

# Programación Concurrente y de Tiempo Real<sup>\*</sup>

## Grado en Ingeniería Informática

### Asignación de Prácticas Número 1

Se le plantean a continuación un conjunto de ejercicios sencillos de programación, que debe resolver de forma individual como complemento a la primera sesión práctica. Para cada uno, debe desarrollar un programa independiente que lo resuelva.

## 1. Ejercicios

1. Escriba un programa en java para calcular el volumen de un cono. Declare una constante que guarde el valor de  $\pi$ . Suponga un cono de 14,2 cm de diámetro en la base y de 20 cm. de altura . Guárdelo en un fichero llamado `Circulo.java`

2. Escriba un programa en java para encontrar el cero de una función  $f(x)$  mediante el método de *Newton-Raphson*. Este método iterativo construye una sucesión  $x_0, x_1, x_2, \dots$  de aproximaciones a la solución utilizando la siguiente ecuación:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

La aproximación inicial será introducida por teclado, junto con el número de iteraciones que permitirán obtener la aproximación a la raíz de la función  $f(x)$ . El programa irá imprimiendo en pantalla las sucesivas aproximaciones que va calculando. Aplique su programa a las funciones siguientes:

- $f(x) = \cos(x) - x^3$  en  $[0, 1]$
- $f(x) = x^2 - 5$  en  $[2, 3]$

Guarde su programa en un fichero llamado `NewtonRaphson.java`.

3. La integral definida en  $[0 - 1]$  de una función real de variable real  $f(x)$  puede calcularse mediante un método de Monte Carlo (probabilístico) inscribiendo la curva de la función en un cuadrado de lado igual a la unidad. Para aproximar el valor de la integral, se generan puntos aleatorios en el marco determinado por el cuadrado, y se cuentan únicamente aquellos puntos que están situados bajo la curva. La razón entre el número de puntos bajo la curva y el número total

---

<sup>\*</sup>©Antonio Tomeu

de puntos es una aproximación al valor buscado que naturalmente, conforme mayor es el número de puntos, mejora la aproximación. Escriba un programa java que permita realizar tal cálculo, leyendo desde teclado el número de puntos con el cuál genera la aproximación para las funciones siguientes:

- $f(x) = \sin(x)$
- $f(x) = x$

Guarde el programa en `intDefinidaMonteCarlo.java`

4. Dado que en un curso como el nuestro, explicar el subsistema completo de E/S de Java requiere un tiempo del que no disponemos, he aquí un código que resume los aspectos elementales sobre cómo hacer entrada/salida de tipos básicos, y que complementa al uso de la ya conocida clase `Scanner`. Estúdielo y, si es necesario, revise las API de las clases involucradas<sup>1</sup>. A partir de aquí, se supondrá que usted sabe leer y escribir tipos básicos. No se le pide escribir ningún programa sobre esto, pero asegúrese de que entiende y sabe utilizar la e/s básica.

```
/**Ejemplo de uso de las clase InputStreamReader junto con la
 * clase Buffered StreamReader, mas los metodos para leer
 * tipos primitivos desde entrada estandar
 * @author Antonio Tomeu
 */
import java.io.*;
public class ClasesES1
{
    public static void main(String[] args)
        throws Exception{
        System.out.println("ahora vamos a leer y escribir desde teclado
a pantalla ");
        BufferedReader dato = new BufferedReader(
            new InputStreamReader(System.in));
        short q = Short.parseShort(dato.readLine());
        System.out.println(2*q);
        int x = Integer.parseInt(dato.readLine());
        System.out.println(2*x);
        float y = Float.parseFloat(dato.readLine());
        System.out.println(2*y);
        byte k = Byte.parseByte(dato.readLine());
        System.out.println(2*k);
        double g = Double.parseDouble(dato.readLine());
        System.out.println(2*g);
    }
}
```

---

<sup>1</sup>Ya de paso, si lee el capítulo correspondiente a E/S de la obra *Thinking in Java*, eso que lleva ganado para el futuro. Usted mismo...

```

long m = Long.parseLong(dato.readLine());
System.out.println(2*m);
//pero para cadenas es mas facil, y no hay que hacer Parse...
String u = dato.readLine();
System.out.println(u);
//y para caracteres...como es conocido
int i = dato.read();
}
}

```

5. El cifrado de César es una técnica elemental de ocultamiento de la información que matemáticamente se describe de forma simple utilizando la siguiente ecuación

$$E(x) = x + n \bmod 27$$

donde  $x$  es la letra que queremos cifrar (representada por su código ASCII o por cualquier otra ordenación válida) y  $n$  es un número que se suma a ese código. Escriba un programa `Cesar.java` que lea el valor de  $n$ , una cadena de texto cualquiera, y muestre en pantalla su representación cifrada. Escriba otro programa llamado `desCesar.java` que efectúe el descifrado de acuerdo a la siguiente ecuación:

$$D(k) = k - n \bmod 27$$

6. Una tarea que será de utilidad durante el curso es la generación de números aleatorio. En Java, esto puede lograrse de dos manera diferentes. Comencemos ahora por la primera; escriba un programa llamado `aleatorios.java` que uso del método `random()` de la clase `Math` para generar una secuencia de números aleatorios. La longitud de la secuencia será fijada mediante un argumento leído por el programa desde la línea de comandos de una ventana de terminal.

## 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.