

Programación Concurrente y de Tiempo Real^{*}

Grado en Ingeniería Informática

Asignación de Prácticas Número 2

Se le plantean a continuación un conjunto de ejercicios sencillos de programación sobre orientación a objetos y clases, que debe resolver de forma individual como complemento a la segunda sesión práctica. Para cada uno, debe desarrollar un programa independiente que lo resuelva. **Documente todo su código con etiquetas (será sometido a análisis con javadoc)**. Si lo desea, puede también agrupar su código en un paquete de clases, aunque no es obligatorio.

1. Ejercicios

1. Escriba una clase en java para modelar la entidad Paciente (de una consulta médica), y que considere al menos los atributos siguientes: nombre, dni, dirección, teléfono y compañía de seguros. Guarde su trabajo en `Paciente.java`. Modele ahora una consulta abstrayéndola como una colección de paciente. Diseñe un menú de usuario que permite insertar, borrar y consultar pacientes utilizando diferentes parámetros de búsqueda. Guarde su trabajo en `Biblioteca.java`

2. Escriba un programa en java que reciba como argumentos desde la línea de comandos dos números naturales n, m y despliegue en pantalla el valor de la función de Ackermann evaluada sobre esos argumentos. Guarde su trabajo en `Ack.java`. Desarrolle una pequeña tabla (**recuerde que debe desarrollar el document con WriteLatex**) donde escribirá algunas combinaciones de n, m y $Ack(m, n)$. ¿Qué observa en relación con esta prueba y esta particularísima función? ¿Cree que la concurrencia podría jugar algún papel útil para evaluarla mejor? ¿Por qué? Guarde la tabla y sus impresiones en un fichero llamado `analisis.pdf`.

3. Escriba un programa en java que lea una secuencia de valores numéricos cuya longitud será determinada por el usuario mediante un parámetro transferido desde la línea de comandos del sistema. El programa leerá la secuencia, y deberá calcular e imprimir parámetros estadísticos estándar como media, moda, varianza y desviación típica, ofreciendo al usuario un menú (estructurado mediante `switch`) donde elegirá el estadístico que desea calcular. Guarde su trabajo en un fichero llamado `Estadistica.java`. Elabora también un documento

^{*}©Antonio Tomez

(recuerde que debe desarrollar el document con WriteLatex) llamado `estadisticos.pdf`, que contendrá la expresión matemática de las ecuaciones que definen a cada estadístico, y una breve descripción textual de su significado.

4. Utilice arrays para implantar una clase que modele número complejos. Dótele de métodos que permitan calcular suma, resta, módulo, producto y cociente. Guarde su trabajo en `Complejos.java`. Escriba ahora un programa llamado `usaComplejos.java` que presenta un menú de usuario que permitirá elegir la operación a realizar y posteriormente leerá los datos necesarios para hacerla, ofreciendo finalmente el resultado en pantalla.

5. Elabore una clase para abstraer el concepto matemático de elipse llamada `Elipse.java`. Además de los modificadores y observadores habituales, dótela de métodos para saber si un punto pertenece o no a una elipse dada. Escriba un programa llamada `usaElipse.java` que cree y utilice algunas instancias de la clase. Elabore también un pequeño documento (recuerde que debe desarrollar el document con WriteLatex) llamado `elipse.pdf`, que deberá incluir una descripción de la ecuación que define a este cónica, una descripción textual, y una imagen de ejemplo.

2. Procedimiento y Plazo de Entrega

Se ha habilitado una tarea de subida en *Moodle* que le permite subir cada fichero que forma parte de los productos de la práctica de forma individual en el formato original. Para ello, suba el primer fichero de la forma habitual, y luego siga la secuencia de etapas que el propio *Moodle* le irá marcando. Recuerde además que:

- Los documentos escritos que no sean ficheros de código deben generarse **obligatoriamente** utilizando Latex, a través del editor WriteLatex, disponible en la nube. Tiene a su disposición en el Campus Virtual un manual que le permitirá desarrollar de forma sencilla y eficiente documentos científicos de alta calidad. Puede encontrar el citado manual en la sección dedicada a Latex en el bloque principal del curso virtual. El url de WriteLatex es: <https://www.writelatex.com/>
- No debe hacer intentos de subida de borradores, versiones de prueba o esquemas de las soluciones. *Moodle* únicamente le permitirá la subida de los ficheros por **una sola vez**.
- La detección de plagio o copia en los ficheros de las prácticas, o la subida de ficheros vacíos de contenido o cuyo contenido no responda a lo pedido con una extensión mínima razonable, invalidará plenamente la asignación, sin perjuicio de otras acciones disciplinarias que pudieran corresponder.
- El plazo de entrega de la práctica se encuentra fijado en la tarea de subida del Campus Virtual.

- Entregas fuera de este plazo adicional no serán admitidas, salvo causa de fuerza mayor debidamente justificadas mediante documento escrito.
- Se recuerda que la entrega de todas las asignaciones de prácticas es recomendable, tanto un para un correcto seguimiento de la asignatura, como para la evaluación final de prácticas, donde puede ayudar a superar esta según lo establecido en la ficha de la asignatura.