



SESIÓN 01: Creación y ejecución de aplicaciones Java. Trabajo con repositorios

Objetivos

- Familiarizarse con el entorno de desarrollo **ECLIPSE**.
- Introducir las perspectivas JAVA y SVN del entorno.
- Conocer las fases de creación y ejecución necesarias de programas en Java y llevarlas a cabo en el entorno.
- Familiarizarse en el uso de repositorios y las operaciones sobre ellos de *Commit* y *Update*.
- Implementar algoritmos sencillos que hagan uso de asignaciones, expresiones y salida de información. Utilizar algunos métodos de la clase **Math**.

Ejercicios propuestos

Utilización del entorno de programación de ECLIPSE. Es recomendable que la Workspace para esta asignatura sea **IP2022**, y debe tener un *Text file encoding* **UTF-8**. Una vez hecho esto, el alumno debe crear un **proyecto Java**, donde se realizarán todas las sesiones de la asignatura y conectarlo con su repositorio personal SVN donde subirán todos los ejercicios resueltos.

El **nombre del proyecto Java del alumno** será:

apellidos y nombre del alumno seguido de **IP2022**, e.j.: **CorralLiriaAntonioLeopoldoIP2022**

Tal y como se verá en el archivo *GuiaRapidaRepositorios.pdf*, cada alumno tiene asignado un **repositorio personal** con un nombre similar a **corralliriaantonioleopoldoip2022**, es decir, los apellidos y nombre en minúscula seguidos de **ip2022**, sin espacios.

Es necesario que el proyecto Java del alumno se suba (**Commit**) al repositorio personal después de cada sesión de prácticas. Estos son los pasos para “conectar” el proyecto Java del alumno con su repositorio personal:

1. En primer lugar tenemos que crear el enlace al repositorio personal desde la *perspectiva SVN* (si aún no lo hemos hecho).

2. Después creamos el proyecto Java para esta asignatura en nuestro *workspace* (perspectiva Java) con el nombre referido anteriormente **CorralLiriaAntonioLeopoldoIP2022**

3. Por último, “conectamos” nuestro proyecto Java con el repositorio personal de este modo:

En la perspectiva Java, sobre el nombre del proyecto de la asignatura, hacemos clic con el botón derecho y seleccionamos: **Team > Share Project... > SVN** (Next) >

y lo vinculamos al enlace a nuestro repositorio personal, que en nuestro ejemplo es

<http://svndoc.ual.es:9090/svn/corralliriaantonioleopoldoip2022>

pulsamos (Next) > y seleccionamos **Advanced Mode** (Next) > **Finish**.

Es muy importante que se seleccione **Advanced Mode**, pues de no hacerlo así el proyecto no colgaría del *Trunk* y entonces no estaría bien enlazado.

Dentro del proyecto, en la carpeta de fuentes **src**, se irán creando **paquetes Java** para cada una de las sesiones. Los nombres de los paquetes deben mantener las normas de estilo propias del lenguaje Java (sólo usar minúsculas para los paquetes).

El esquema de nombrado de **paquetes** será:

org.ip.sesionnumero Ejemplo: **org.ip.sesion01**

En ese paquete se crearán todos los programas que se proponen en la sesión dándoles un nombre alusivo a lo que realiza y que se indica en cada ejercicio entre paréntesis y en negrita.

Ejercicios Propuestos

1. Desarrolle un programa en Java (**DivisionPorCero**) que muestre por pantalla el cociente y el resto de una división por cero con enteros y reales.

Ejemplo de ejecución:



EJEMPLOS DE DIVISIONES POR CERO CON ENTEROS Y REALES

17.0 / 0.0 = Infinity

17.0 % 0.0 = NaN

17 / 0 =

Exception in thread "main" java.lang.ArithmeticException: / by zero
at org.pi.sesion01.DivisionPorCero.main(DivisionPorCero.java:20)

2. Desarrolle un programa (**ConversorEurosPesetas**) que convierta el valor de una cantidad en Euros (E) a Pesetas (P), y viceversa.

$$P = \lfloor E \times 166.386 \rfloor$$

$$E = P / 166.386$$

Ejemplo de ejecución:



131,25 euros => 21838 pesetas

3247 pesetas => 19,51 euros

3. Desarrolle un programa (**Distancia**) que muestre la distancia de un punto (x, y) al origen (0, 0), donde x e y son enteros. Comprueba con distintas ejecuciones las distancias que se indican a continuación. Distancia de (3, 4) a (0, 0) = 5.0, distancia de (5, 12) a (0, 0) = 13.0, distancia de (2, 1) a (0, 0) = 2.23606797749979. Utiliza la clase **Math**.¹

Ejemplo de ejecución:



La distancia del punto (2, 1) al punto (0, 0) es 2.23606797749979

¹ Para usar las funciones matemáticas que necesites, puedes utilizar los métodos de la clase **Math**. No es necesario importar dicha clase en un programa para usar sus métodos. Esta clase forma parte del paquete java.lang que se importa automáticamente. Algunos de los métodos que puedes necesitar son:

sqrt(x) raíz cuadrada de x . Ejemplo: sqrt(900.0) es 30.0

pow(x, y) x elevado a y. Ejemplo: pow(2,7) es 128. pow(9, .5) es 3

En el 4º ejercicio se usará **random**. Para utilizarlos daremos <nombre de la clase>.<nombre del método>(<parámetros>).

Ejemplo: **System.out.println(Math.sqrt(c1 + d *f));**

Con c1 = 13, d = 3.0 y f = 4.0 mostraría 5.0

Trabajo autónomo

4. Desarrolle un programa (**EnteroAleatorio**) que muestre un entero pseudo-aleatorio entre -15 y 15. Pruebe distintas ejecuciones. Utiliza el método *random()* de la clase **Math**. Debe tener en cuenta que dado un número *aleatorio* generado por el método *random()* ($0 \leq \text{aleatorio} < 1.0$) para conseguir un número entero entre M y N con $M < N$ y ambos incluidos, se debe usar la siguiente fórmula:

$$\text{Valor Entero} = \lfloor \text{aleatorio} * (N - M + 1) + M \rfloor$$

Para revisar el uso del método *random()* se puede visitar la siguiente dirección Web

http://chuwiki.chuidiang.org/index.php?title=Generar_n%C3%BAmeros_aleatorios_en_Java

Ejemplo de ejecución:



```
Vamos a generar un entero aleatorio entre -15 y 15  
El entero generado aleatoriamente es: -1
```

5. Dado el *radio* de una circunferencia, desarrolle un programa en Java (**Circunferencia**) que permita calcular la longitud de dicha circunferencia y el área del círculo que hay en su interior. Para ello, debe tener en cuenta que el *radio* es una variable real, que el valor de la constante π que está en la librería **Math** (**Math.PI**) y que hay que utilizar las siguientes fórmulas (siendo *r* el radio). Muestre por pantalla los mensajes exactamente como se indica a continuación, para ello recuerde como muestra Java datos por pantalla con la función **printf**.

$$\text{Diametro} = 2r$$

$$\text{Longitud} = 2\pi r$$

$$\text{Area} = \pi r^2$$

Ejemplo de ejecución:



```
Radio = 4.57  
Diametro de la circunferencia = 9.14  
Perimetro de la circunferencia = 28,714  
Area de la circunferencia = 65,612
```

6. Implemente un programa en Java (**CalculoIntereses**) que muestre por pantalla cuánto le abonará su banco después de 6 meses si pone 2500,00€ en una cuenta a plazo fijo al 1,75% anual (el número de meses de un año es constante e igual a 12). Recuerde que, al pagarle los intereses, su banco le retendrá el 21% para hacienda (éste es un interés fijo y puede utilizar también una constante). Utilice variables para manejar las cantidades y realizar los cálculos, todas con un nombre apropiado. Muestre por pantalla los mensajes exactamente como se indica a continuación, para ello recuerde como muestra Java datos por pantalla con la función `printf`.

Ejemplo de ejecución:



```
*** Calculo de intereses ***  
Dinero ingresado: 2500,00€  
Interes anual: 1,75%  
Intereses a los 6 meses: 21,88€  
Retencion realizada: 4,59€  
Intereses cobrados: 17,28€
```

Nota importante: Cuando se hayan estudiado los repositorios, el alumno deberá cargar el trabajo realizado en su repositorio personal mediante su nombre de usuario y su clave. Además, los profesores proporcionarán una clave para comprobar la asistencia a la sesión, que se proporcionará en el comentario que se ha de añadir al hacer el *Commit* al terminar la clase de prácticas.