



25 de septiembre de 2017

En esta primera práctica vamos a

- Establecer el procedimiento de crear/desarrollar/almacenar las prácticas de la asignatura.
- Repasar el uso del IDE netbeans para programar en Java.
- Reconocer algunos errores o erratas de programación típicos.
- Utilizar el paquete `java.math` y objetos `BigDecimal` para trabajar con aritméticas de distinta precisión, midiendo algunos errores de redondeo en cálculos y en procesos iterativos.

1 INSTRUCCIONES:

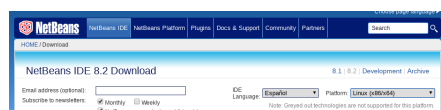
Todo está en internet

- Para instalar JavaSE JDK + netbeans :
<http://www.oracle.com/technetwork/java/javase/downloads>



NetBeans with JDK 8

Para instalar sólo netbeans: <http://netbeans.org>



Algunos repositorios de Linux también tienen paquetes de instalación de los dos programas.

- Un buen tutorial de Java: enlazar con javahispano.org haciendo la búsqueda en Google de «java básico con ejemplos»
- Ante cualquier duda, consultar el API de Java. Para ello, será suficiente con buscar en Google «(clase) API java 8» donde (clase) es el nombre de la clase que contiene la variable o el método acerca del cual tenemos alguna duda.

Por ejemplo, hacer la búsqueda con « Math API java 8 ».

¿Dónde vamos a ir poniendo las prácticas?

Con el programa IDE `netbeans` vamos a crear la carpeta (PROYECTO) donde se irá guardando el trabajo de la prácticas de todo el curso. Cuando trabajemos en las prácticas en cada sesión, abriremos nuestro PROYECTO con `netbeans`, iremos incorporando los programas nuevos utilizando un PAQUETE para cada práctica y al finalizar guardaremos nuestro PROYECTO antes de salir de `netbeans`.

Para tener acceso rápido a clases y métodos susceptibles de ser utilizados en distintos problemas, crearemos un PAQUETE auxiliar donde tenerlos ubicados.

Utilizaremos el AULA VIRTUAL de la UMU para comunicarnos y para guardar nuestro trabajo. Para guardar copias de vuestro trabajo también podéis usar vuestro espacio `UmuBox`¹ de la UMU. (Podréis utilizar unidades de memoria USB para tener copias adicionales, pero **no es recomendable trabajar directamente sobre estas unidades**)

Si trabajamos en las ADLAS de la Facultad, antes de empezar deberemos DESCARGAR nuestro proyecto desde el AULA VIRTUAL y cuando terminemos deberemos guardarlo SUBIENDOLO otra vez a nuestro `UmuBox` o al AULA VIRTUAL.

Una vez creado, el PROYECTO, es una carpeta que contiene carpetas y ficheros. Para mover la carpeta, utilizaremos el compresor de ficheros que tengamos en el ordenador para crear un solo fichero comprimido que contendrá todo el PROYECTO.

Ejercicios

Ejercicio 1

1. Descarga desde la zona de recursos del AULA VIRTUAL el fichero `CN1vEjemplosIniciales.zip` y descomprímelo. Una vez descomprimido ya tienes a tu disposición un PROYECTO de `netbeans`. Inicia el programa `Netbeans`;
2. Abre el proyecto `CN1vEjemplosIniciales` y renómbralo como `20170925CN1vXXXXX`, donde `XXXXX` será nuestro nombre y uno o los dos apellidos (**¡SIN ESPACIOS EN BLANCO Y SIN ACENTOS!**). Para hacerlo
 - (a) seleccionaremos la opción « Archivo / Abrir Proyecto » (`File / Open Project`) del menú, buscaremos el proyecto `CN1vEjemplosIniciales` y pulsaremos en el botón «Abrir proyecto»
 - (b) Para renombrar, en la ventana «Proyectos» de `netbeans`, marcaremos el proyecto `CN1vEjemplosIniciales` y pulsando el botón del ratón aparecerá un menú desplegado en el que seleccionaremos la opción «renombrar» (`refactor/rename`).
 - (c) Escribiremos en la ventana del « nombre del proyecto » (`project name`) el nuevo nombre `20170925CN1vXXXXX`;
Marcaremos también la opción de «Renombrar también la carpeta del proyecto» (`rename the project folder`) y pulsaremos el botón «Cambiar de nombre» (`refactor`).

Ahora tendremos una carpeta en el escritorio con el nombre de nuestro proyecto `20160926CALNUM1XXXXX`. En condiciones generales, gestionaremos esta carpeta desde la ventana de navegación de `Netbeans` o desde su menú principal. En la ventana de proyectos será suficiente COLOCAR EL PUNTERO DEL RATÓN SOBRE EL ELEMENTO (proyecto, paquete o fichero) Y PULSAR EL BOTÓN DERECHO DEL RATÓN para trabajar desde el menú desplegable o simplemente arrastrar ficheros para mover o copiar ficheros.

3. Dentro del proyecto crea un paquete con el nombre `practical` donde vamos a ir creando los ficheros de esta práctica. Para hacerlo
 - (a) seleccionaremos la opción « Archivo / Archivo Nuevo » (`File / New File`) del menú
 - (b) marcaremos la opción «Java / paquete de Java» (`Java/ Java Package`) y pulsaremos «Siguiente» (`Next`)
 - (c) Escribiremos en la ventana del « nombre del paquete » (`Package name`) `practical`; mantendremos las selecciones relativas a la «ubicación del paquete» (`location`), y para finalizar pulsaremos el botón «Terminar» (`Finish`)
4. Usando el botón derecho del RATÓN marca el paquete con el nombre `practical` y selecciona la opción « Nuevo / Clase Main de Java » (`New/ java main class`). Si no aparece la opción, selecciona «Nuevo / Otro» (`New/ Other`) y luego en tipo de archivo eliges «Java / Clase main de java» y pulsas «Siguiente» (`Next`).

Ahora escribe en la ventana del «Nombre de la clase» `Ejercicio1.java` (ATENCIÓN A LA SINTAXIS) y para finalizar pulsa el botón «Terminar» (`Finish`)

Escribe² el código siguiente en el fichero `Ejercicio1`:

¹`UmuBox` es la «nube» de la Universidad de Murcia accesible desde cualquier ordenador o dispositivo móvil con la aplicación de cliente de «ownCloud».

²Si tienes abierto el PDF con la práctica usa las herramientas de edición: CORTAR y PEGAR, en lugar de escribir el texto.

```
//public class Ejercicio3 {
    public static void main(String[] args) {
        // Explica los resultados de las siguientes instrucciones.
        double[] a = {12.6, 2.3, 6.28};
        double[] c=cambiar(a);
        double[] d=nocambiar(a);
        System.out.println(c[0]);
        System.out.println(d[0]);
    }
    public static double[] cambiar(double[] b) {
        double[] c = new double[b.length];
        c = b;
        c[0] = 3455.23;
        return c;
    }
    public static double[] nocambiar(double[] d) {
        double[] e = new double[d.length];
        e = d;
        return e;
    }
}
//}
```

Ejecuta esta clase y añade los comentarios oportunos explicando los resultados de la ejecución y los cambios realizados para obtener otro resultado.

¡Busca información sobre `System.arraycopy`!

Ejercicio 2 (Fórmula de Heron) El área de un triángulo de lados a, b y c , con $a \geq b \geq c$ puede expresarse mediante la fórmula

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

donde $s = (a + b + c)/2$ es el semiperímetro.

En esta fórmula puede aparecer pérdida de precisión cuando el triángulo es muy plano ($a \approx b + c$) porque s y a están próximos. El problema de trasladar la resta de números próximos a c que por ser menor que a puede dar lugar a una menor pérdida de precisión es reescribir la fórmula anterior en función del perímetro

$$A = \sqrt{(a+b+c)(c-(a-b))(c+(a-b))(a+(b-c))}$$

Crea una clase ejecutable, `Ejercicio2.java` y escribe un bucle en el que al ejecutarla, se evalúen y muestren las dos fórmulas y el error relativo entre las dos aproximaciones al área con los valores

$a = 9$; $e = 0.5E - k$ ($k = 0, 1, \dots, 10$), $b = 5.5 + e$ y $c = 3.5 + e$.

Analiza los resultados.

Cambia los valores de a, b y c para que el triángulo no sea demasiado plano y observa como no debe haber diferencias apreciables en las evaluaciones de las dos fórmulas.

Ejercicio 3 (Números de Máquina) Una máquina de números en base 10, está determinada por la precisión que es el número de dígitos a considerar en la mantisa y el método de redondeo a utilizar para las operaciones aritméticas o de asignación. Java nos ofrece en la distribución JDK el paquete `java.math` que contiene cuatro clases que permiten trabajar con cualquier precisión. La idea es definir la mantisa como un entero de tantos dígitos como señale la precisión escalado (multiplicado) por una potencia de 10.

Completad (sin copiar y pegar, **TECLEANDO MUY DESPACIO**), el código que sigue en una clase principal de `Ejercicio3.java` y analizar el comportamiento de los distintos comandos (acudid a las API de java si es necesario)

```
package practical;

import java.math.BigDecimal;
import java.math.MathContext;
import java.math.RoundingMode;

public class Ejercicio3 {

    public static void main(String[] args) {
        int precision=3;
        MathContext mc=new MathContext(precision,RoundingMode.HALF_EVEN);
        System.out.println("Math.PI =" + Math.PI);
        BigDecimal a=new BigDecimal(Math.PI, mc);
        System.out.println("BigDecimal a = " + a);
        System.out.println("Math.PI - BigDecimal a = " + (Math.PI - a.doubleValue()));
    }
}
```

Prueba a cambiar la precisión a 20, 35, o 150. ¿Qué observas?

A continuación, utilizando un contexto matemático `MathContext` que trabaje con 4 dígitos de precisión y redondeo a la paridad (ver el código anterior)

1. Calcula los valores de las siguientes operaciones y los errores absolutos y relativos cometidos:

$0.6688 \oplus 0.3334$; $1000 \ominus 0.05001$; $2.000 \otimes 0.6667$; y $25.00 \oslash 16.00$.

- Busca ejemplos de números de máquina a, b, c tales que $(a \oplus b) \oplus c \neq a \oplus (b \oplus c)$.
- Busca ejemplos de números de máquina a, b tales que $a \neq b$ y $1 \oslash a = 1 \oslash b$.
- Busca ejemplos de números de máquina a, b, c tales que $a \otimes (b \oplus c) \neq (a \otimes b) \oplus (a \otimes c)$.
- Resuelve la ecuación $x^2 + 800x + 1 = 0$ utilizando la fórmula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. Evalúa los errores cometidos. Recalcula las raíces utilizando fórmulas apropiadas que reduzcan los errores.

(MathContext permite trabajar con números de 128bits ver API)

Ejercicio 4 El propósito de este ejercicio es comprobar distintas formas de aproximar el número irracional $\sqrt{2}$ y en último extremo encontrar los 1000 primeros dígitos decimales exactos de su representación decimal

- Empezamos utilizando la representación en serie de potencias de la función

$$f(x) = (1+x)^{1/2} = \sum_{k=0}^{+\infty} \binom{1/2}{k} x^k$$

$$\text{donde } \text{binom}[k] = \binom{1/2}{k} = \frac{0.5 * (0.5 - 1) * (0.5 - 2) * \dots * (0.5 - k + 1)}{k * (k - 1) * (k - 2) * \dots * 1}.$$

- Simplifica el cociente $\text{binom}[k]/\text{binom}[k-1]$ para tener una fórmula recurrente que evalúe los números $\text{binom}[k]$.
- Trabajando con números `double` y la acotación del resto para las series alternadas, cuántos sumandos son necesarios para aproximar

$$\text{raizAp} \approx \sqrt{2} = f(1)$$

- Prueba con $n=1000$ sumandos o $n=10000$ sumandos para ver que precisión obtienes. Mide también el error $|2 - \text{raizAp}^2|$
- Intenta hacer los cálculos más estables sumando por separado los terminos positivos y los negativos y haciendo la suma con los terminos de menor a mayor tamaño para ver si mejoran los resultados.

- En segundo lugar utiliza la representación en serie de potencias de la función

$$g(x) = (1+x)^{-1/2} = \sum_{k=0}^{+\infty} \binom{-1/2}{k} x^k$$

para aproximar $\sqrt{2} = g(-1/2)$.

En este caso los sumandos de la serie tienen signo constante y la serie converge bastante más rápidamente. Repite los pasos del apartado anterior con esta nueva serie para comprobarlo.

- Por último, buscando formas de aproximar fáciles y de implementar con gran precisión podemos acudir a las tablillas babilónicas de hace más de 9000 años para encontrar aproximaciones muy precisas con los primeros términos de la sucesión

$$b_0 = 1; \quad b_n = \frac{1}{2} \left(b_{n-1} + \frac{2}{b_{n-1}} \right) = b_{n-1} + \frac{1}{2} \left(\frac{2}{b_{n-1}} - b_{n-1} \right).$$

Esta sucesión coincide con la que proporciona el método de Newton aunque se puede analizar directamente observando que si $0 < a < \sqrt{2}$ entonces $0 < a < \sqrt{2} < 2/a$ y que si $\sqrt{2} < a$ entonces $0 < 2/a < \sqrt{2} < a$. Se puede precisar más y comprobar que la sucesión b_n es decreciente a partir del segundo término y converge hacia $\sqrt{2}$.

Comprueba la bondad de estas aproximaciones con números de tipo «double».

Usando números de la clase «BigDecimal» calcula los 1000 primeros dígitos de la representación decimal de $\sqrt{2}$

Para finalizar, (alrededor de 5 minutos antes de acabar la práctica) tienes que guardar todo tu trabajo. Para ello vas a seguir el siguiente proceso obligatorio

Cierra el programa NetBeans después de guardar todos los ficheros.

Con el explorador marca la carpeta donde está tu proyecto para crear un fichero comprimido que contenga todos los ficheros dando al botón de la derecha y buscando la opción correspondiente.

Por último, sube el fichero comprimido a tu sitio en el AULA VIRTUAL de la UMU o a tu nube UmuBox.