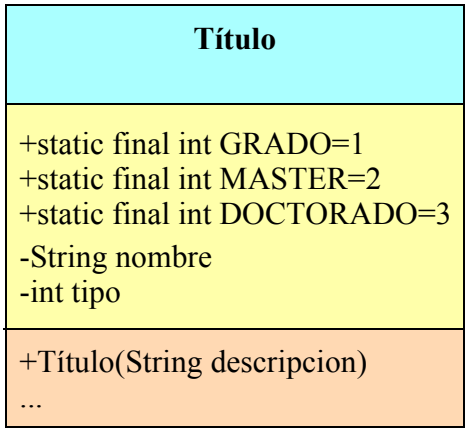
Examen de Programación
(Grados en Física y Matemáticas)

Junio 2015

Primera parte (1.25 puntos por cuestión, 50% nota del examen)

1) Se desea escribir el constructor de la clase Título que representa un título universitario y cuyo diagrama se muestra.



El constructor recibe como parámetro un texto que contiene el tipo de estudios (GRADO, MASTER o DOCTORADO) y el nombre del título. El tipo de estudios se define con el texto "Grado en", "Máster Universitario en" o "Doctorado en", respectivamente para cada uno de los tres tipos. Ejemplos:

```
Grado en Información y Documentación
Máster Universitario en Contabilidad y Finanzas
Doctorado en Ciencia y Tecnología
```

El constructor busca en el string el texto que describe el tipo de estudios y pone en el atributo tipo el número correspondiente. Además, guarda el nombre del título en el atributo nombre.

Para buscar un texto al principio de un String se dispone entre otros del método de la clase String startsWith(String str) que retorna un booleano indicando si el String comienza por el parámetro str. Para obtener el nombre se puede usar el método substring que permite obtener una parte de un String. Observar que la longitud del tipo de estudios es siempre fija para cada tipo. Por ejemplo, "Grado en" siempre tiene 8 caracteres y por tanto el nombre de un grado siempre comienza a partir del décimo carácter (quitando el espacio anterior al nombre).

Para los tres ejemplos que se muestran, el resultado del constructor será, respectivamente:

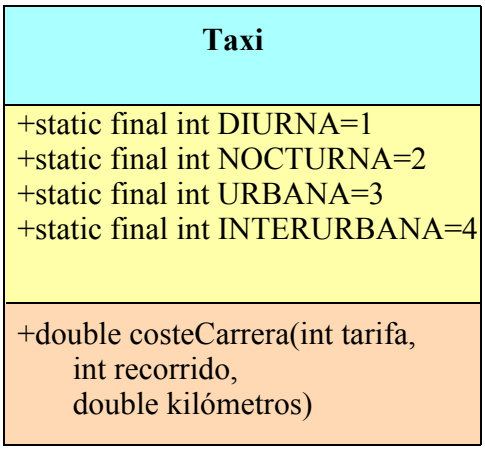
```
tipo=1  nombre="Información y Documentación"
tipo=2 nombre="Contabilidad y Finanzas"
tipo=3 nombre="Ciencia y Tecnología"
```

2) Escribir el *pseudocódigo* de un método iterativo que recibe como parámetro un String multilínea que contiene un programa Java en código fuente y que retorna un booleano indicando si hay llaves desparejadas o no.

Para hacer este método se puede crear un contador de llaves abiertas, que inicialmente es cero. Se hace un recorrido de todos los caracteres del String. Cuando se encuentra una llave izquierda '{' el contador de llaves abiertas se incrementa en uno. Cuando se encuentra una llave derecha '}' el contador de llaves abiertas se decrementa. Las llaves están desparejadas si encontramos una de estas dos situaciones:

- al final del método el contador de llaves abiertas no es cero
- en cualquier momento el contador de llaves abiertas es negativo

3) Se desea escribir el método costeCarrera para la clase Taxi cuyo diagrama se muestra en la figura. El método calcula y retorna el coste, en euros, de la carrera dada la tarifa (DIURNA o NOCTURNA), el recorrido (URBANO o INTERURBANO) y los kilómetros. El cálculo se hace según los valores de esta tabla, que vienen en euros:



Concepto	URBANA Diurna	URBANA Nocturna	INTERURBANA Diurna	INTERURBANA Nocturna
Bajada Bandera	1.35	1.75	1.35	1.65
Km. recorrido	0.92	1.20	0.60	0.70

Se valorará la eficiencia de la implementación.

4) En un computador con sistema operativo Linux se dispone de un directorio llamado carrera situado en el directorio del usuario y que dentro contiene 3 subdirectorios llamados primero, segundo y tercero.

Hacer un dibujo con un esquema de la estructura de directorios de partida.

Se desea escribir un *script* que se ejecute comenzando desde el directorio del usuario y que haga lo siguiente:

- crear en el directorio del usuario un nuevo directorio llamado definitivo y dentro de él otros dos llamados apuntes y programas
- copiar en apuntes todos los ficheros acabados en .docx que estén en primero, segundo y tercero
- mover a programas todos los ficheros acabados en .java que estén en primero, segundo y tercero
- borrar de tercero todos los ficheros que acaben en .class
- borrar los directorios primero y segundo con todos sus contenidos

Hacer también un dibujo con un esquema de la estructura de directorios resultante al final.

Examen de Programación
(Grados en Física y Matemáticas)

Junio 2015

Segunda parte (5 puntos, 50% nota del examen)

El control del consumo eléctrico puede ayudar a planificar los gastos de un hogar. Los electrodomésticos tienen un consumo fijo y, para los que calientan agua como la lavadora o el lavaplatos, uno variable que depende de la temperatura media del agua en cada mes del año. Por otro lado, las luces tienen un consumo que depende del número de horas de luz de cada mes del año.

Se desea hacer parte de un programa que permita simular los consumos eléctricos de un hogar a lo largo del año. Cada electrodoméstico se almacena en el programa mediante un objeto de la clase `Electrodoméstico` cuyo diagrama se muestra abajo y que ya está implementada. Sus atributos son:

- `nombre`: es un texto que describe el electrodoméstico
- `consumoMensualFijo`: consumo mensual independiente de la temperatura del agua o de las horas de luz, en kWh
- `consumoMensualVar`: consumo mensual variable que depende de la temperatura del agua o de las horas de luz, en kWh/grado o kW

Como puede observarse, se dispone de un constructor al que se le pasan los datos del electrodoméstico. También hay métodos observadores, uno para cada atributo. Por último está el método `consumoMensual()` al que se le pasa el `términoVariable` (temperatura del agua en grados u horas de luz) y que calcula el consumo mensual total en kWh, que es simplemente la suma del `consumoMensualFijo` y del producto del `consumoMensualVar` por el `términoVariable`:

Electrodoméstico
-String nombre -double consumoMensualFijo -double consumoMensualVar
+Electrodoméstico (String nombre, double consumoMensualFijo, double consumoMensualVar) +String nombre() +double consumoMensualFijo() +double consumoMensualVar() +double consumoMensual(double términoVariable)

Hogar
-ArrayList<Electrodoméstico> lista -double[] horasLuz -double[] temperaturaMensual
+Hogar(String nombreFichero) +double consumoAnual(Electrodoméstico ele) +void listadoElectrodoméstico(String nombre) throws NoEncontrado +double consumoAnualHogar()

Lo que se pide es implementar en Java la clase `Hogar` cuyo diagrama de clases se muestra arriba. La clase dispone de los siguientes atributos:

- lista: contiene una lista de objetos de la clase Electrodoméstico
- horasLuz: es un array de 12 casillas conteniendo las horas de luz en cada mes del año (la casilla 0 para enero, la 1 para febrero, ...)
- temperaturaMensual: es un array de 12 casillas conteniendo la temperatura media del agua del grifo en cada mes del año (igual que antes, enero en la casilla 0, ...)

Los métodos de la clase hacen lo siguiente:

- *constructor*: lee de un fichero de texto cuyo nombre se pasa como parámetro los datos de todos los electrodomésticos de la casa, crea un objeto de la clase Electrodoméstico con cada uno y añade estos objetos al atributo lista. El fichero contiene una primera línea de encabezamiento y luego un electrodoméstico en cada línea, con el formato que se muestra en este ejemplo:

```
Nombre      Consumo Fijo(kWh)   Consumo Variable (kWh/grado o kW)
Luces             0.0                    0.6
Lavadora          1.9                   -0.036
Lavaplatos        2.1                   -0.052
```

  Se puede suponer que el fichero es correcto.

  Además, el constructor crea y da valor a los atributos horasLuz y temperaturaMensual sacando los valores de esta tabla:

	ene	feb	mar	abr	may	jun	jul	ago	sep	oct	nov	dic
horas de luz	9.6	10.5	11.7	13.2	14.1	15.0	14.7	13.7	12.5	11.1	9.8	9.2
temperatura media (C)	13.1	12.8	12.9	13.0	15.1	16.5	20.0	21.9	20.2	16.8	16.0	14.1

- consumoAnual(): Calcula y retorna (en kWh) el consumo anual del electrodoméstico ele que se pasa como parámetro. Para ello se sigue el siguiente pseudocódigo:

```
método consumoAnual(Electrodoméstico ele) retorna real
   // la variable total acumula el consumo de cada mes
   real total=0
   // recorremos los meses del año
   para mes desde 0 hasta 11 hacer
      si nombre de ele = "Luces" entonces
         // para las luces el término variable son las horas de luz
         total=total+consumo mensual de ele con términoVariable=horasLuz[mes]
      si no
         // para el resto de electrodomésticos el término variable es la temperatura
         total=total+ consumo mensual de ele con términoVariable=
               temperaturaMensual[mes]
      fin si
   fin para
   retorna total
fin método
```

- listadoElectrodoméstico(): Busca el electrodoméstico cuyo nombre coincide con el parámetro nombre. Si lo encuentra hace un listado por pantalla del consumo de ese electrodoméstico en cada mes del año poniendo un encabezamiento con el nombre del electrodoméstico y luego un consumo mensual en cada línea indicando el número del mes (de 1 a 12) y el consumo en ese mes en kWh. El consumo mensual se obtiene con el método consumoMensual pasándole como término variable las horas de luz o la

temperatura mensual según el nombre del electrodoméstico sea "Luces" o no, similarmente a lo que se hace en el pseudocódigo de arriba.

Se valorará que el listado esté formateado en columnas de tamaño apropiado y con el consumo en formato de coma fija con tres decimales.

Si no se encuentra el electrodoméstico con el nombre indicado se lanza la excepción NoEncontrado.

- consumoAnualHogar(): Retorna el consumo total de todos los electrodomésticos del hogar en un año, en kWh. Para ello se hace un recorrido de la lista y para cada electrodoméstico se calcula su consumo con el método consumoAnual() y se suman todos.

La excepción NoEncontrado está ya creada en una clase aparte.

Finalmente, se pide hacer un programa principal en una clase aparte que haga lo siguiente:

a. Crea un objeto de la clase Hogar a partir del fichero de nombre "electrodomesticos.txt"

b. Muestra en pantalla el listado del electrodoméstico "Lavadora"

c. Muestra en pantalla el listado del electrodoméstico "Luces"

   Si en alguno de los pasos b) o c) se lanza NoEncontrado se muestra un mensaje de error y luego se sigue por el paso e), es decir, no se hace el d)

d. Muestra en pantalla el consumo anual del hogar

e. Para probar si la excepción NoEncontrado se lanza correctamente intenta mostrar en pantalla el listado del electrodoméstico inexistente "xxxx". En el manejador de la excepción se mostrará un mensaje de error.

Valoración:

- encabezamiento de la clase, atributos y constructor: 1.5 puntos
- consumoAnual: 0.5 puntos
- listadoElectrodoméstico: 1 punto
- consumoAnualHogar: 1 punto
- programa principal: 1 punto