Examen de Programación
(Grados en Física y Matemáticas)

Septiembre 2016

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

1) Escribir un método java al que se le pasa un array de Strings de tamaño n, cada uno conteniendo la representación textual de dos números enteros separados por un espacio en blanco. El método debe retornar un array de números enteros conteniendo en las n primeras casillas todos los enteros situados a la izquierda en los strings y en las últimas n casillas todos los enteros situados a la derecha. Por ejemplo, si el array que se pasa contiene {"1 2", "3 4", "20 30"}, el resultado será el array {1,3,20,2,4,30}.

   El método tendrá la siguiente cabecera:

   ```
   public static int[] separarCifras(String[] lista)
   ```

   La obtención de las cifras izquierda o derecha puede hacerse de diversas formas. Una de ellas es con los métodos indexOf() (para localizar el espacio en blanco) y substring() de la clase String y luego la conversión de un String s a número entero se hace con siguiente el método estático de la clase Integer:

   ```
   public static int parseInt(String s) throws NumberFormatException
   ```

   Si el String no representa un número el método parseInt() falla lanzando la excepción NumberFormatException. Tratar esa excepción para que en caso de que se lance se ponga un mensaje de error en pantalla y se retorne null.

2) Escribir el *pseudocódigo* de un método iterativo que evalúa de forma aproximada la función tangente hiperbólica usando el siguiente desarrollo en serie:

$$\tanh(x) = \frac{1 \cdot x}{1} - \frac{1 \cdot x^3}{3} + \frac{2 \cdot x^5}{15} - \frac{17 \cdot x^7}{315} + \frac{62 \cdot x^9}{2835} - \frac{1382 \cdot x^{11}}{155925}$$

   El método recibe como parámetro el valor de x. Crear un array con los valores de los numeradores y otro con los denominadores. Para hacer el cálculo más eficiente no utilizar la operación "*elevar a*". Basta observar que cada potencia de x se calcula como la anterior multiplicada por x^2. Así, cada término i del desarrollo será numerador[i]*potencia/denominador[i].

3) La siguiente tabla muestra las tarifas de correos para envíos nacionales de cartas en euros:

Peso	Normal	Urgente
Hasta 20gr	0,45	2,65
Más de 20 hasta 50 gr	0,57	2,77
Más de 50 hasta 100 gr	0,95	3,15
Más de 100 hasta 500gr	2.10	4,30

Peso	Normal	Urgente
Más de 500 hasta 1000gr	4,67	9,37
Más de 1000 hasta 2000gr	5,20	9,90

Se desea escribir un método que reciba como parámetros el peso de la carta en gramos y un booleano que indique si es urgente o no. El método retornará el coste. Si el peso es negativo o superior a 2000 gr se lanzará la excepción PesoIncorrecto.

4) En un computador con sistema operativo Linux se dispone de un programa capaz de convertir un fichero de datos de extensión ".tsv" en otro de formato ".xlsx". El programa se llama convierteAxlsx, se encuentra en /bin y necesita como primer argumento el nombre del fichero a convertir y como segundo parámetro el nombre del directorio donde se depositará el resultado de la conversión, que será un fichero con el mismo nombre que el original y extensión ".xlsx". Por ejemplo:

```
/bin/convierteAxlsx   fichero1.tsv   directorio_destino
```

El primer argumento acepta caracteres comodín, por lo que la conversión se podrá aplicar a muchos ficheros a la vez.

Se desea escribir un script para procesar un conjunto de archivos situados en la carpeta datosUE situada en el directorio del usuario. Esta carpeta contiene a su vez dos carpetas, una llamada desempleo y otra ocupacion.

Hacer un dibujo con un esquema de la estructura de directorios de partida y otro con la estructura de directorios resultante de ejecutar el script. Suponer para ello que el directorio del usuario (cuyo nombre es desconocido) cuelga de /users.

El *script* se ejecutará comenzando desde el directorio del usuario y debe hacer los siguientes pasos para convertir los ficheros contenidos en datosUE y hacer una copia de seguridad en un disco duro externo situado en /mnt/disk1

- borrar del disco duro externo la carpeta copiaDatosUE con todos sus contenidos
- crear en el directorio del usuario la carpeta datosConvertidos, vacía.
- crear en esa nueva carpeta dos nuevas carpetas llamadas desempleo y ocupación
- convertir mediante el programa convierteAxlsx todos los ficheros situados en datosUE/desempleo que acaben en ".tsv" depositando los resultados de la conversión en datosConvertidos/desempleo
- hacer lo mismo con la carpeta datosUE/ocupacion depositando los resultados de la conversión en datosConvertidos/ocupacion
- crear en el disco duro externo una nueva carpeta llamada copiaDatosUE
- copiar en esa carpeta todas las carpetas y ficheros contenidos en datosConvertidos
- cambiar de nombre a las carpetas desempleo y ocupacion copiadas en datosConvertidos poniendo los nuevos nombres a copiaDesempleo y copiaOcupacion

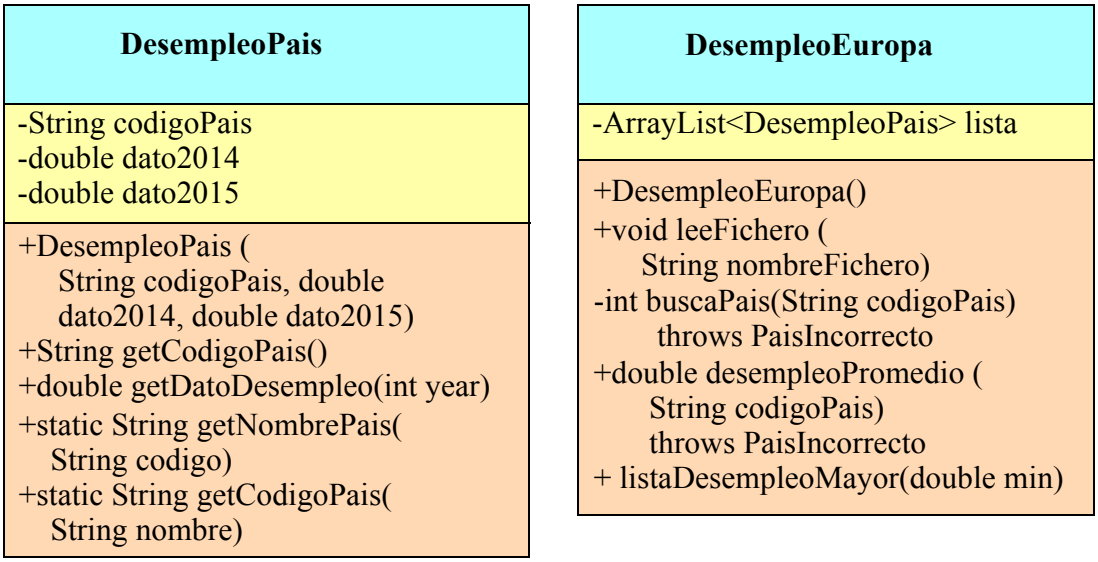
Examen de Programación
(Grados en Física y Matemáticas)

Septiembre 2016

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

Se desea hacer parte de un programa que permita tratar datos de la evolución del desempleo en los países de la unión europea en 2014 y 2015. Los datos de cada país se guardan en un objeto de la clase DesempleoPais cuyo diagrama se muestra y que se supone ya hecha. Es una clase con un constructor al que se pasan los valores iniciales de los atributos: codigoPais según la tabla oficial de la Unión Europea y dato2014 y dato2015 que son los datos de desempleo de los años 2014 y 2015 en % sobre la población activa. La clase DesempleoPais tiene un método observador para el atributo codigoPais y un método que retorna el dato de desempleo de 2014 o de 2015, según en valor del parámetro year. También tiene un par de métodos estáticos que permiten obtener el nombre de un país a partir del código, y viceversa.

Lo que se pide es crear la clase DesempleoEuropa que contiene un ArrayList de objetos de la clase DesempleoPais y que obedece al diagrama que se muestra. Sus métodos hacen lo siguiente:



- *constructor*: crea la lista vacía
- leeFichero(): Lee la lista de datos de desempleo del fichero de texto cuyo nombre se indica. El fichero tiene una línea de encabezamiento seguida de una línea para cada país. En la línea del país hay una palabra inicial seguida de 2 números reales con el desempleo en % sobre la población activa, correspondientes a los años 2014 y 2015. Los datos están separados por espacios en blanco. En los años para los que no se dispone de datos figura la palabra ":" en lugar del número.

  La palabra inicial contiene a su vez varias palabras separadas por comas y sin espacios en blanco. La última de estas palabras contiene el código del país.

  Si el fichero no existe se pone un mensaje en pantalla. Se supone que el fichero está correcto.

  Para este método utilizar el siguiente pseudocódigo:

```
Intentar abrir el fichero usando un scanner
    configurar el scanner en inglés
    // ignoramos la linea de encabezamiento
    leer línea del scanner
    // creamos array para recoger los datos de desempleo de un país
    array de reales datos= nuevo array de números reales de tamaño 2
    // leemos mientras queden palabras por leer
    mientras el scanner tiene una palabra disponible
      // leemos la primera palabra y la partimos por las comas
      texto primeraPalabra=leer palabra del scanner
      array de textos palabra=primeraPalabra.split(",")
      // el código de país es la ultima palabra tras la coma
      texto codigo=última casilla del array palabra
      // leer datos de 2014 y 2015
      para i desde 0 hasta 1
        si el scanner tiene un número real disponible entonces
          // hay un numero y lo leemos
          datos[i]=leer número real del scanner
        si no
          // no hay un número, sino la palabra ':'
          leer palabra del scanner
          datos[i]=valor NaN
        fin si
      fin para
      // Añadimos los datos a la lista
      añadir a lista nuevo objeto DesempleoPais(codigo,datos[0],datos[1])
    fin mientras
capturar excepción de fichero no existente
    poner mensaje de error en pantalla
fin intentar
```

- buscaPais(): Busca el pais cuyo código se indica en la lista. Si se encuentra retorna el índice de la casilla donde está. Si no se encuentra lanza PaisIncorrecto.
- desempleoPromedio(): Calcula y retorna el desempleo promedio para el pais cuyo codigo se indica, para los años 2014 y 2015. Si el código no existe en la lista se lanza PaisIncorrecto. En este método primero hay que buscar en la lista el país cuyo código se indica. Esto se hará llamando a buscaPais(). A partir del índice obtenido se obtienen los datos del desempleo y finalmente se retorna el promedio.
- listaDesempleoMayor(): Listado de los países con desempleo > min. Primero se pone en pantalla un encabezamiento. Luego, una línea por país. Para cada país se pone su código, su nombre completo y el desempleo medio obtenido con desempleoPromedio(). La excepción PaisIncorrecto se trata dentro del método

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

a. Crea un objeto de la clase DesempleoEuropa

b. Lee los datos de desempleo del fichero de nombre "tsdec400.tsv"

c. Muestra en pantalla los datos del desempleo promedio en "España" en 2014 y 2015. El código de país se obtendrá con el método getCodigoPais(). Si se lanza PaisIncorrecto en este paso poner un mensaje de error.

d. Muestra en pantalla el listado de países con desempleo mayor que el 10%. Este paso debe realizarse incluso si se lanzase PaisIncorrecto en el paso anterior.

Valoración:

- encabezamiento de la clase, atributos y constructor: 0.5 puntos
- `desempleoPromedio`: 0.5 puntos
- `leeFichero`, `buscaPais`, `listaDesempleoMayor` y programa principal: 1 punto cada uno