

**EJERCICIOS T. 4. Programación modular. Los métodos**

**NOTA:** Todos los métodos que se desarrollen para resolver los siguientes ejercicios, habrán de ir acompañados de código en el método *main* que pruebe el funcionamiento correcto de los mismos.

1. Escriba un método que realice operaciones aritméticas. El método recibirá como parámetros los dos operandos, y el operador con el que actuar, y devolverá el resultado de aplicar la operación sobre los operandos.
2. Escriba un método *longitud* que devuelva la longitud de una circunferencia cuyo radio se le pasa como parámetro. Escriba también el método *area* que retorne el área del círculo cuyo radio se le proporciona como parámetro.
3. Escriba un método llamado *maximo*, que reciba como parámetros dos valores enteros positivos, y devuelva el mayor de los dos.
4. Desarrolle un programa que use el método *maximo* que acaba de escribir para determinar el mayor de tres valores solicitados por teclado.
5. Realice un método en Java que reciba tres valores como parámetros, e informe con el valor devuelto de si están o no ordenados de mayor a menor.
6. Desarrolle un método llamado *factorial* que reciba como parámetro un valor entero, y devuelva su factorial. Desarrolle dos versiones de dicho método: uno no recursiva, y otro recursivo.
7. Programe el método *primo*, que determine si un valor que recibe como parámetro es o no primo, devolviendo la conclusión como resultado de dicho método.
8. Ayúdese de una tabla ASCII para crear un programa que, dado un carácter introducido por teclado, muestre si se trata de una letra, un número o un carácter especial. Para ello válgase de un método que reciba dicho carácter, y que devuelva un valor diferente para cada uno de los casos. Emplee constantes para hacer más legible el código.
9. Realice un método que reciba como parámetros un mensaje a mostrar por pantalla, y el número de veces que ha de ser mostrado. El método mostrará dicho mensaje por pantalla tantas veces como se haya indicado.
10. Programe un método que solicite los datos correspondientes a los campos de una asignatura almacenada a partir de una clase, y devuelva los valores proporcionados. Programe también el método que permita mostrar los datos de una asignatura pasada como parámetro a dicho método.

#### T. 4. Programación modular: los métodos

11. Escriba un método *tabla* que reciba como parámetro un número entero, y muestre por pantalla la tabla de multiplicar típica correspondiente a ese valor. Use dicho método para crear un programa que muestre por pantalla la tabla típica con las tablas de los 10 primeros valores.

12. Realice un método *producto* que reciba dos enteros y devuelva su producto, realizado mediante sumas.

13. Desarrolle el método *potencia*, que reciba como parámetros una base y un exponente enteros, y retorne el resultado de elevar la base al exponente, sin hacer uso del método *pow* de la clase *Math* (que es precisamente la que está usted "implementando").

14. Haga uso de métodos para crear una aplicación que permita calcular perímetros de figuras, y áreas comprendidas por las mismas. Se habrá de mostrar un menú principal como el siguiente:

1. Triángulo
2. Cuadrilátero
3. Circunferencia
4. Salir

El programa mostrará dicho menú hasta que se pulse la opción de salir, en cuyo caso el programa finalizará normalmente.

Al pulsarse cualquiera de las otras opciones, se mostrará un submenú como el que sigue:

1. Perímetro
2. Área comprendida
3. Volver

El programa mostrará dicho submenú hasta que se pulse la opción de Volver, en cuyo caso se volverá al menú principal.

Al pulsarse cualquiera de las otras opciones, se solicitarán los datos necesarios, y se mostrará, según sea el caso, el perímetro de la figura, o el área comprendida en su interior.

15. Escriba un método que reciba una letra y devuelva otra. Si la letra era minúscula, la devolverá en mayúscula, y viceversa.

16. Cree un método *cuadro*, que reciba como parámetro la coordenada superior izquierda de un cuadrado ( $X_1$ ,  $Y_1$ ), y la coordenada inferior derecha del mismo ( $X_2$ ,  $Y_2$ ), y muestre por pantalla dicho cuadrado. Considérese la unidad como el espacio que ocupa cada carácter.

17. Escriba un método que reciba los coeficientes de una ecuación de segundo grado, y devuelva las raíces que tiene.

#### T. 4. Programación modular: los métodos

18. Realice un programa para crear el juego de dados *Craps*, en el que lanzas los dados una vez o más hasta que ganas o pierdes. El juego se puede simular en una computadora sustituyendo el lanzamiento real de los dados con la generación de números aleatorios.

Si en la primera tirada se obtiene un 7 ó un 11, se gana directamente.

Si en la primera tirada se obtiene un 2, 3 ó 12, se pierde directamente.

Si en la primera tirada se obtiene 4, 5, 6, 8, 9 ó 10, se sigue tirando. Se ganará si se obtiene esa misma puntuación en alguna tirada de las siguientes, antes de sacar un 7, en cuyo caso se pierde.

Se simulará un lanzamiento de dados cada vez que se pulse la tecla Intro. Se obtendrá entonces un mensaje informando de lo que ha ocurrido en el lanzamiento. Al final de la partida, el programa preguntará si se desea seguir jugando o no, e irá informando acerca del número de partidas jugadas, cuántas de ellas ganadas y cuántas perdidas.

Habrán de usarse, al menos, los métodos *tirada* (para tirar los dados) y *juego* (para jugar una nueva partida)

NOTA: Pregunte al profesor por el uso de métodos de obtención de números aleatorios.

19. Desarrolle un método recursivo que calcule el resultado de elevar una base a un exponente (esto es, un método recursivo para resolver una potencia).

20. El máximo común divisor (M.C.D.) de dos números se puede calcular recursivamente teniendo en cuenta las siguientes premisas:

$\text{mcd}(n,0)=n$

$\text{mcd}(n,m)=\text{mcd}(m,n \bmod m)$

Implemente, pues, el método recursivo que calcule el máximo común divisor de dos números.

21. Programe un método recursivo llamado *fibonacci*, que reciba como argumento un valor  $n$ , y devuelva el  $n$ -ésimo elemento de la serie de Fibonacci. Para resolverlo, tenga en cuenta que en la posición 0 habrá un 0, en la posición 1 habrá un 1, y en el resto de posiciones, la suma de los dos valores anteriores (esto es, empezando en la posición 0, la serie sería 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...)

22. Realice un método que proporcione la longitud de una circunferencia, y el área de un círculo, cuyo radio se le proporciona.

23. Cree un método llamado *ordena*, que reciba tres valores proporcionados por el usuario, y que los devuelva ordenados de mayor a menor. Además, el método devolverá un valor definido por una constante para indicar que ya estaban ordenados, si lo estaban, y otro distinto para indicar que no estaban ordenados, y ha tenido que ordenarlos el propio método.

24. Desarrolle un método en Java que reciba como parámetro un número natural, y que muestre las cifras que componen dicho número. Dicho método, además, deberá devolver la cantidad de cifras que componen el número. Pruebe el funcionamiento del método

#### T. 4. Programación modular: los métodos

invocándolo desde main con diferentes números solicitados al usuario por teclado. Habrá de funcionar con números naturales de hasta 8 cifras, aunque tiene que implementarse de modo que resulte sencillo el cambio a realizar para que funcione con números más grandes.

25. Escriba un programa que obtenga el número primo inferior a un valor introducido por teclado. Se recomienda implementar un método que determine si un número es primo o no, y usarlo para ir probando números desde el valor introducido hacia abajo.

26. Implemente los siguientes métodos llamados suma:

- Uno que reciba dos enteros, y proporcione su suma.
- Otro que reciba tres enteros, y proporcione su suma.
- Otro que reciba tres reales, y proporcione su suma.
- Otro que reciba dos cadenas, y devuelva otra con la concatenación de ambas.
- Otro que solicite por pantalla sucesivos valores enteros hasta que el usuario introduzca un 0. Éste método devolverá la suma de todos ellos.

27. Escriba diferentes programas de prueba que le permitan practicar con los diferentes aspectos de los métodos en Java, tratados a lo largo de esta unidad.