else {  
            milesPerHour = Math.round(kilometersPerHour / 1.609);  
        }  
        return milesPerHour;  
    }  
  
    public static void printConversion(double kilometersPerHour){  
        if(kilometersPerHour<0){  
            System.out.println("Valor inválido");  
        }else {  
            System.out.println(Math.round(kilometersPerHour)+"km/h = "  
+toMilesPerHour(kilometersPerHour)+"mi/h");  
        }  
    }  
}
```

3. Tenemos un perro al que le gusta ladrar. ¡Necesitamos despertarnos si el perro ladra por la noche!

Escribe un método llamado `hayQueDespertarse` que tenga 2 parámetros. El primer parámetro debe ser de tipo booleano y debe llamarse `ladrido`, representa si nuestro perro está ladrando o no.

El segundo parámetro representa la hora del día y es de tipo `int` con el nombre `horaDelDia` y tiene un rango válido de 0-23.

Tenemos que despertarnos si el perro está ladrando antes de las 8 o después de las 22 horas, en ese caso el método devuelve verdadero, en todos los demás casos devuelve falso.

Si el parámetro `horaDelDia` es menor a 0 o mayor a 23 devuelve `false`.

Ayuda. Utiliza la instrucción `if else` con múltiples condiciones.

El método `hayQueDespertarse` debe definirse como `public static` como lo hemos estado haciendo hasta ahora.

Ejemplos input/output

`hayQueDespertarse(true, 1)` → debe devolver `true`

`hayQueDespertarse(false, 2)` → debe devolver `false`, ya que el perro no está ladrando

`hayQueDespertarse(true, 8)` → debe devolver `false`, ya que no es antes de las 8

`hayQueDespertarse(true, -1)` → debe devolver `false`, ya que `horaDelDia` va desde las 0-23.

```
public class DespertarsePerro {
    public static void main(String[] args) {

        System.out.println(hayQueDespertarse(true,2));
    }
    public static boolean hayQueDespertarse(boolean ladrido, int horaDelDia){
        if(horaDelDia<0 || horaDelDia>23){
            return false;
        }else if ((horaDelDia<8 || horaDelDia>22) && ladrido==true){
            return true;
        }else {
            return false;
        }
    }
}
```

4. Diseña e implementa un programa que realice otro tipo de conversión que tú elijas como hicimos en el ejercicio 2.

```
public class PulgadasACm {  
    public static void main(String[] args) {  
        System.out.println(pulgadasToCm(2) + " Cm");  
    }  
    public static double pulgadasToCm(double entradapulgadas) {  
        double inchesToCm = 0;  
        if (entradapulgadas < 0) {  
            return inchesToCm = -1;  
        } else {  
            inchesToCm = entradapulgadas * 2.54;  
            return inchesToCm;  
        }  
    }  
}
```