



OBJETIVOS

El objetivo del presente trabajo es la evaluación de los temas vistos hasta el momento en clase, la definición de clases con atributos y métodos (con pasaje de valores por parámetro y retorno cuando corresponda), la creación de objetos y uso de los mismos para simplificar la complejidad y la construcción de una interfaz de usuario para un primer programa y la utilización Vectores y Matrices.

ENUNCIADO: TENISTA EMPEDERNIDO.

Un jugador de tenis muy aplicado a su entrenamiento necesita un programa que le permita llevar un registro de las actividades asociadas al entrenamiento.

De las sesiones de entrenamiento registra siempre el día de la semana (número entero de 1 a 7) y la duración (número entero con la cantidad de minutos), además distingue los registros entre las sesiones en las que toma clases con un profesor y las que simplemente se encuentra con un amigo a jugar un partido.

De las clases registra el golpe trabajado específicamente (un número que representa: 1 – drive, 2 – revés, 3 – volea y 4 – saque)

Y por otro lado de los partidos registra el nombre del jugador al que enfrentó y si ganó o perdió el partido.

Se sabe además que el gasto calórico es de 55 KCal por cada 10 minutos de actividad si se trata de un partido y si se trata de una clase es de 45 KCal por cada 10 minutos si el golpe trabajado fue el drive o el revés y 40 si el trabajo fue sobre el saque o la volea.

Se necesita desarrollar un programa, que permita cargar sesiones y obtener resultados, que deberán funcionar como sigue:

EJERCICIO:

1. Cargar las Sesiones

- Debe solicitar los datos comunes de la sesión, si es una clase o un partido y los datos particulares en cada caso y luego grabar en el archivo dicha sesión.
- Tener en cuenta que al grabar las sesiones solo voy a grabar Clases o Partidos y nunca Sesiones en sí mismas porque no tendría sentido.

- Cada Sesión sea clase o partido debe ser cargada en una clase que contenga un vector de sesiones y que servirá para construir los comportamientos que se piden a continuación.

Una vez cargadas las sesiones en el vector y a partir del mismo resolver lo siguiente.

2. Calcular y mostrar el tiempo total en horas y minutos empleado en jugar al tenis.
3. Calcular y mostrar qué porcentaje del tiempo del punto anterior se empleó en partidos y qué porcentaje en clases.
4. Calcular el gasto total calórico generado en la práctica de tenis.
5. Generar una matriz con la cantidad de minutos que trabajó cada golpe en cada día de la semana, es decir una matriz con tantas filas como días de la semana (7 filas) y tantas columnas como golpes (4 columnas) y que en cada posición contiene el acumulado de las duraciones (nota: tener en cuenta que solo debo analizar las clases y no los partidos).
6. Generar un vector con el tiempo total acumulado por golpe, es decir de cada una de las clases la duración acumulada de acuerdo al número de golpe.
7. Porcentaje de partidos ganados frente a los partidos perdidos por día de la semana. (pista: Vectores paralelos)

Bonus

1. Mostrar el nombre del jugador al que le gané más veces. (pista: Conteo de Frecuencias y en este caso si bien se puede usar un vector auxiliar lo ideal sería usar una lista e ir agregando un nodo por cada nombre nuevo que aparezca en el conjunto y contar en ese nodo las victorias y derrotas para con ese contrincante.)

Pistas:

Para las sesiones deberemos construir una jerarquía de clases con la Clase Sesión como clase base y las clases Clase y Partido como derivadas.

Tener en cuenta el comportamiento especial que tendrá el método `caloríasConsumidas()` para poder ser utilizado en las diferentes situaciones donde haga falta.

Construir además la clase Estadística que contiene el vector de sesiones y los comportamientos para resolver las estadísticas solicitadas.

Luego recordar la división de responsabilidades y tratar de separar correctamente la interfaz de usuario del comportamiento de los objetos.

INSTRUCCIONES

1. Temas

- Desarrollo de clases: atributos y métodos, herencia y polimorfismo.
- Creación de instancias de una clase (objetos).

- Vectores de Objetos y/o tipos simples.
- Vectores y Matrices de conteo.

2. Condiciones

El trabajo puede ser realizado entre **4 (cuatro)** alumnos como **máximo y 3 (tres)** alumnos como mínimo.

La resolución consiste en un programa Java que responda a la consigna planteada en el enunciado. No solo deben ser correctos sus resultados, sino también el modelo de clases elegido para resolverlo y la lógica implementada en los métodos.

3. Forma de entrega

A partir de la publicación del enunciado, estará disponible en la Universidad Virtual una tarea identificada como Trabajo Practico Nro. 3.

Cada alumno **individualmente** deberá responder a esta tarea de la siguiente forma:

- Adjuntar un archivo que contenga el programa que desarrollaron (comprimir el proyecto completo en un solo archivo) y previo a la finalización ejecutar la compilación total del proyecto, es decir usar la opción "Clean & Build" de netbeans o "Limpiar y Construir" en castellano.
- El nombre del archivo comprimido debe estar compuesto de los apellidos de los integrantes del grupo separados por guiones (-).
- Agregar también un archivo de texto con los datos completos de los alumnos Legajo, Apellido y Nombre.
- Cada persona recibirá automáticamente la misma calificación **aunque haya enviado algo diferente y 0 si no realizó el envío con su usuario**. Es decir el docente corrige el primero que revisa del grupo.
Importante! Es responsabilidad de los alumnos revisar de ver la tarea en el aula virtual y si no pueden verla deberán avisar al JTP.

4. Evaluación

Además de los elementos mencionados en el Enunciado y en el Ejercicio, la nota de la evaluación seguirá el siguiente esquema. Resolviendo el trabajo práctico sin el punto bonus, la nota máxima será 9.

Para aspirar al 10 hace falta implementar el punto bonus y que la solución funcione correctamente y el diseño cumpla con los lineamientos impartidos en clase.