Proyecto UT3 (para hacer en casa y entregar en GitHub)

Objetivos *Saber:*

- *definir y utilizar constantes y atributos (variables de instancia) dentro de una clase*
- *definir constructores*
- *definir y utilizar parámetros en los constructores y métodos*
- *definir métodos accesores y mutadores*
- *usar las sentencias de asignación y escritura*
- *construir una sentencia if*
- *construir una sentencia switch*
- *expresar el algoritmo correspondiente a un método*
- *usar operadores aritméticos y relacionales*
- *usar métodos de la clase Math*

Antes de empezar

- Este ejercicio es para realizar de forma individual en casa.
- El proyecto de partida está en https://github.com/montsemsanz/ENTRE-UT3-Podometro. Deberás hacer un *fork* a tu cuenta y clonarlo en tu PC desde BlueJ tal y como se explicó en clase
- Una vez completado desde BlueJ haz un *push* del último *commit* a GitHub
- No olvides entregar vía Moodle el texto de la actividad "*Terminado proyecto UT3 Podómetro*" y pulsar *Enviar para calificar*
- Se valorará en la corrección que el programa esté probado (compila y ejecuta bien) y que esté claramente escrito y organizado (se respetan las reglas de estilo del lenguaje Java, nombres descriptivos, código no duplicado, ...)

- La fecha tope de entrega es el **Jueves 14 Octubre** hasta las **23,30h**.

- Se anulará automáticamente la corrección del ejercicio y se **evaluará con un 0** si se detecta que ha sido copiado o dejado copiar a algún compañero/a
- Se penalizará si no se siguen las normas de entrega del ejercicio
  - ✗ no se ha hecho un *fork* / no se sube vía *commit*
  - ✗ hay algún *commit* posterior a esta fecha de entrega
  - ✗ no se ha enviado el texto de la actividad vía Moodle
- El profesorado podrá convocar al alumno/a para defender oralmente el proyecto

Especificaciones

En este proyecto vamos a modelar mediante una clase un sencillo podómetro. El podómetro va a registrar información acerca de los pasos, distancia, tiempo caminado, etc... que una persona ha realizado en una semana.

Haz el *fork* del proyecto **ENTRE-UT3-Podometro** desde https://github.com/montsemsanz a tu cuenta GitHub y desde BlueJ clona el proyecto a tu PC.

Abre el proyecto BlueJ. Tienes que completar únicamente la clase `Podometro`. La clase `DemoPodometro` no tienes que modificarla, te servirá para probar la otra.

No olvides escribir tu nombre después de la etiqueta *@author*.

Define dentro de la clase `Podometro` las siguientes constantes y atributos (deduce los tipos de datos adecuados):

- dos constantes que indican el sexo de una persona
  - HOMBRE con el valor asociado 'H'
  - MUJER con el valor asociado 'M'
- dos constantes
  - ZANCADA_HOMBRE con el valor asociado 0.45
  - ZANCADA_MUJER con el valor asociado 0.41
  - estos valores representan un porcentaje (45% y 41% respectivamente) sobre la altura de la persona que se usarán para calcular la longitud de su zancada
- dos constantes que indican el nº de día de la semana
  - SABADO con el valor asociado 6
  - DOMINGO con el valor asociado 7

- los siguientes atributos o variables de instancia
  - *marca* - guarda la marca del podómetro
  - *altura* - guarda la altura de la persona en centímetros
  - *sexo* - guarda el sexo de una persona (un carácter)
  - *longitudZancada* - almacena la longitud de la zancada de la persona en centímetros

  - *totalPasosLaborables* – guarda el nº de pasos dados en días laborable (de lunes a viernes)
  - *totalPasosSabado* – guarda el nº de pasos dados el sábado
  - *totalPasosDomingo* – guarda el nº de pasos dados el domingo

  - *totalDistanciaSemana* – almacena la distancia recorrida a lo largo de toda la semana (en Kilómetros)
  - *totalDistanciaFinSemana* – almacena la distancia recorrida a lo largo del fin de semana (en Kilómetros)

  - *tiempo* – tiempo total caminado en toda la semana (en minutos)
  - *caminatasNoche* – nº caminatas (paseos) dados a partir de las 21h. En toda la semana

No debes incluir más atributos. Solo los indicados. Respeta los nombres que se te dan

Completa los siguientes métodos:

- el **constructor**, recibe un parámetro, el nombre de la marca del podómetro. Inicializa el resto de atributos a 0 salvo el sexo que se inicia con el valor por defecto MUJER.
- un accesor **getMarca()** para el nombre de la marca
- el método **void configurar(double queAltura, char queSexo)** – simula que se configura el podómetro. A partir de los parámetros indicados inicia los atributos *altura* y *sexo* y da un valor a la longitud de la zancada. Esta última se calcula en centímetros dependiendo del sexo de la persona: para hombres es el 45% de su altura siempre redondeando hacia arriba, para mujeres es el 41% de su altura siempre redondeando hacia abajo.
  Busca en la clase `Math` los métodos que redondean hacia arriba o hacia abajo.

- **public void registrarCaminata(int pasos, int dia, int horaInicio, int horaFin) {**

Este método recibe cuatro parámetros que supondremos correctos. Son los datos de una caminata:

- *pasos* – nº pasos caminados
- *dia* – nº día semana en que se ha hecho la caminata (2 si es martes, 5 si viernes, 7 si domingo,....)
- *horaInicio* – hora de inicio de la caminata. Es un valor entero de hasta 4 dígitos, los dos (o uno) más significativos representan la hora de inicio de la caminata y los dos menos significativos los minutos de inicio. *Ej.* el valor 1425 representa la hora 14:25, el valor 654 representa la hora 06:54, el valor 1005 representa la hora 10:05
- *horaFin* – hora de fin de la caminata. Con el mismo formato que la hora de inicio
- por simplicidad supondremos que una caminata empieza y acaba en el mismo día

Ej. *registarCaminata(2000, 3, 1730, 1815);* significa que se han dado 2000 pasos el miércoles desde las 17:30h. a las 18:15h.

registarCaminata(8500, 6, 1915, 2100); significa que se han dado 8500 pasos el sábado desde las 19:15h. a las 21:00h.

A partir de estos parámetros el método debe hacer cálculos para actualizar el estado del podómetro (sus atributos) adecuadamente:

• qué atributos hay que actualizar? *totalDistanciaSemana*, *tiempo*, *caminatasNoche*, etc....
• la distancia recorrida en la caminata son los pasos dados por la longitud de la zancada
• utiliza una sentencia **switch** para analizar el día
• evita repetir código

- el método **printConfiguración()** - muestra en pantalla la configuración del podómetro (altura, sexo y longitud de la zancada) (*ver resultados de ejecución*)

```
Configuración del podómetro
**********************************
Altura: 1.57 mtos
Sexo: MUJER
Longitud zancada: 0.64 mtos
```

- el método **printEstadisticas()** - muestra en pantalla información acerca de la distancia recorrida, pasos, tiempo total caminado, ... (*ver resultados de ejecución*)

```
Estadísticas
**********************************
Distancia recorrida toda la semana: 48.1152 Km
Distancia recorrida fin de semana: 20.48 Km

Nº pasos días laborables: 43180
Nº pasos SÁBADO: 17000
Nº pasos DOMINGO: 15000

Nº caminatas realizadas a partir de las 21h.: 3

Tiempo total caminado en la semana: 11h. y 48m.
Día/s con más pasos caminados: LABORABLES
```

- el método **String diaMayorNumeroPasos()** - devuelve el nombre del día en el que se ha caminado más pasos - "SÁBADO" "DOMINGO" o "LABORABLES" (puede haber coincidencias)

- el método **void reset()** - restablece los valores iniciales del podómetro. Todos los atributos se ponen a cero salvo el sexo que se establece a MUJER. La marca no varía.

Posible ejecución

Para probar que la clase Podometro funciona correctamente:

a) crea un objeto de la clase DemoPodometro

a) llama al método **iniciar()**

Tendrás que obtener los resultados de la figura:

```
*****************************************
*******    Podómetro SMARTWALK       *******
*****************************************


Configuración del podómetro
*********************************
Altura: 1.57 mtos
Sexo: MUJER
Longitud zancada: 0.64 mtos

Estadísticas
*********************************
Distancia recorrida toda la semana: 48.1152 Km
Distancia recorrida fin de semana: 20.48 Km

Nº pasos días laborables: 43180
Nº pasos SÁBADO: 17000
Nº pasos DOMINGO: 15000

Nº caminatas realizadas a partir de las 21h.: 3

Tiempo total caminado en la semana: 11h. y 48m.
Día/s con más pasos caminados: LABORABLES

Pulse <Intro> para continuar
```

```
*****************************************
*******    Podómetro SMARTWALK       *******
*****************************************


Configuración del podómetro
*********************************
Altura: 1.85 mtos
Sexo: HOMBRE
Longitud zancada: 0.84 mtos

Estadísticas
*********************************
Distancia recorrida toda la semana: 16.665599999999998 Km
Distancia recorrida fin de semana: 5.04 Km

Nº pasos días laborables: 13840
Nº pasos SÁBADO: 0
Nº pasos DOMINGO: 6000

Nº caminatas realizadas a partir de las 21h.: 2

Tiempo total caminado en la semana: 6h. y 35m.
Día/s con más pasos caminados: LABORABLES
```

```
*****************************************
*******    Podómetro SMARTWALK       *******
*****************************************


Configuración del podómetro
*********************************
Altura: 1.65 mtos
Sexo: MUJER
Longitud zancada: 0.67 mtos

Estadísticas
*********************************
Distancia recorrida toda la semana: 5.1255 Km
Distancia recorrida fin de semana: 3.1155 Km

Nº pasos días laborables: 3000
Nº pasos SÁBADO: 1650
Nº pasos DOMINGO: 3000

Nº caminatas realizadas a partir de las 21h.: 2

Tiempo total caminado en la semana: 5h. y 30m.
Día/s con más pasos caminados: LABORABLES DOMINGO
```

```
*****************************************
*******    Podómetro SMARTWALK       *******
*****************************************


Configuración del podómetro
*********************************
Altura: 1.9 mtos
Sexo: HOMBRE
Longitud zancada: 0.86 mtos

Estadísticas
*********************************
Distancia recorrida toda la semana: 5.16 Km
Distancia recorrida fin de semana: 3.44 Km

Nº pasos días laborables: 2000
Nº pasos SÁBADO: 2000
Nº pasos DOMINGO: 2000

Nº caminatas realizadas a partir de las 21h.: 0

Tiempo total caminado en la semana: 4h. y 45m.
Día/s con más pasos caminados:  LABORABLES  SABADO  DOMINGO
```

Rúbrica evaluación	
ctes / atributos	5,10
construtor	4,00
accesor	4,00
configurar	8,00
registrarCaminata	40,00
printConfiguracion	6,00
printEstadisticas	8,9
diaMayorNumeroPasos	15,00
reset	3,00
buen estilo programación	6
	100
Penalización (no compila)	-0,5 (sobre 10)