



UNIVERSIDAD DE MURCIA

DEPARTAMENTO DE MATEMÁTICAS

MÉTODOS NUMÉRICOS. PRÁCTICA 1

INSTRUCCIONES:

El objetivo de esta práctica inicial es repasar el uso del entorno Netbeans para programar en JAVA, al tiempo que aprovecharemos para recordar algunos errores típicos de programación en este lenguaje.

Para empezar hemos creado un proyecto de JAVA con el nombre `MetodosNumericos`. Para ello, tras abrir el programa Netbeans, pinchamos **File/New Project...**, luego pulsamos **Next**, y en la ventana que aparece escribimos `MetodosNumericos` en **Project Name** y el directorio en que vamos a trabajar en **Project Location**. Luego desmarcamos las casillas **Set as Main Project** y **Create Main Class** y por último pulsamos **Finish**.

Este proyecto constará de cuatro grandes subproyectos (computación, métodos lineales, ecuaciones polinómicas, e interpolación y aproximación) que iremos desarrollando a lo largo de estas prácticas.

Dentro del proyecto hemos creado un paquete con el nombre `computacion.practica1` donde vamos a ir colocando los ficheros de esta práctica. Para ello, dentro de la carpeta **Projects** de NetBeans y usando el botón derecho del ratón marcamos nuestro proyecto recién creado, seleccionamos la opción **New** y luego la opción **Java Package...** En la ventana emergente escribimos `computacion.practica1` en **Package Name** y luego pulsamos **Finish**.

A continuación fuimos creando los ficheros que aparecen en el paquete. Por ejemplo, para crear los ficheros `ExpedienteXno1.java` y `AuxExpedienteXno1.java` hicimos lo siguiente:

- (i) Dentro de la carpeta **Projects** de NetBeans y usando el botón derecho del ratón marcamos el paquete `computacion.practica1`. Luego seleccionamos la opción **New** y luego la opción **File/Folder...**
- (ii) Dentro de la ventana emergente, en el marco titulado **Categories**, seleccionamos **Java Classes** y en el marco **File Types** seleccionamos **Java Main Class** (clase ejecutable de JAVA).
- (iii) Ahora en la ventana **Class Name** escribimos `ExpedienteXno1` (¡cuidado con la sintaxis!) y presionamos el botón **Finish**.
- (iv) Se creó así el fichero `ExpedienteXno1.java` con parte del código ya escrito.
- (v) Completamos el contenido de `ExpedienteXno1.java`.
- (vi) Repetimos los pasos anteriores pero seleccionando al final, en la ventana **File Types**, la opción **Java Class**. En la ventana **Class Name** escribimos `AuxExpedienteXno1` (¡cuidado con la sintaxis) y presionamos el botón **Finish**.

Análogamente creamos las clases `ExpedienteXno2`, `ExpedienteXno3`, `ExpedienteXno4`, `ExpedienteXno5`, `ExpedienteXno6` (del tipo **Java Main Class**) y `AuxExpedienteXno4` (del tipo **Java Class**).

Todas las aplicaciones compilan y ejecutan correctamente, pero no producen los resultados que aparentemente deberían dar. El objetivo de la práctica es reemplazar las clases de JAVA por otras en las que se hayan corregido los errores, añadiendo en cada fichero comentarios que aclaren la razón de los cambios. Las nuevas aplicaciones y métodos conservan los nombres anteriores pero ahora se añade "Solucion" al principio (así, `ExpedienteXno1.java` pasa a `SolucionExpedienteXno1.java`). Por ejemplo, para modificar la clase `ExpedienteXno1.java` realizaremos el siguiente proceso:

- (i') Dentro de la carpeta **Projects** de NetBeans y usando el botón derecho del ratón marcamos el fichero `ExpedienteXno1` que se va a modificar y seleccionamos la opción **Copy**.
- (ii') Marcamos ahora, también con el botón derecho del ratón, el paquete `computacion.practica1` y seleccionamos la opción **Paste**, con lo que se habrá creado una copia del fichero `ExpedienteXno1` con el nombre `ExpedienteXno1_1`.

(iii') Sólo nos queda renombrar la copia que hemos realizado. Para ello **NO VAMOS A USAR EL RATÓN CON UN CLICK** (en el botón izquierdo) **PARA REESCRIBIR EL NUEVO NOMBRE**. En vez de esto, vamos a marcar el fichero usando el botón derecho y usaremos la opción **Refactor** y luego **Rename....** Se abrirá una ventana en la que escribiremos el nuevo nombre `SolucionExpedienteXno1` (comprobar que está marcada la casilla **Apply Rename on Comments**) tras lo que pulsaremos **Next**. Ahora tendremos una nueva ventana donde se muestran los cambios en los atributos de la clase que se van a realizar automáticamente y pulsamos **Do Refactoring**. Tendremos renombrado el fichero y modificado el código interno correspondiente al nuevo nombre.

EJERCICIOS:

1. La aplicación `ExpedienteXno1.java` usa `AuxExpedienteXno1.java`, que a su vez consta de los métodos `double[] cambiar(double[] b)` y `double[] nocambiar(double[] d)`. El primero reemplaza la primera coordenada del vector por 3455,23, y el segundo no altera el vector. Por tanto, cuando en la aplicación instanciamos un objeto `aux` de la clase, aplicamos sucesivamente los métodos `cambiar` y `nocambiar` al vector (12.6, 2.3, 6.28) y escribimos las respectivas primeras coordenadas de los vectores resultantes, deberíamos obtener 3455.23 y 12.6. Sin embargo se obtiene dos veces el valor 3455.23.
2. La aplicación `ExpedienteXno2.java` debería escribir los enteros del 20 al 0 en sentido descendente pero no hace nada.
3. La aplicación `ExpedienteXno3.java` define la variable booleana `a` como `false` y a continuación escribe “a es verdadera” si el valor de `a` es `true` y “a es falsa” si el valor de `a` es `false`. Por tanto debería escribir el mensaje “a es falsa”. Sin embargo escribe “a es verdadera”.
4. La aplicación `ExpedienteXno4.java` usa `AuxExpedienteXno4.java`, que a su vez contiene los métodos `void sumar89Matrices(int[] b)` y `void sumar89Enteros(int d)`. El primer método imprime el resultado de sumarle 89 a la primera coordenada del vector `b`, y el segundo el resultado de sumarle 89 al entero `d`. La aplicación `ExpedienteXno4.java` implementa dos veces el método `sumar89Matrices` al vector (1, 2, 7) y dos veces el método `sumar89Enteros` al número 1 y por tanto debería aparecer cuatro veces en pantalla el número 90. Sin embargo la lista que aparece es 90, 179, 90, 90.
5. La aplicación `ExpedienteXno5.java` debería imprimir el número $3/2-1$, es decir, 0.5, pero imprime 0.0.
6. La aplicación `ExpedienteXno6.java` debería imprimir los números 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0 y detenerse, pero sigue imprimiendo números indefinidamente.