

Examen de Introducción al Software (Ingeniería Informática)

Septiembre 2013

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

- 1) Escribir un método Java siguiendo la cabecera que se muestra abajo que convierta el string código en un entero y lo retorne. Se supone que el parámetro código contiene dos letras (no es preciso comprobar errores). Para la conversión a número, se convertirá cada letra a número usando la tabla que se muestra abajo. Luego se retornará la conversión de la primera letra multiplicada por 16 y sumada a la conversión de la segunda letra.

public int convierte (String código)

| Letra | Número |
|----------------------|--------|
| A | 10 |
| B | 11 |
| C | 12 |
| D | 13 |
| E | 14 |
| F | 15 |
| Cualquier otra letra | -1 |

- 2) Escribir el *pseudocódigo* de un método al que se le pasa como parámetro un array de dos dimensiones conteniendo una matriz cuadrada de números reales. El método debe retornar un booleano indicando si en la matriz hay una cantidad de ceros superior a $n/2$, siendo n el número de casillas del array (es decir, el número de filas por el número de columnas).
- 3) Escribir un método Java al que se le pasa como parámetro un array unidimensional de números enteros y que calcule y retorne cuántos de esos números son iguales al mínimo valor entero.
- 4) Escribir en Java un método recursivo que ponga en pantalla los n primeros números enteros positivos en orden del n al 1. El valor n se le pasa como parámetro. El caso directo se da cuando n vale 0, y en ese caso el método no hace nada. En el caso recursivo se pone en pantalla en valor de n y se invoca al mismo método para $n-1$.
- 5) Contestar *razonadamente* a las siguientes preguntas. Utilizar un *máximo* de 3 líneas para cada respuesta:
- a. En una hoja de cálculo, ¿cómo se sabe si en una celda hay una fórmula o un valor numérico?
 - b. Escribir una fórmula para una hoja de cálculo que calcule la suma de las cuatro

primeras casillas de la columna C, de forma que si la fórmula se copia en otro lugar la suma siga haciéndose con esas casillas, y no otras.

- c. Describe brevemente los pasos a realizar en una hoja de cálculo para insertar un gráfico que use los datos contenidos en las dos primeras columnas de la tabla.
- d. Una tabla de una base de datos ¿puede tener varias claves primarias? ¿por qué?
- e. ¿Es posible acceder a una consulta de una base de datos desde una página Web? En caso afirmativo, pon un ejemplo que conozcas.

Nota: en esta cuestión, las respuestas correctas suman 0.2 puntos y las incorrectas restan 0.1 puntos. Se valora la *precisión* de la respuesta.

Examen de Introducción al Software (Ingeniería Informática)

Septiembre 2013

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

Se desea realizar una parte del software de una biblioteca. Para ello se dispone de la clase Libro, ya realizada y que se corresponde al diagrama de clases. Sus métodos hacen lo siguiente:

- *Constructor*: Construye el libro a partir del título, autor y signatura pasados como parámetros; pone disponible a true
- autor(): observador del atributo autor
- titulo(): observador del atributo titulo
- signatura(): observador del atributo signatura
- disponible() Retorna un booleano que indica si el libro está disponible para prestarlo, o no
- cliente(): Retorna el nombre del cliente al que se ha prestado el libro, o el texto "-" si el libro está disponible
- presta(): realiza el préstamo del libro al cliente indicado, que se almacenará en el atributo cliente; pone disponible a false
- devuelve(): pone disponible a true
- toString(): Retorna una línea de texto con los datos del libro

| Libro |
|--|
| - String titulo, autor, signatura, cliente - boolean disponible |
| + Libro(String titulo, String autor, String signatura) + String autor() + String titulo() + String signatura() + String cliente() + boolean disponible() + void presta(String cliente) + void devuelve() + String toString() |

Se pide escribir la clase Biblioteca que contiene un ArrayList de libros y dispone de operaciones para su manipulación. La clase obedece al diagrama de clases que se muestra. Sus métodos hacen lo siguiente:

- *Constructor*: crea la lista vacía.
- añade(): Primero comprueba si en la biblioteca hay algún libro con la misma signatura que lib. En caso afirmativo retorna false. En caso negativo (ningún libro tiene la misma signatura) añade el libro lib al final de la lista y retorna true para indicar que el libro se ha podido añadir. Para la búsqueda de un libro con la misma signatura debe utilizarse uno de los *esquemas de búsqueda* vistos en clase.
- devuelveTodos(). Recorre todos los libros de la lista y para cada uno de ellos que no esté disponible y cuyo cliente coincida con el parámetro cliente se invoca a su método devuelve().

| Biblioteca |
|---|
| - ArrayList<Libro> lista |
| + Biblioteca() + boolean añade(Libro lib) + devuelveTodos(String cliente) + double mediaLibrosPorCliente() |

- `mediaLibrosPorCliente()`. Devuelve la media de libros que tiene prestados cada cliente. Utilizar el siguiente pseudocódigo:

```

método mediaLibrosPorCliente() retorna real
  // Atributo para guardar los clientes de la biblioteca
  ArrayList<String> clientes = nueva lista vacía
  // Atributo para guardar el número de libros de cada cliente
  entero[] numLibros = nuevo array de tamaño igual al número total de libros

  // Recorrer todos los libros de la biblioteca
  para cada lib en lista hacer
    si lib está disponible entonces
      entero indice = clientes.indexOf(cliente de lib)
      si indice = -1 entonces
        // el cliente es nuevo; lo añadimos a la lista
        añadir el cliente de lib a clientes
        // poner el número de libros de este cliente a 1
        numLibros[tamaño de clientes-1]=1
      si no
        // el cliente ya estaba; sumamos uno a su número de libros
        numLibros[indice]++
      fin si
    fin si
  fin para

  // Ahora calculamos la suma de los resultados
  entero sum=0
  entero n=tamaño de clientes
  para i desde 0 hasta n-1 hacer
    sum=sum+numLibros[i]
  fin para

  // Retornamos la media
  retorna (conversión a real de sum)/n
fin método

```

Notas:

- el método `indexOf` de la clase `ArrayList` retorna el índice de la primera casilla de la lista cuyo contenido coincide con el parámetro, o -1 si no hay ninguna casilla que coincida.
- en la asignatura de *Estructuras de Datos* se verán formas más eficientes de resolver este método, pero aquí debe usarse el pseudocódigo indicado.

Valoración:

- 1) Encabezamiento de la clase, atributo y constructor: 0.5 puntos
- 2) Resto de los métodos: 1.5 puntos cada uno