

## **Métodos. Continuación**

17.- Implemente los siguientes métodos enteros:

- a) El método `centígrados` que devuelve la equivalencia en grados centígrados de una temperatura en grados fahrenheit, utilizando el calculo

$$\text{centígrados} = 5.0 / 9.0 * (\text{fahrenheit} - 32);$$

- b) El método `fahrenheit` que devuelve la equivalencia en grados fahrenheit de una temperatura en grados centígrados, utilizando el calculo

$$\text{fahrenheit} = 9.0 / 5.0 * \text{centígrados} + 32;$$

- c) Utilice los métodos de las partes (a) y (b) para escribir una aplicación que permita al usuario, ya sea escribir una temperatura en grados fahrenheit y mostrar su equivalente en grados centígrados, o escribir una temperatura en grados centígrados y mostrar su equivalente en grados fahrenheit.

19.- Se dice que un numero entero es un *número perfecto* si sus factores, incluyendo 1 (pero no el numero entero), al sumarse dan como resultado el numero entero. Por ejemplo, 6 es un numero perfecto ya que  $6 = 1 + 2 + 3$ . Escriba un método llamado `perfecto` que determine si el parámetro numero es un numero perfecto. Use este método en una aplicación que determine y muestre todos los números perfectos entre 1 y 1000. Imprima los factores de cada numero perfecto para confirmar que el numero sea realmente perfecto. Ponga a prueba el poder de su computadora, evaluando números mas grandes que 1000. Muestre los resultados.

22.- Escriba una aplicación que simule el lanzamiento de monedas. Deje que el programa lance una moneda cada vez que el usuario seleccione la opción del menú “Lanzar moneda”. Cuente el numero de veces que aparezca cada uno de los lados de la moneda. Muestre los resultados. El programa debe llamar a un método separado, llamado `tirar`, que no tome argumentos y devuelva `false` en caso de cara, y `true` en caso de cruz. [Nota: si el programa simula en forma realista el lanzamiento de monedas, cada lado de la moneda debe aparecer aproximadamente la mitad del tiempo.]

26.- Escriba una aplicación que juegue a “adivina el numero” de la siguiente manera: su programa elige el numero a adivinar, seleccionando un entero aleatorio en el rango de 1 a 1000. La aplicación muestra el indicador Adivine un número entre 1 y 1000. El jugador escribe su primer intento. Si la respuesta del jugador es incorrecta, su programa debe mostrar el mensaje Demasiado alto. Intente de nuevo. o Demasiado bajo. Intente de nuevo., para ayudar a que el jugador “se acerque” a la respuesta correcta. El programa debe pedir al usuario que escriba su siguiente intento. Cuando el usuario escriba la respuesta correcta, muestre el mensaje Felicidades. Adivino el numero! Y permita que el usuario elija si desea jugar otra vez.