

PRACTICA 3: Introducción a la programación en Java

Factor de ponderación [0-10]: 5

3.1. Objetivos

Profundizar de forma práctica en el conocimiento de Java, realizando pequeños programas en este lenguaje.

3.2. Ejercicios

Los ejercicios que se plantean son independientes entre sí. Se propone finalizar todos ellos, realizándolos en un directorio separado para cada uno, y conservar los ficheros con el código fuente. Todos los programas deben tener comentarios de cabecera y utilizar comentarios de `javadoc` para documentar las funcionalidades más relevantes. Asimismo, los códigos deben seguir las reglas de codificación expuestas en el documento [1] que tienen a su disposición en el aula virtual de la asignatura.

[1] Code Conventions for the Java Programming Language.
<http://www.oracle.com/technetwork/java/codeconv-138413.html>

3.2.1. Ejercicio 1: la clase Racional

Construir una clase Racional que permita representar y manipular números racionales. Un número racional permite representar la relación a/b entre dos números enteros siendo a el numerador y b el denominador (distinto de cero). Se deben implementar **al menos** los siguientes métodos para la clase Racional:

1. `String toString()`
2. `Racional suma(Racional b)`
3. `Racional resta(Racional b)`
4. `Racional producto(Racional b)`
5. `Racional division(Racional b)`

Para detalles sobre la aritmética de números racionales, puede consultar la Wikipedia [2].

[2] <http://xurl.es/b3q7q>

3.2.2. Ejercicio 2: la clase String

Desarrollar un programa que solicite introducir una cadena de caracteres por teclado y visualice por pantalla el número de caracteres a y A que contiene dicha cadena.

Ejemplo de ejecución y salida del programa:

```
$>java NumeroA
Introduce una cadena de caracteres: Este caballo se llama America
El numero de caracteres a o A es: 6
```

3.2.3. Ejercicio 3: Vectores en Java

Desarrollar un programa que devuelve el rango (diferencia entre el valor máximo y el mínimo) de los elementos de un array de números reales.

Los datos contenidos en el vector han de leerse de un fichero cuyo nombre se le pase al programa en la línea de comandos. El programa debe simplemente, imprimir en pantalla la diferencia entre máximo y mínimo.

3.2.4. Ejercicio 4: Precisión en punto flotante

Los números reales representados en punto flotante tienen una precisión limitada y las operaciones que se realicen con ellos pueden generar errores de redondeo. Por ejemplo, en una división, en el cálculo de la raíz cuadrada de un número, etc. Esto hace que puedan surgir problemas a la hora de comparar dos valores numéricos reales mediante una relación de igualdad. En su lugar se debería comparar si están lo suficientemente cercanos, es decir, si el valor absoluto (mejor, relativo) de la diferencia es menor de un determinado valor muy pequeño.

Matemáticamente esta situación puede expresarse de la siguiente forma:

$$\frac{|x-y|}{\max(x,y)} < \epsilon$$

Para evitar problemas con la división en el caso de que el divisor sea muy cercano a cero, la expresión anterior puede transformarse en:

$$|x - y| \leq \max(x, y) * \epsilon$$

En base a lo anterior, construir un programa `Casicero` que devuelva verdadero o falso si dos valores numéricos reales dados como parámetros son iguales o no.

Pueden emplearse las rutinas predefinidas de la clase `Math` de Java que sean necesarias.

Una ejecución del programa debería producir la siguiente salida por pantalla:

```
$>java CasiCero  
10.0 y 10.0000000000000002 son iguales: false  
10.0 y 10.0000000000000000 son iguales: true
```

3.2.5. Ejercicio 5: fechas

Desarrollar un programa que devuelva la edad de una persona. Los parámetros del programa son dos fechas que han de leerse en la línea de comandos: la fecha actual y la fecha de nacimiento de la persona. Un ejemplo de entrada para el programa sería:

```
$>java Edad 20/02/1985 13/02/2013
```