

Examen de Introducción al Software (Ingeniería Informática)

Septiembre 2018

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

1) La siguiente tabla muestra unas tarifas de luz y gas sin incluir el IVA:

Concepto	Franja de consumo	Luz	Gas
Término Fijo		4,34 €/mes	9,64 €/mes
Término de Potencia Contratada		46,093426 €/kW*año	
Término de Energía	>=0 y <200 kWh	0,1445 €/kWh	0,05444 €/kWh
	>=200 y <300 kWh	0,097058 €/kWh	
	>=300 kWh	0,079509 €/kWh	

Las tarifas de luz y gas se hallan guardadas en sendos arrays estáticos:

```
static double[] luz={4.34, 46.093426, 0.1445, 0.097058, 0.079509};
static double[] gas={9.64, 0.05444};
```

Escribir un método estático al que se le pasen los consumos de luz y de gas realizados por un hogar en un mes, en kWh, así como la potencia contratada, en kW (total 3 parámetros). El método calcula y retorna el pago a realizar en ese mes, que es la suma de los productos de los consumos por sus respectivas tarifas, más los términos fijos de gas y luz y más el término de potencia contratada por la potencia contratada dividida entre 12 meses que hay en el año. Posteriormente hay que añadir un 21% de IVA. Si alguno de los consumos o la potencia contratada son negativos retornar Double.NaN para indicar el error.

Nota: Se valorará la eficiencia y tamaño compacto del código.

2) Escribir un método iterativo que evalúa de forma aproximada la siguiente función integral que no se puede expresar mediante funciones elementales, usando el siguiente desarrollo en serie, válido para valores de x pequeños:

$$\int \frac{1}{\sqrt{1-x^4}} dx \cong \frac{x}{1} + \frac{x^5}{10} + \frac{x^9}{24} + \frac{5x^{13}}{208} + \frac{35x^{17}}{2176} + \frac{3x^{21}}{256}$$

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^4 . Así, cada término i del desarrollo será la potencia por el numerador i -ésimo dividido entre el denominador i -ésimo.

3) Se desea escribir un método *recursivo* que encuentra el k -ésimo elemento más pequeño de un array desordenado de números distintos. El método debe seguir el siguiente pseudocódigo:

```

método buscaPorRanking (ArrayList<real> lista, entero k) retorna real
// Elegimos p, un elemento cualquiera de lista
entero i=número aleatorio entre 0 y el tamaño de lista menos uno
real p = elemento número i de lista
// Creamos dos listas: para los elementos menores que p y para los mayores
ArrayList<real> lis1 = lista vacía
ArrayList<real> lis2 = lista vacía
// Rellenamos las listas lis1 y lis2
para cada x en lista
  si x<p entonces
    añade x a lis1
  si no, si x>p entonces
    añade x a lis2
  fin si
fin para
// Ahora lista = {lis1 unión {p} unión lis2}, siendo cada elemento de lis1
// menor que p y cada elemento de lis2 mayor
si k<= tamaño de lis1 entonces
  // el elemento buscado está en lis1 y seguimos buscando
  retorna buscaPorRanking (lis1,k)
si no, si k>=tamaño de lis1+2 entonces
  // el elemento buscado está en lis2 y seguimos buscando
  retorna buscaPorRanking (lis2,k-1- tamaño de lis1)
si no
  // el elemento buscado es p directamente
  retorna p
fin si
fin método

```

- 4) Contestar *razonadamente* a las siguientes preguntas. Utilizar un *máximo* de 5 líneas para cada respuesta:
- En una hoja de cálculo tenemos en la celda B2 un valor numérico que representa un porcentaje de impuesto. En la columna C tenemos escrita una serie de valores numéricos con precios de productos. Queremos que en la columna D aparezcan los precios con impuestos incluidos, es decir los valores de la columna C añadiendo el impuesto indicado en la celda B2. ¿Qué fórmula debemos escribir en la celda D1 para ello? Queremos poder copiar esa fórmula en el resto de casillas de la columna.
 - Disponemos de una hoja de cálculo con valores entre las filas 4 a 40 en la columna A. Indicar los pasos que darías para que en las mismas filas de la columna C aparezca el valor de la columna A de su misma fila o -1 si este es negativo.
 - Explica con brevedad para qué se utiliza una clave secundaria o extranjera en una base de datos.
 - Describe cómo se seleccionarían en una consulta registros que cumplan estas tres propiedades: su añoDeNacimiento es superior a 1960. Su ciudadDeResidencia es "Torrelavega". Su sueldoNeto es superior a 30000 euros.

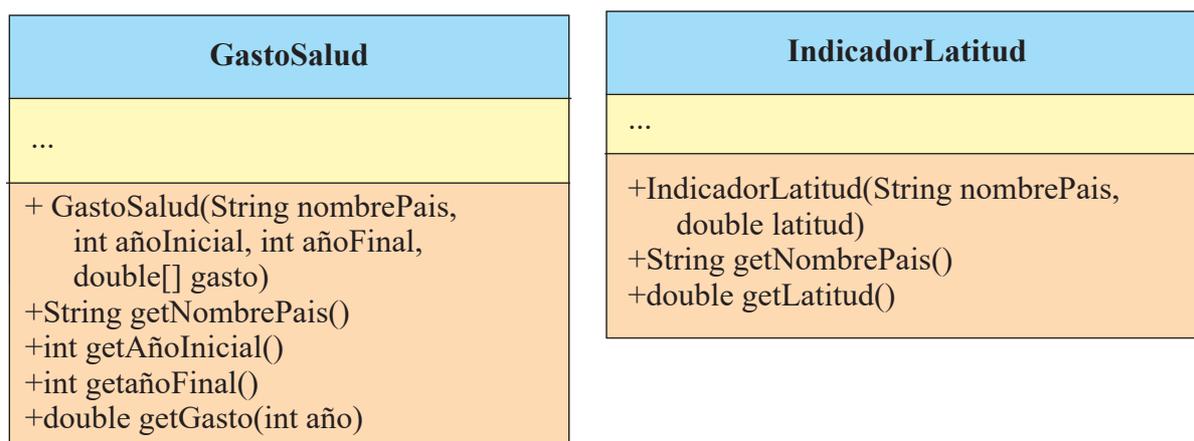
Nota: En esta cuestión, las respuestas que pasen de 5 líneas no se evaluarán. Se valora la *precisión* de la respuesta.

Examen de Introducción al Software (Ingeniería Informática)

Septiembre 2018

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

Se desea escribir parte de una aplicación para analizar datos de gastos en salud de diferentes países. Se dispone ya de tres clases realizadas cuyos diagramas de clase se muestran aquí. La clase GastoSalud contiene los datos de gastos en salud per cápita de un país en diferentes años, IndicadorLatitud contiene datos de la latitud de un país y LecturaDatos contiene métodos para leer listas de datos.



Los métodos de la clase GastoSalud hacen lo siguiente:

- *Constructor*: Construye el objeto con los datos que se pasan como parámetros
- `getNombrePais()`: Retorna el nombre del país
- `getAñoInicial()`: retorna el año inicial para el que hay datos de gasto para este país
- `getAñoFinal()`: retorna el año final para el que hay datos de gasto para este país
- `getGasto()`: Retorna el gasto en salud en el año indicado, en US\$, o Double.NaN si no hay datos para el año indicado

Y los de IndicadorLatitud:

- *Constructor*: Construye el objeto con los datos que se pasan como parámetros, estando la latitud en grados.
- `getNombrePais()`: observador del atributo nombrePais
- `getLatitud()`: observador del atributo latitud, en grados

Se pide escribir la clase DatosSaludPaises, excepto su método buscaLatitud() que se supone ya hecho. Sus atributos son una lista de objetos de la clase GastoSalud y una lista de objetos de la clase IndicadorLatitud. Los métodos aparecen en el diagrama de clase y deben hacer :

- *Constructor*: lee los datos de gastos en salud y latitudes de sendos ficheros cuyos nombres se pasan como parámetros. Para ello utiliza los métodos de la clase LecturaDatos, que se supone ya hecha (ver diagrama de clase).
- `buscaLatitud()`: Busca un dato de latitud por el nombre del país. Retorna el dato si se encuentra, o null si no se encuentra. Este método se supone ya hecho y **no se pide**.

- `gastoMedioLatitudSuperior()`: Calcula y retorna el gasto medio en salud en el año indicado considerando solo los países que tienen dato de latitud y de gasto en salud para el año indicado, y cuyo valor absoluto de latitud es mayor que el parámetro `latitudMinima`, expresado en grados. Si no hay ningún país que cumple los criterios se retornará `Double.NaN`.

LecturaDatos
+static ArrayList<GastoSalud> leeGastosSalud(String nombreFichero)
+static ArrayList<IndicadorLatitud> leeLatitudes(String nombreFichero)

- `buscaGastoMayor()`: Busca y retorna el nombre del primer país cuyo gasto máximo en salud entre 1995 y 2010 excede el parámetro `gastoMinimo`, expresado en US\$. Solo se tendrán en cuenta los países con año inicial `<=1995` y año final `>=2010`. Si no encuentra ningún país que cumpla los criterios se retornará `null`.

DatosSaludPaises
- ArrayList<GastoSalud> listaGasto
- ArrayList<IndicadorLatitud> listaLatitud
+DatosSaludPaises (String nombreFicheroGastos, String nombreFicheroLatitudes)
-IndicadorLatitud buscaLatitud(String nombrePais)
+double gastoMedioLatitudSuperior(int año, double latitudMinima)
+String buscaGastoMayor(double gastoMinimo)
+double latitudMedia(double gastoMinimo)

- `latitudMedia()`: Calcula y retorna la latitud media (en valor absoluto y en grados) de los países cuyo gasto medio en salud para los años en que tenga datos, supera el gasto mínimo indicado en US\$. Solo se tienen en cuenta los países con datos de latitud. Retorna `Double.NaN` si ningún país cumple los requisitos indicados. Para este método se seguirá el siguiente pseudocódigo:

```

método latitudMedia(double gastoMinimo) {
    real sumaLatitudes=0.0
    entero numPaises=0
    para cada g en listaGasto
        IndicadorLatitud lat=buscaLatitud(nombre del país de g);
        // consideramos solo los países con dato de latitud
        si lat distinto de null
            // Calculamos el gasto medio en salud
            real sumaGastos=0.0
            entero num=0
            para a desde el año inicial de g hasta el año final de g
                suma 1 a num
                sumaGastos=sumaGastos+gasto de g en el año a
            fin para
            real gastoMedio=sumaGastos/num
            // comprobamos el requisito de gasto mínimo
            si gastoMedio>gastoMinimo
                suma 1 a numPaises
                sumaLatitudes=sumaLatitudes+valor absoluto de la latitud de lat
            fin si
        fin si
    fin para
    retorna sumaLatitudes/numPaises
fin método

```

Escribir además en una clase aparte un programa de prueba con un main que haga:

- a) Crear un objeto de la clase DatosSaludPorPais leyendo de los ficheros health-spending.csv e indicator-latitude.csv.
- b) Muestra en pantalla el gasto medio en salud en 2010 para países con latitud superior a 20° en valor absoluto, usando el método gastoMedioLatitudSuperior().
- c) Muestra en pantalla el nombre del primer país cuyo gasto máximo en salud entre 1995 y 2010 excede 8000 US\$, usando el método buscaGastoMayor(). Si el valor retornado por ese método es null, se pondrá un mensaje de error en pantalla.
- d) Muestra en pantalla la latitud media (en valor absoluto) de los países cuyo gasto medio en salud supera 5000 US\$

Valoración (sobre 5):

- 1) Encabezamiento de la clase, atributo y constructor: 0.5 puntos
- 2) gastoMedioLatitudSuperior: 1 punto
- 3) buscaGastoMayor: 1 punto
- 4) latitudMedia: 1 punto
- 5) main: 1.5 puntos