

CONTRARRELOJ CICLISTA

Vamos a escribir un programa para llevar la clasificación y otras tareas de una contrarreloj ciclista (aunque serviría para cualquier otra carrera contrarreloj).

Primero, por si alguno/a no sabe lo que es una contrarreloj aclarar que es una carrera en la que no salen todos los corredores a la vez sino que van saliendo de uno en uno y que cuando cada uno llega a la meta se anota el tiempo que hizo y se pone en una clasificación provisional. Dicha clasificación va cambiando cada vez que llega un nuevo corredor, hasta que llega el último. Si no está claro buscar a alguien que os lo explique mejor :-)

Entonces queremos un programa que nos presente un menú con varias opciones

- 1 - Mostrar la clasificación
- 3 - Llegada de un corredor
- 5 - Mostrar información de un corredor
- 7 - Mostrar la lista de corredores
- 8 - Mostrar los equipos participantes
- 9- Mostrar información de un equipo
- 99- Salir del programa

Es posible que más adelante se añadan otras opciones como mostrar la clasificación por equipos, por ejemplo.

1_Mostrar la clasificación : lo que tiene que hacer es mostrar todos los corredores que ya llegaron a la meta , ordenados por tiempo (de menos a más), con su dorsal, su nombre, su equipo.

3_ Llegada de un corredor : es anotar el dorsal, el tiempo (en principio vamos a manejar los tiempos en segundos) y una vez anotados esos valores , ponerlo donde le corresponda en la clasificación provisional.

5_Mostrar información de un corredor: Sería teclear un dorsal y que nos informe de sus datos en la carrera: nombre, equipo + en que puesto está (si es que ya llegó) qué tiempo hizo y a cuanto está del primero.

7_Mostrar la lista de corredores: es mostrar todos los datos de los corredores, ordenadamente por dorsal: dorsal, nombre, equipo, nacionalidad (y si ganó otras carreras u otra información).

8_Mostrar los equipos participantes: pues sería el nombre del equipo y su abreviatura (tres letras).

9_ Mostrar la información de un equipo: sería mostrar todos los corredores de un equipo. Esto por ahora lo dejamos abierto, sin concretar.

Como ya sabéis, entender el enunciado forma parte del problema. Así que volver a leer lo anterior; el **qué** queremos hacer .

~ ~ ~ * ~ ~ ~

Ahora hay que empezar a pensar en el **cómo**, cómo implementarlo; es decir, qué estructuras de datos (qué arrays nos interesa tener) y que tipo de datos (qué objetos podemos crear). Luego todo será más fácil.

Si tuviéramos que llevar todo esto "a mano" que tablas tendríamos ?

Pues una tabla de clasificación (un array) donde la información sería: dorsal| tiempo - con eso solo nos llega- y todos los datos de los corredores los tenemos en otro(s) arrays que no varían nunca.

	DORSAL	TIEMPO
1	18	1800
2	9	1865
3	25	2000
.		
.		
.		.
NC		

Esta tabla de clasificación en principio va a estar "vacía" , (0 corredores llegaron) pero cada vez que llegue uno se anota en el lugar que corresponda. El array tendrá tantas celdas como corredores compitan, NCORREDORES.

No confundirse. Ése es un array **de una dimensión**, donde en cada celda vamos a anotar dos cosas (dorsal y tiempo) así que ahí nos aparece un objeto tendremos que darle un nombre: DORTI (vaya nombre, no ?)

Y tendremos un array de DORTIs

Por otro lado tendremos un array con los datos de los corredores inscritos: dorsal, nombre, equipo, nacionalidad, otras cosas.

También es un array de una dimensión donde en cada celda tendremos todos los campos (atributos) anteriores. Aparece un nuevo objeto que tiene todos esos datos, llamémosle CORREDOR.
Y tendremos un array de nceldas (NCORREDORES) de tipo CORREDOR.

Bien, para no perdernos vamos a empezar a escribir algo de pseudocódigo ...

```

INICIO
    // carga de los datos de los corredores

    leer_opcion <---  mostrar menú ()

    hacer mientras ( opcion < > 99)    // o !=99

        case (opcion)    //o switch (opcion) o según (opcion)

            1 : Mostrar_clasificación ( ??? )

            3 - Anotar_Corredor ( ??? )
                Clasificar_Corredor ( ????? )
                mostrar en que puesto provisional está

            5 - Mostrar_información_corredor (dorsal)

            7 - Mostrar_lista_corredores

            8 - Mostrarequipos_participantes

            9-  Mostrar_información_equipo

        fin_case

        leer_opcion <---  mostrar menú

    fin_hacer
FIN

```

El programa puede tener una opción oculta para poder hacer la inscripción inicial de los corredores, o podemos suponer que se leen los datos de un fichero (o base de datos) o podemos ponerlos en el código directamente y una vez cargados en el array Corredores ya se usan desde allí.

Vamos a tener dos clases, en la clase contrarreloj tendremos la main y ahí llamaremos a los métodos que van a estar en la clase carrera donde también definiremos los objetos.

Para empezar ya podéis escribir el código del método que muestra el menú.

Y también en el método main, probar ya el esquema general y ver si lee bien la opción, aunque no se procese nada.