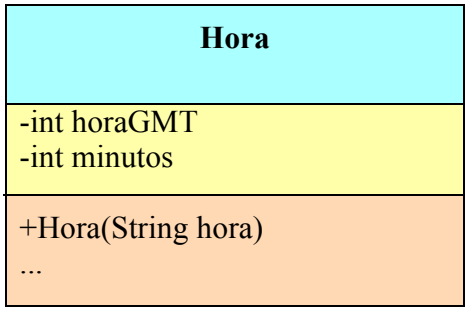
Examen de Programación
(Grados en Física y Matemáticas)

Junio 2014

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

1) Los objetos de la clase que se muestra en el diagrama contienen horas y minutos. Se pide escribir el constructor, que recibe un String con la hora, minutos y zona horaria en el formato de estos ejemplos:




```
13:56 GMT+3
 8:03 GMT-4
```

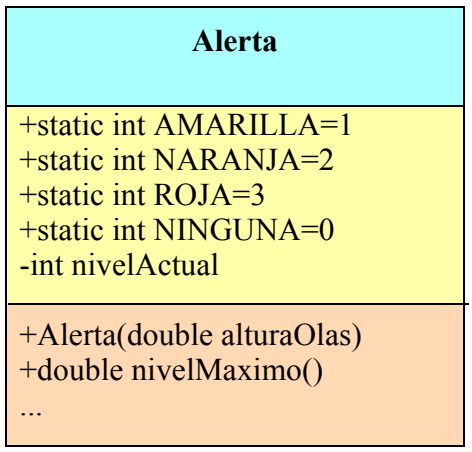
El constructor debe guardar:

- en el atributo minutos, los minutos
- en el atributo horaGMT, la hora sumada (con su signo) al número de la zona horaria (para los ejemplos de arriba, 13+3=16 y 8-4=4)

2) Escribir un método estático que lee líneas de un fichero de texto cuyo nombre se le pasa como parámetro. Cada línea contiene una hora, minutos y zona horaria en el formato de la cuestión 1. Tras leer cada línea el método debe crear un objeto de la clase Hora usando su constructor y añadírselo a un ArrayList que será retornado cuando ya se hayan leído todas las líneas del fichero. La cabecera del método debe ser:

```
public static ArrayList<Hora> leeHoras(String nombreFichero)
```

3) Se dispone de la clase Alerta con cuatro constantes enteras públicas que indican un nivel de alerta marítima que dependerá del oleaje previsto en el mar. El atributo nivelActual contendrá el nivel de alerta actual, que será uno de esos cuatro valores.



Se desea escribir el constructor, al que se le pasa la altura prevista del oleaje y debe calcular el nivel de alerta y guardarlo en nivelActual. El cálculo se hace según los datos de la tabla.

Por otro lado, escribir también el método nivelMaximo(), que retorna el valor máximo de las olas en función del nivelActual (3.0 para NINGUNA, 4.5 para AMARILLA, 8.0 para NARANJA e *infinito* para ROJA.)

Nivel de alerta	Altura de Olas
ROJA	> 8m
NARANJA	(4.5 , 8] m
AMARILLA	(3 , 4.5] m
NINGUNA	< =3m

4) Escribir el *pseudocódigo* de un método iterativo que calcula el siguiente desarrollo en serie de la función coseno:

$$\cos(x) \cong \sum_{i=0}^{n} \frac{(-1)^i}{(2i)!} x^{2i}$$

El método recibe como parámetros los valores x y n. Para hacer más eficiente el cálculo no utilizar las operaciones "elevar a" ni factorial. Observar que de la iteración i a la siguiente:

- el numerador cambia de signo alternando entre +1 y -1
- el denominador para la próxima iteración se calcula multiplicando al actual por $(2i+1)*(2i+2)$
- la potencia de x se obtiene multiplicando a la anterior por x^2

5) En un computador con sistema operativo Linux se dispone de un directorio llamado `datos` que está dentro del directorio del usuario. A su vez, dentro de `datos` hay varios subdirectorios cuyos nombres son años: `2010`, `2011`, `2013`, `2014`.

Se desea escribir un *script* que haga lo siguiente:

- crear en el directorio del usuario un nuevo directorio llamado `resumen` y dentro de él otros dos llamados `res2013` y `res2014`
- copiar en `res2013` todos los ficheros acabados en `.dat` que estén en `2013`
- mover a `res2014` todos los ficheros acabados en `.txt` que estén en `2014`
- borrar de `2013` todos los ficheros que acaben en `.dat` y de `2014` los que acaben en `.txt`
- borrar los directorios `2010` y `2011`con todos sus contenidos

Utilizar todas las órdenes que creas convenientes. Hacer también un dibujo con un esquema de la estructura de directorios resultante al final.

Examen de Programación
(Grados en Física y Matemáticas)

Junio 2014

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

Se desea escribir parte del software de análisis de los datos obtenidos de un mareógrafo, que es un sistema que mide la altura del nivel del mar en un punto concreto. Cada mareógrafo proporciona de forma periódica una medida que se guardará en un objeto de la clase `Medida` cuyo diagrama se muestra abajo y que ya está implementada. Sus atributos son:

- `horaGMT`: es un texto con la fecha y hora referidos al meridiano de Greenwich.
- `niveldelMar`: en metros
- `alturaOlas`: en metros

Como puede observarse, se dispone de un constructor al que se le pasan los datos de la medida excepto la fecha y hora que toma del reloj del sistema. También hay métodos observadores, uno para cada atributo:

Medida
-String horaGMT -double nivelDelMar -double alturaOlas
+Medida (double nivelDelMar, double alturaOlas) +double nivelDelMar() +double alturaOlas() +String horaGMT()

Mareografo
-ArrayList<Medida> lista -String codigo -int cadencia
+Mareografo(String codigo, int cadencia) +int numMedidas() +Medida medidaNum(int num) throws IndiceIncorrecto +double alturaMediaOlas(int intervalo) throws NoHayMedidas +boolean hayOlasMayores(double altura) +conecta()

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

- `lista`: guarda una lista de medidas obtenidas en instantes consecutivos a lo largo del tiempo. El primer elemento de la lista es el más antiguo, y el último el más nuevo.
- `codigo`: un String que identifica el mareógrafo.
- `cadencia`: el tiempo entre cada medida, en minutos.

Los métodos de la clase hacen lo siguiente:

- *constructor*: copia en los respectivos atributos el `codigo` y la `cadencia`. Además, crea el atributo `lista` vacío.
- `numMedidas()`: retorna el número de medidas que contiene la lista.

- medidaNum(): retorna la medida que ocupa la posición num en la lista (se numeran desde cero). Si num indica una medida no existente en la lista se lanzará la excepción IndiceIncorrecto definida en una clase aparte.
- alturaMediaOlas(): Retorna la altura media de las olas en el último intervalo expresado por el parámetro intervalo, en minutos. Para ello recorre la lista de medidas desde atrás sumando las alturas de las olas. En total debe recorrer un número de casillas igual a intervalo/cadencia. Finalizado el bucle retorna la media de las alturas de las olas. Si el tamaño de la lista es inferior a intervalo/cadencia lanza la excepción NoHayMedidas definida en una clase aparte.
- hayOlasMayores(): retorna un booleano que indica si entre las medidas hay alguna cuya altura de olas supere el parámetro altura.
- conecta(): se conecta por satélite con el mareógrafo y obtiene sus medidas, guardándolas en la lista. Este método no se pide

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

a. Crear un objeto de la clase Mareografo con código "PUERTO_CHICO_001" y cadencia 10 minutos
b. Invocar el método conecta() para establecer la conexión con el mareógrafo
c. Mostrar en pantalla los datos de la primera medida de la lista y de la última
d. Mostrar en pantalla la altura media de las olas en los últimos 60 minutos
e. Mostrar en pantalla si hubo olas mayores a 10 metros o no

En este programa tratar las excepciones con mensajes de error. Si se produce alguna excepción, continuar con los siguientes pasos que se piden.

Valoración:

- constructor, numMedidas y medidaNum: 1 punto
- alturaMediaOlas: 1.5 puntos
- hayOlasMayores: 1 punto
- programa principal: 1.5 punto