



RETO 4 – PROGRAMACIÓN BÁSICA

CONTEXTO

Usted forma parte del comité de premiación de una carrera atlética realizada en beneficio de un refugio de animales. El comité necesita contar con una solución que le facilite a todos ustedes obtener algunos datos del desempeño de los corredores.

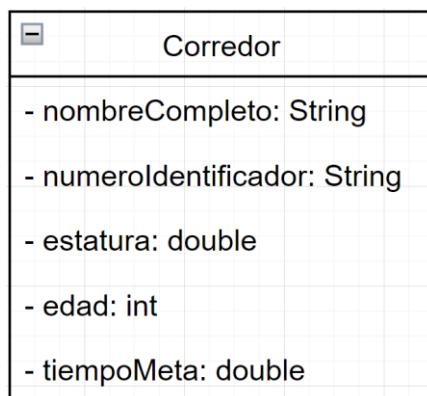
Su solución le permitirá conocer a los demás integrantes del comité la siguiente información:

- El tiempo promedio que toma a los corredores llegar a la meta.
- El nombre del corredor más rápido.
- El tiempo que le toma al corredor más rápido llegar a la meta.
- El nombre del corredor más lento.
- El tiempo que le toma al corredor más lento llegar a la meta.

TAREAS

Realizar un método en Java (Llamado `reporte`) que le permita al comité de premiación conocer datos sobre la carrera, teniendo en cuenta las siguientes especificaciones:

El conjunto de corredores se va a representar como un `ArrayList` de competidores, donde cada elemento es un objeto de la clase `Corredor` de la cual el equipo de ingeniería de software le hace entrega del siguiente diagrama:





Recuerde que los métodos relacionados al constructor, getters y setters son obviados en el diagrama de clases, pero deberán ser incluidos en el código; estos métodos deben ser creados con el estándar camel case, por ejemplo, si el atributo se llama `nombreCompleto`, sus métodos correspondientes a get y set serían `getNombreCompleto` y `setNombreCompleto`.

A partir de este `ArrayList`, usted deberá calcular el promedio de los tiempos que le toma a todos los corredores llegar a la meta (La suma de todos los tiempos a la meta de cada corredor dividida por la cantidad de corredores), determinar cuáles son los tiempos a la meta más corto y más largo y los corredores que poseen estos dos tiempos.

ENTRADAS

Su método recibirá como parámetro un `ArrayList` de instancias de la clase `Corredor` que representa el conjunto de competidores. NO CREE LOS DATOS DE ENTRADA, DEBE USAR LOS QUE SE RECIBEN COMO PARÁMETRO.

SALIDAS

Su método debe retornar un `Array` de `Object` donde:

- En la primera posición se guardará el promedio de los tiempos a la meta.
- En la segunda posición irá el nombre del corredor con menor tiempo a la meta.
- En la tercera posición irá el menor tiempo a la meta.
- En la cuarta posición irá el nombre del corredor con mayor tiempo a la meta.
- En la quinta posición irá el mayor tiempo a la meta.





EJEMPLO

Suponga que se le entrega el siguiente Array:

```
1. public static void main(String[] args) {  
2.     ArrayList<Corredor> carrera = new ArrayList<>();  
3.     carrera.add(new Corredor("Luis Gomez", "12", 1.68, 20, 22));  
4.     carrera.add(new Corredor("Juan Corredor", "28", 1.75, 24, 21));  
5.     carrera.add(new Corredor("Carlos Perez", "19", 1.71, 28, 24));  
6.     carrera.add(new Corredor("Maria Cadavid", "13", 1.80, 32, 22));  
7.     Object retorno[] = reporte(carrera);  
8.     System.out.println(Arrays.toString(retorno));  
9. }
```

En la línea 2 del main de prueba, se declaró el ArrayList, en las líneas 3 a la 6 se añaden 4 corredores a la carrera, cada uno con sus respectivos atributos, en la línea 7 se invoca el método que usted desarrolló y al imprimirlo en la línea 8 se obtiene:

```
[22.25, Juan Corredor, 21.0, Carlos Perez, 24.0]
```

NOTA ACLARATORIA

Usted podrá desarrollar la prueba en Netbeans. Al final debe copiar y pegar el código en la herramienta VPL, pero **NO** deberá subir archivos, es decir:

Modo incorrecto:

NO SUBIR NINGÚN ARCHIVO

Descripción Entrega Editar Ver entrega

Entrega

Comentarios

Seleccione un archivo... Tamaño máximo para archivos nuevos: 5MB

solucion.py

Puede arrastrar y soltar archivos aquí para añadirlos

Enviar Cancelar





Modo correcto:



¡MUCHOS ÉXITOS EN EL DESARROLLO DEL RETO 4 TRIPULANTE!

ACLARACIÓN DE PLAGIO

El objetivo es que los tripulantes cuenten con una oportunidad de aprendizaje relacionada con la programación. La colaboración académica es buena mientras no se lleve a un engaño académico, ya que el engaño académico inflige las buenas conductas del saber y del aprendizaje. El engaño académico hace referencia al plagio o envío de ideas que no son propias.

Colaborar implica compartir ideas, explicar a alguien cómo podría hacer su trabajo (más no hacer el trabajo por el otro) y ayudar al otro si tienes problemas a la hora de ejecutar o encontrar errores en el código.

En aras de evitar el plagio se recomienda colaborar, pero no compartir su código o proyecto, no compartir sus soluciones, no usar un código encontrado en internet u otras fuentes que las propias. (Mason, Gavrilovska, y Joyner, 2019)

Los ejercicios enviados a verificación deben cumplir con la política antiplagio. Es decir, cualquier envío que sea una copia textual de otro trabajo puede ser suspendido o no aprobado por parte del equipo evaluador. El acto de copiar





material de otro estudiante es un comportamiento inaceptable para el desarrollo de las competencias individuales y su progreso en este curso.

Referencia.

Mason, T., Gavrilovska, A., y Joyner, D.A. (2019). *Collaboration vesus cheating*. 50th ACM Technical Symposium on Computer Science Education SIGCSE 2019, Mineapolis, MN. DOI: 10.1145/3287324.3287443

