

Práctica 4: Clases y algoritmos

Ejercicio 4.1: Funciones de clase y variables de clase

En este ejercicio se ve la definición y llamada a una función (o método) de una clase en Java.

Se definen también variables de clase (variables miembro) y variables en las funciones (variables locales). La variable *area*, a diferencia de lo que dicen las guías de estilo, tiene el mismo nombre en los tres casos, ya que se quiere observar cuando y cómo se utiliza cada una. La variable de clase también es *static*.

```
class Ejer1 {
    static double area;    // variable de la clase

    public static void main(String args[]){
        double radio;    // variable de la funcion main
        double area;    // variable de la funcion main

        if ( args.length < 1){
            System.out.println("Pasar el radio. Ej. : >java Ejer6 12.5");
            System.exit(0);
        }
        radio = Double.parseDouble(args[0]);
        area = Ejer1.Area(radio);
        System.out.println("Area del circulo r="+radio + "m. = " + area+" m2");
        area = Ejer1.area;
        System.out.println("Area del circulo r="+radio+"m. = "+area+" Has.");
    }

    public static double Area ( double rd) {
        double area; //variable local de la funcion Area
        //usar la variable local
        area = Math.PI*Math.pow(rd,2.0);
        //acceder a la variable de la clase
        Ejer1.area = area/10000; //obtener el area en Has.
        return area; //devuelve el valor
    }
}
```

Observa que la variable **area**, declarada como **double**, ha sido definida tanto en la clase como dentro de los métodos **main()** y **Area()**. La variable **area** de la clase puede ser accedida por cualquier método (o función) de la clase, no así las variables declaradas dentro de cada método, ya que éstas son locales.

El método estático **pow()** de la clase **Math** eleva el primer argumento a la potencia del segundo.

Introducir la función **Perimetro()** para el cálculo del perímetro en metros y kilómetros, e imprimir ambos valores.

Ejercicio 4.2: Algoritmo de ordenación

El siguiente ejercicio llama a una función que realiza la ordenación de los números que se dan como argumentos. La utilización del programa es:

```
java Ejer2 5 7 2 45 12 5 -4
```

El código del programa es el siguiente:

```
class Ejer2 {
    // variables miembro de la clase Ejer2
    static int Numero;
    static int[] Datos; //array para almacenar los numeros

    public static void main(String [] args) {
        Ejer2.Numero = args.length;
        Ejer2.Datos = new int[Ejer2.Numero];

        for (int i=0;i< Ejer2.Numero;i++ ) {
            Ejer2.Datos[i] = Integer.parseInt(args[i]);
        }

        //Ahora los ordenaremos de menor a mayor
        Ejer2.Ordena(); //llamamos a la funcion ordena
    }

    public static void Ordena() {
        System.out.println("\nLos numeros sin ordenar son: ");
        for (int i=0;i<Numero;i++)
            System.out.println("Num: "+i+" valor: "+Datos[i]);
        //usaremos un sistema de ordenamiento sencillo
        for (int i=0;i < Numero-1;i++) {
            for (int j=i+1;j<Numero;j++) {
                if ( Datos[i] > Datos[j]) { // hay que permutar los valores
                    int tmp = Datos[j];
                    Ejer2.Datos[j] = Datos[i];
                    Ejer2.Datos[i] = tmp;
                }
            }
        }
        System.out.println("\nLos numeros ordenados de menor a mayor son: ");
        for (int i=0;i<Numero;i++)
            System.out.println("Num: "+i+" valor: "+Datos[i]);
    }
}
```

Cambiar el programa para que los números en lugar de recibirlos como argumentos los genere de forma aleatoria con valores entre 0 y 100. La cantidad de números a generar se introduce como argumento del programa. De esta forma el comando:

```
java Ejer2 10
```

Genera 10 números aleatorios y los ordena.

Ejercicio 4.3: Utilización de clases.

En Java no existe como tipo predefinido el número complejo. Para realizar operaciones con complejos se utiliza en este ejercicio una clase que define los números complejos y sus operaciones.

La clase se puede descargar de:

<http://www.cs.princeton.edu/introcs/32class/Complex.java>

Se pide realizar:

1. Un programa que cree instancias de dos números complejos (por ejemplo $3 + 2i$ y $5 - 8i$) y realice su suma, su multiplicación, el conjugado de uno de ellos y el producto de uno por su conjugado. Se puede utilizar como modelo la función `main()` que está en la propia clase `Complex.java`.
2. Crear un array de 10 posiciones de números complejos.
3. A continuación, realizar un bucle que rellene las 10 posiciones con números complejos aleatorios, con la parte real e imaginaria entre 0 y 10.
4. Un bucle que multiplique cada número por su conjugado. Se puede comprobar que la parte imaginaria es nula.

Ejercicio 4.4: Creación de un método de una clase.

En la clase anterior `Complex.java`, crear el método estático **`Ordena(Complex[])`**, que recibe un vector de números complejos y los ordena. Utilizar dicha función con el vector de números complejos creados en el paso 3 del ejercicio anterior. La función no tiene que devolver ningún valor, por lo que se definirá como **`void`**.

La ordenación se realiza en base al módulo del número que se obtiene con la función **`abs()`** de la clase **`Complex`**.

Ejercicio 4.5: Utilización y definición de clases.

En este ejercicio se pide realizar lo mismo que en el ejercicio 4.3 pero utilizando otra implementación de la clase `Complex` basada en coordenadas polares. Se puede obtener en la dirección:

<http://www.cs.princeton.edu/introcs/33design/Complex.java>

Será necesario guardarla en un directorio distinto que la anterior por tener el mismo nombre.

El primer paso es comprobar las operaciones ya definidas, suma y multiplicación, con el mismo código anterior pero utilizando esta clase, para ello el código que utiliza esta clase debe estar en el mismo directorio.

Se debe completar la definición de la clase con la función conjugado. Nota: el conjugado del complejo $(r, \text{ángulo})$ es $(r, -\text{ángulo})$.

Crear el array de 10 posiciones y los dos bucles que lo rellenan y multiplican por el conjugado.