**Desarrollo Avanzado de Aplicaciones I**

**Evaluación Continua 1**

**Ejercicio 1**. Desarrollar una aplicación que solicite el ingreso de un número y muestre un mensaje indicando si es divisible entre 3. Considerar las siguientes instrucciones:

1. La clase java (archivo .java) debe tener como nombre el apellido paterno del alumno y debe contener todos los métodos que se usen.
2. Se debe solicitar que ingrese un número y se debe llamar a un método para validar que es divisible entre 3.
3. El método debe tener el modificador de acceso privado.
4. El método debe recibir como parámetro el número a evaluar y debe devolver el resultado de la evaluación (verdadero o falso).
5. La llamada al método se debe hacer de forma directa, no se debe instanciar la clase.
6. Se debe mostrar un mensaje de la siguiente forma: “El número XXX es divisible entre 3.” o “El número XXX NO es divisible entre 3.” dependiendo del resultado de la evaluación.

**Ejercicio 2**. Desarrollar una aplicación que pida ingresar una temperatura en grados Celsius y que muestre su equivalente en grados Farenheit. Considerar las siguientes instrucciones:

1. La clase java (archivo .java) debe tener como nombre el apellido materno del alumno y debe contener todos los métodos que se usen.
2. Se debe solicitar que ingrese un número con el siguiente mensaje: “Ingrese la temperatura en grados Celsius: “.
3. Se debe utilizar un método para la conversión de grados Celsius a grados Farenheit.
4. El método debe tener el modificador de acceso protegido.
5. El método no debe ser estático. Para llamar al método se debe instanciar la clase.
6. La clase debe tener dos variables globales, una para cada unidad de medida de la temperatura y estas variables deben ser usadas en los métodos.
7. Para la conversión utilizar la siguiente fórmula: Grados Farenheit = (Grados Celsius \* 1.8) + 32.
8. El resultado debe mostrar el siguiente mensaje “Su equivalente en grados Farenheit es: XX”, donde XX se debe de mostrar como un número entero y se debe de redondear por defecto o por exceso según corresponda.

**Ejercicio 3**. Desarrollar una aplicación que calcule el perimétro de una figura geométrica. Considerar las siguientes instrucciones:

1. La clase java (archivo .java) debe tener como nombre las iniciales del alumno.
2. Las figuras geométricas son cuadrado y rectángulo.
3. El cálculo de los perimétros de las figuras geométricas debe realizarse en otra clase llamada “FiguraGeometrica” que debe estar en el mismo archivo.
4. La clase “FiguraGeometrica” debe tener dos métodos sobrecargados llamados “calcularPerimetro”.
5. El perímetro del cuadrado es: 4 \* lado. El perímetro del rectángulo es: 2 \* (base + altura).
6. Se debe solicitar la elección de la figura geométrica con el siguiente mensaje: “Ingrese C para cuadrado o R para rectángulo: " (Tip: Usar el método charAt(0) para leer la letra).
7. Utilizar un switch para ejecutar el código correspondiente a cada figura geométrica.
8. En caso sea cuadrado, se debe solicitar que ingrese el lado del cuadrado y llamar al método correspondiente.
9. En caso sea rectángulo se debe solicitar que ingrese la base y altura del rectángulo y llamar al método correspondiente.
10. En caso ingresen otra letra, se debe mostrar el mensaje: “Opción NO válida.”.
11. Cada método de la clase “FiguraGeometrica” debe tener los parámetros que necesite para el cálculo del perímetro de su figura geométrica.
12. La llamada a los métodos se debe hacer de forma directa, no se debe instanciar la clase.
13. Cada método debe tener un modificador de acceso diferente.
14. Los resultados se deben mostrar desde cada uno de los métodos y deben ser de la siguiente forma “El perímetro del cuadrado es: XX” o “El perímetro del rectángulo es: XX”.

**Ejercicio 4**. Desarrollar una aplicación que solicite ingresar dos cadenas de caracteres y que realice la siguiente validación: ¿Alguna de ellas está contenida en la otra?. Considerar las siguientes instrucciones:

1. La clase java (archivo .java) debe tener como nombre el nombre de pila del alumno.
2. La clase debe tener dos variables globales para las cadenas de caracteres.
3. Se debe solicitar el ingreso de dos cadenas y asignar los valores a las variables globales.
4. La validación debe realizarse en un método de otra clase que debe llamarse “Validaciones” y debe estar en el mismo archivo.
5. El método debe llamarse “contiene” y debe tener como modificador de acceso predeterminado, debe recibir como parámetros dos cadenas para realizar la validación y debe devolver como resultado verdadero o falso.
6. Para llamar al método se debe instanciar la clase y se debe enviar como argumentos los valores de las variables globales.
7. Se debe evaluar si la primera cadena contiene a la segunda cadena, en cuyo caso se debe mostrar el mensaje “La cadena 1 contiene a la cadena 2.” o si la segunda cadena contiene a la primera cadena, en cuyo caso se debe mostrar el mensaje “La cadena 2 contiene a la cadena 1.” y sino se cumple ninguna de las dos situaciones se debe mostrar el mensaje "Las cadenas son totalmente diferentes."