Programacion Concurrente y de Tiempo Real
Práctica 1

José Manuel Barba González

26 de octubre de 2014

Resumen

Este documento corresponde a la primera práctica de la asignatura, donde se hace énfasis en la entrada de datos por teclado y la generación de números reales mediante funciones.

Introducción

En esta práctica se entregan cinco de los seis problemas indicados puesto que el número cuatro es un ejemplo de cómo aplicar o manejar el subsistema de E/S. El primero nos pide calcular el volumen de un cono. El segundo hallar el cero de dos funciones $f(x)$ mediante el método Newton-Raphson. El tercero nos pide sucesiones aleatorias con el método MonteCarlo. El quinto realiza la técnica de cifrado del Cesar y el sexto ejercicio solo crearemos una secuencia aleatoria de números.

1. Problema 1

Problema que resuelve la ecuación del volumen de un cono.

1.1. Enunciado

Escriba un programa en java para calcular el volumen de un cono. Declare una constante que guarde el valor de π. Suponga un cono de 14,2 cm de diametro en la base y de 20 cm. de altura. Guárdelo en un fichero llamado **Circulo.java**.

1.2. Solución

```
/* Fichero Circulo.java
  *@author Jose Manuel Barba Gonzalez
  *@version 1.0
  *Programacion Concurrente y de Tiempo Real
  *Area de CC. de la Computacion e I.A.
*/

import java.util.Scanner;
import java.lang.*;

public class Circulo
{
        //v=(pi*r^2*h)/3
        public static void main(String[] args)
        {
                double pi = 3.141516;
```

```
                Scanner diametro, altura;
                double volumen;
                double d, h, r;

                System.out.println("Introduzca la altura: ");
                altura = new Scanner(System.in);
                h = altura.nextDouble();
                System.out.println("Introduzca el diametro: ");
                diametro = new Scanner(System.in);
                d = diametro.nextDouble();

                r = d/2;
                volumen = (pi*(r*r)*h)/3;
                System.out.println("El volumen del cono con radio "+r+" y
                altura "+h+" es "+volumen);
        }
}
```

2. Problema 2

Problema que resuelve el cero de una función.

2.1. Enunciado

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 $x_0, x_1, x_2, \ldots$ 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 **Newton-Raphson.java**

2.2. Solución

```
/* Fichero NewtonRaphson.java
  *@author Jose Manuel Barba Gonzalez
  *@version 1.0
  *Programacion Concurrente y de Tiempo Real
  *Area de CC. de la Computacion e I.A.
*/

import java.io.*;
import java.util.*;
import java.lang.*;
```

```
public class NewtonRaphson
{

        public static void main(String[] args)
        throws IOException
        {
                double p,h;
                int d,i,j;
                Scanner aproximacion, iteraciones;
                double [] vector;
                double [] vect;

                System.out.println("Introduzca la aproximacion inicial: ");
                aproximacion = new Scanner(System.in);
                h = aproximacion.nextDouble();
                System.out.println("Introduzca las iteraciones para la
                aproximacion(minimo 10): ");
                iteraciones = new Scanner(System.in);
                d = iteraciones.nextInt();
                if(d < 10) d = 10;

                System.out.println("Funcion f(x) = cos(x) - x^3");
                vector = new double[d];
                vector[0] = (h/1) - ((Math.cos(h)-1)/(-Math.sin(h)-1));
                for(i = 1; i < d && vector[i-1] > 0.0; i++)
                {       //f(x)=cos(x) - x^3
                        vector[i] = (h/i) - ((Math.cos(h)-(i*i*i))/(-Math.
                        sin(h)-(i*i)));
                        System.out.println("Iteracion "+i+", aproximacion:
                        "+vector[i]);
                }

                System.out.println("Funcion f(x) = x^2 - 5");
                vect = new double[d];
                vect[0] = (h/1) - (((1*1)-5)/1);
                for(j = 1; j < d && vect[j-1] > 0.0; j++)
                {       //f(x)=x^2 - 5
                        vect[j] = (h/j) - (((j*j)-5)/j);
                        System.out.println("Iteracion "+j+", aproximacion:
                        "+vect[j]);
                }
        }
}
```

3. Problema 3

Problema que resuelve el método probabilístico MonteCarlo.

3.1. Enunciado

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 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 su programa en un fichero llamado **intDefinidaMonteCarlo.java**

3.2. Solución

```
/* Fichero intDefinidaMonteCarlo.java
  *@author Jose Manuel Barba Gonzalez
  *@version 1.0
  *Programacion Concurrente y de Tiempo Real
  *Area de CC. de la Computacion e I.A.
*/

import java.io.*;
import java.util.*;
import java.lang.*;

public class intDefinidaMonteCarlo
{
        public static void main(String[] args)
        {
                int d,i;
                Scanner puntos;
                double suma1 = 0, suma2 = 0;

                System.out.println("Introduzca el numero de puntos
                (minimo10): ");
                puntos = new Scanner(System.in);
                d = puntos.nextInt();
                if(d < 10) d = 10;

                for(i = 0; i < d; i++)
                //f(x)=sin(x)
                        suma1 = suma1 + Math.sin(Math.random());

                System.out.println("Resultado de la aproximacion MonteCarlo
                Funcion f(x) = sin(x): " + suma1/d);

                for(i = 0; i < d; i++)
                        suma2 = suma2 + Math.random();//f(x) = x

                System.out.println("Resultado de la aproximacion MonteCarlo
                Funcion f(x) = x: " + suma2/d);
        }
```

}

4. Problema 5

Problema que resuelve el cifrado del César.

4.1. Enunciado

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

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

donde x es la letra que queremos cifrar (representada por su código ASCII 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. Escribra otro programa llamado desCesar.java que efectúe el descifrado de acuerdo a la siguiente ecuación:

$$D(x) = x - n \ mod \ 27$$

4.2. Solución

Código de cifrado **Cesar.java**

```
/* Fichero Cesar.java
  *@author Jose Manuel Barba Gonzalez
  *@version 1.0
  *Programacion Concurrente y de Tiempo Real
  *Area de CC. de la Computacion e I.A.
*/

import java.io.*;
import java.util.*;
import java.lang.*;

public class Cesar
{
        public static void main(String[] args)
        throws Exception
        {
                int n, i, value;
                char []tmp;
                char []tmpcif;
                String clave, clave_cif;
                Scanner numero;

                System.out.println("Introduzca el numero para sumar
                a la clave: ");
                numero = new Scanner(System.in);
                n = numero.nextInt();

                System.out.println("Introduzca su contrasena: ");
                BufferedReader dato = new BufferedReader
                (new InputStreamReader(System.in));
```

```
                clave = dato.readLine();

                System.out.println("Funcion E(x) = x + n mod 27");
                tmp = new char[clave.length()];
                tmp = clave.toCharArray();
                tmpcif = new char[clave.length()];
                for(i = 0; i < clave.length(); i++)
                {
                        value = (tmp[i] + n %27);
                        tmpcif[i] = (char) value;
                }
                clave_cif = new String(tmpcif);
                System.out.println(clave_cif);

                desCesar descifra = new desCesar(n,clave_cif);
                clave = descifra.Descifra();
                System.out.println(clave);
        }
}
```

Código de descifrado **desCesar.java**

```
/* Fichero desCesar.java
  *@author Jose Manuel Barba Gonzalez
  *@version 1.0
  *Programacion Concurrente y de Tiempo Real
  *Area de CC. de la Computacion e I.A.
*/

import java.io.*;
import java.util.*;
import java.lang.*;

public class desCesar
{
        private int n;
        private String clave;

        public desCesar(int num, String cad)
        {
                n = num;
                clave = new String(cad);
        }
        public int Numero(){return n;}
        public String Clave(){return clave;}

        public String Descifra()
        {
                int i, value;
                char []tmp;
                char []tmpcif;
                String clave_cif;
```

```
                System.out.println("Funcion E(x) = x - n mod 27");
                tmp = new char[clave.length()];
                tmp = clave.toCharArray();
                tmpcif = new char[clave.length()];
                for(i = 0; i < clave.length(); i++)
                {
                        value = (tmp[i] - n %27);
                        tmpcif[i] = (char) value;
                }
                clave_cif = new String(tmpcif);
                //System.out.println(clave_cif);
                return clave_cif;
        }
}
```

5. Problema 6

Problama que genera números aleatorios y los muestra por pantalla.

5.1. Enunciado

Una tarea que será de utilidad durante el curso es la generación de números aleatorios. En Java, esto puede lograrse de dos maneras 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.

5.2. Solución

```
/* Fichero aleatorios.java
  *@author Jose Manuel Barba Gonzalez
  *@version 1.0
  *Programacion Concurrente y de Tiempo Real
  *Area de CC. de la Computacion e I.A.
*/

import java.io.*;
import java.util.*;
import java.lang.*;

public class aleatorios
{
        public static void main(String[] args)
        throws Exception
        {
                int n;
                double a;
                Scanner numero;

                System.out.println("Introduzca el numero para generar
                la secuencia aleatoria(minimo 10): ");
```

```
                numero = new Scanner(System.in);
                n = numero.nextInt();
                if(n < 10) n = 10;

                for(int i = 0; i < n; i++)
                {
                        a = Math.random();
                        System.out.println("n "+(i+1)+": "+a);
                }
        }
}
```

6. Bibliografía

Referencias

[1] Apis de docs oracle. `http://docs.oracle.com/javase/8/docs/api/`

[2] Apuntes de JavaBasico.pdf.

[3] Resumen diapostivas Tema21.pdf.

[4] Resumen diapostivas Tema22.pdf.

[5] Resumen diapostivas Tema23.pdf.