

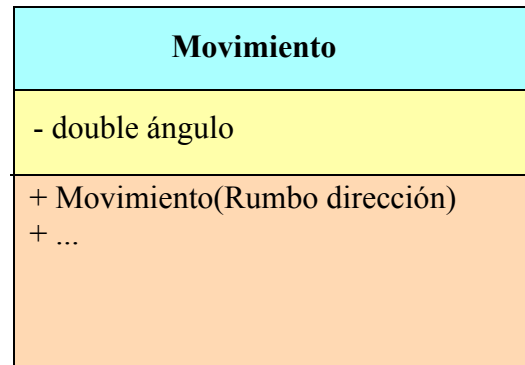
Examen de Introducción al Software (Ingeniería Informática)

Febrero 2013

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

- 1) Escribir el constructor de una clase que obedece al diagrama de clases que se muestra parcialmente y que da valor al atributo ángulo según el valor que se le pasa en el parámetro dirección, usando los valores de la tabla inferior.

dirección	ángulo (°)
NORTE	0
SUR	180
ESTE	90
OESTE	270



Rumbo es una clase enumerada definida de la forma siguiente:
`public enum Rumbo {NORTE, SUR, ESTE, OESTE}`

- 2) Escribir el pseudocódigo de un método que pone a cero todos los elementos de la fila i de un array de números reales, a , de dos dimensiones (la 1ª dimensión es la fila y la 2ª es la columna). El método recibe como parámetros i y a , y no retorna nada.
- 3) Se dispone de una clase llamada Fecha cuyos objetos representan fechas del calendario. La clase dispone de un método estático cuya cabecera se muestra abajo y que retorna la diferencia de días entre dos fechas que se pasan como parámetros.
`public static int diferencia(Fecha f1, Fecha f2)`

Escribir un método Java con la cabecera que se indica abajo que calcula la suma de los valores absolutos de todas las diferencias de fechas que hay entre cada fecha y la siguiente dentro de un array de objetos de la clase Fecha que se pasa como parámetro.
`public static int sumaDiferencias(Fecha[] fechas)`

- 4) Escribir en Java un método recursivo que calcule y retorne el número n -ésimo de la sucesión de Fibonacci, siendo n un número natural. El caso directo se da cuando n vale 0 o 1. En ese caso hay que retornar n . En el caso recursivo hay que retornar el número de Fibonacci $(n-1)$ -ésimo sumado al número de Fibonacci $(n-2)$ -ésimo.
- 5) Contestar *razonadamente* a las siguientes preguntas. Utilizar un *máximo* de 3 líneas para cada respuesta:
- En una hoja de cálculo, ¿qué diferencia hay entre una celda con una etiqueta o con un valor numérico?
 - Escribir una fórmula para una hoja de cálculo que calcule la media de los 50

primeros elementos de la columna B, de forma que si la fórmula se copia en otra casilla la media siga haciéndose con esas casillas, y no otras.

- c. En una hoja de cálculo ¿podemos meter documentos e imágenes? o ¿sólo datos numéricos y de texto?
- d. Un registro de una base de datos, ¿puede tener varias claves primarias? y ¿varias claves extranjeras?
- e. ¿Qué diferencia hay entre usar una base de datos con SQL y hacerlo con un gestor basado en una interfaz gráfica como base?

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)

Febrero 2013

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

Se desea realizar una parte del software perteneciente a un sistema de reserva de billetes de avión. Para ello se dispone de las siguientes clases ya realizadas:

Fecha	Billete
...	...
+ Fecha(int dia, int mes, int año) + String toString() + static Fecha hoy() + boolean equals(Fecha f) + static int diferencia(Fecha f1, Fecha f2)	+ Billete(String nombrePasajero, String códigoReserva, Fecha fechaVuelo, String númeroVuelo) + String nombrePasajero() + String códigoReserva() + Fecha fechaVuelo() + String númeroVuelo() + boolean tarjetaEmitida() + Fecha fechaEmisiónTarjeta() + emiteTarjetaEmbarque()

La clase Fecha contiene una fecha del calendario. Sus métodos hacen lo siguiente:

- *Constructor*: Construye la fecha a partir del día, mes y año pasados como parámetros
- *toString()*: Convierte la fecha a texto
- *hoy()*: Retorna una fecha correspondiente al día de hoy
- *equals()* Retorna un booleano que indica si la fecha actual es igual a la que se pasa como parámetro
- *diferencia()*: Retorna la diferencia de días entre dos fechas que se pasan como parámetros (f2-f1)

La clase Billete representa un billete de avión de un pasajero concreto. sus métodos son:

- *Constructor*: construye el billete usando los parámetros con el nombre del pasajero, el código de reserva, la fecha del vuelo y el número del vuelo
- *nombrePasajero()*: Retorna el nombre del pasajero
- *códigoReserva()*: Retorna el código de reserva
- *fechaVuelo()*: Retorna la fecha de vuelo
- *númeroVuelo()*: Retorna el número de vuelo
- *tarjetaEmitida()*: Retorna si la tarjeta de embarque está emitida o no
- *fechaEmisiónTarjeta()*: Retorna la fecha de emisión de la tarjeta de embarque, o null si no ha sido emitida aún
- *emiteTarjetaEmbarque()*: Emite tarjeta de embarque con la fecha de hoy

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

ListaReservas
- ArrayList<Billete> lista
+ ListaReservas() + boolean añade(Billete b) + muestraTarjetasEmbarque (String códigoReserva) + double mediaDiasEntreVuelos()

- *Constructor*: crea la lista vacía
- *añade()*: Añade el billete b a la lista si no existe ya un billete con el mismo pasajero, fecha y número de vuelo. Retorna un booleano que indica si se pudo añadir el billete o no
- *muestraTarjetasEmbarque()*: Muestra en la pantalla del sistema las tarjetas de embarque de la reserva que se indica (pueden ser varias, para pasajeros que viajan juntos). Cada tarjeta tiene una línea con el código de reserva, el número de vuelo y la fecha de vuelo; en una segunda línea aparece el nombre del pasajero; en la tercera línea, la fecha de emisión de la tarjeta; luego dos líneas en blanco. Cada dato viene precedido de una etiqueta breve con un texto explicativo. Si no hay ninguna tarjeta con ese código de reserva se pone en pantalla un mensaje de error
- *mediaDiasEntreVuelos()*: Obtiene el número medio de días transcurridos entre un vuelo y el siguiente para la misma persona, y retorna la media para todos los pasajeros de la lista. Se supone que las reservas de cada persona en la lista están ordenadas por fechas de vuelo

Para el método *mediaDiasEntreVuelos()* utilizar el siguiente pseudocódigo:

```

método mediaDiasEntreVuelos() retorna real
  // Variables para meter la respuesta final
  real sumaMedias=0.0
  entero numMedias=0
  // crea una lista de pasajeros únicos
  ArrayList<String> nombres= nueva lista vacía
  para cada Billete b en lista hacer
    // si el nombre de b no está en la lista lo metemos
    si nombres no contiene a b.nombrePasajero() entonces
      añade a nombres b.nombrePasajero()
    fin si
  fin para
  // para cada pasajero calcula la media de los días entre vuelos
  entero sumaDiferencias=0
  entero numDiferencias=0
  // Recorre todos los nombres
  para cada String nombre en nombres hacer
    // aquí anotamos la última fecha de vuelo encontrada
    Fecha ultimaFecha=null
    // Recorre toda la lista de reservas
    para cada Billete b en lista hacer
      // si el nombre del pasajero coincide con nombre, lo procesamos
      si b.nombrePasajero() igual a nombre entonces
        // Si había una fecha anterior sumamos la diferencia con la actual
        si ultimaFecha!=null entonces
          sumaDiferencias=sumaDiferencias+
            diferencia de días entre b.fechaVuelo() y ultimaFecha
          numDiferencias++

```

```
        fin si
        // anotamos la nueva fecha como la última
        ultimaFecha=b.fechaVuelo()
    fin si
fin para
// calculamos la media y la sumamos al resultado
si numDiferencias>0 entonces
    real media= conversión a real de sumaDiferencias/numDiferencias
    sumaMedias=sumaMedias+media
    numMedias++;
fin si
fin para
// retornamos la media, o NaN si no hay datos
si numMedias > 0 entonces
    retorna sumaMedias/numMedias
si no
    retorna Double.NaN // no hay media
fin si
fin método
```

Notas: para saber si un ArrayList llamado lista contiene un dato d puede usarse el método lista.contains(d), que retorna un booleano. Recordar que la diferencia de días se obtiene con el método diferencia de la clase Fecha.

Valoración:

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