

UD2Tarea10 - Métodos con if

Andrés Pérez Guardiola - 1st DAW

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

El Código de clase:

```
public class Main {  
  
    public static void main(String[] args) {  
        boolean gameOver = true;  
        int puntuacion = 5000;  
        int nivelCompletado = 5;  
        int bonus = 10;  
  
        int score = calcularPuntuacion(gameOver, puntuacion,  
nivelCompletado, bonus);  
        System.out.println(score);  
  
        //otra forma de hacerlo es pasarle directamente el valor de  
las variables  
        score = calcularPuntuacion(true, 1000, 10, 30);  
        System.out.println(score);  
  
    }  
  
    public static int calcularPuntuacion(boolean gameOver, int  
puntuacion, int nivelCompletado, int bonus) {  
        int puntuacionFinal = -1;  
        if (gameOver) {  
            puntuacionFinal = puntuacion + (nivelCompletado *  
bonus);  
            puntuacionFinal += 100;  
        }  
        return puntuacionFinal;  
    }  
}
```

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.”

```
public static void mostrarPosicionRankinPuntuaciones(String
nombre, int posicion){
    System.out.println(nombre + " consiguió llegar a la posición "
+ posicion + " en el ranking de puntuaciones.");
}
```

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files
5150
1400
Andres consiguió llegar a la posición 3 en el ranking de puntuaciones.

Process finished with exit code 0
```

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;

```
public static int calcularPosicionRankingPuntuaciones(int
puntuacion){
    if(puntuacion >= 1000) return 1;
    else if(puntuacion >= 500) return 2;
    else if(puntuacion >= 100) return 3;
    else return 4;
}
```

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

```
mostrarPosicionRankinPuntuaciones("Andres",  
    calcularPosicionRankingPuntuaciones(1500));  
mostrarPosicionRankinPuntuaciones("Sara",  
    calcularPosicionRankingPuntuaciones(900));  
mostrarPosicionRankinPuntuaciones("Michel",  
    calcularPosicionRankingPuntuaciones(400));  
mostrarPosicionRankinPuntuaciones("Alvaro",  
    calcularPosicionRankingPuntuaciones(30));
```

```
Andres consiguió llegar a la posición 1 en el ranking de puntuaciones.  
Sara consiguió llegar a la posición 2 en el ranking de puntuaciones.  
Michel consiguió llegar a la posición 3 en el ranking de puntuaciones.  
Alvaro consiguió llegar a la posición 4 en el ranking de puntuaciones.
```

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

1º y 2º porque en mi caso he usado >= de primeras. Nada que corregir.

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.

No sé a qué se refiere, pero tengo esto con respecto a lo de devolver 4 directamente:

```
public static int calcularPosicionRankingPuntuaciones(int  
puntuacion) {  
    if(puntuacion >= 1000) return 1;  
    else if(puntuacion >= 500) return 2;  
    else if(puntuacion >= 100) return 3;  
    return 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 static int calcularPosicionRankingPuntuaciones(int
puntuacion){
    int position = 4;
    if(puntuacion >= 1000) position = 1;
    else if(puntuacion >= 500) position = 2;
    else if(puntuacion >= 100) position = 3;
    return position;
}
```

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)` → imprime en la consola: 1.5 km/h = 1 mi/h

`printConversion(10.25)` → imprime en la consola: 10.25 km/h = 6 mi/h

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

printConversion(75.114) → 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.

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

(Ayuda con el enunciado ...)

Resumen del enunciado:

- Crear una clase SpeedConverter
- Método: toMilesPerHour()
 - Recibe parámetro kilometersPerHour tipo double
 - Devuelve un `long`?
 - Si recibe aceptar negativos, devuelve -1
 - Usar `Math.round(valor)` (Si que devuelve long)
- Método printConversion()
 - Recibe un parámetro kilometersPerHour tipo double
 - No devuelve nada (void)
 - Pinta en pantalla "XX km/h = YY mi/h"
 - (Usa el método toMilesPerHour() internamente)
 - Si kilometersPerHour es negativo imprime "Valor inválido"

`factorConversion = 1 / 1.609d;`

```

public class SpeedConverter
{
    public static void main(String[] args) {
        printConversion(10.5);
    }

    public static long toMilesPerHour(double kilometersPerHour)
    {
        if(kilometersPerHour < 0.0d) {
            return -1L;
        }
        double factorConversion = 1 / 1.609d;
        double milesPerHour = kilometersPerHour * factorConversion;
        return Math.round(milesPerHour);
    }

    public static void printConversion(double kilometersPerHour)
    {
        if (kilometersPerHour < 0.0d)
        {
            System.out.println("Valor inválido");
        } else {
            long milesPerHour = toMilesPerHour(kilometersPerHour);
            System.out.println(kilometersPerHour + " km/h = " +
milesPerHour + " mi/h");
        }
    }
}

```

```

"C:\Program Files\Java\jdk-22\bin\java.exe
7
10.5 km/h = 7 mi/h

Process finished with exit code 0

```

```

"C:\Program Files\Java\jdk-22\bin\ja
Valor inválido

Process finished with exit code 0

```


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.`

Resumen del enunciado:

- Método hayQueDespertarse()
 - Parámetro 1: booleano ladrido
 - Parámetro 2: int horaDelDia
 - Devuelve booleano
 - if par2 < 0 o par > 23. return false; // fuera de límites
 - Else if par2 > 8 o par2 < 22. return true;
 - Else, return false;

```
public class Main
{
    public static void main(String[] args)
    {
        System.out.println(hayQueDespertarse(true, 2));
    }

    public static boolean hayQueDespertarse(boolean ladrido, int
horaDelDia)
    {
        if (ladrido)
        {
            if (horaDelDia < 0 || horaDelDia > 23) return false;
            if (horaDelDia <= 8 || horaDelDia >= 22) return true;
        }
        return false;
    }
}
```

Si ladrido true y hora del dia 2:

```
"C:\Program Files\Java\jdk-22\bin\java.exe
true

Process finished with exit code 0
```

Si ladrido true y hora del dia -1:

```
"C:\Program Files\Java\jdk-22\bin\java
false

Process finished with exit code 0
```

Si ladrido true y hora del dia 23:

```
"C:\Program Files\Java\jdk-22\bin\  
true  
  
Process finished with exit code 0
```

Si ladrido true y hora del día 24:

```
"C:\Program Files\Java\jdk-22\bin\  
false  
  
Process finished with exit code 0
```

Si ladrido false (en todos los caso):

```
"C:\Program Files\Java\jdk-22\bin\  
false  
  
Process finished with exit code 0
```

