

**Trabajo Práctico Nro. 2**  
*Titular: Lic. Claudio O. Biale*

1) Modifique el ejemplo de la clase *Circulo* para que guarde la cantidad de objetos creados. Imprima la cantidad de objetos creados luego de crear un nuevo objeto.

2) Diseñar una clase llamada *Accion* que contenga:

- Un atributo denominado *simbolo* de tipo cadena de caracteres que especifica el símbolo de la acción.
- Un atributo denominado *nombre* de tipo cadena de caracteres que especifica el nombre de la acción.
- Un atributo denominado *precioCierrePrevio* de tipo double que almacena el precio de la acción en el día anterior.
- Un atributo denominado *precioActual* de tipo double que almacena el precio actual de la acción.
- Un método denominado *getPorcentajeCambio()* que devuelve el porcentaje de cambio entre el precio de la acción en el día anterior con respecto al precio actual.
- Un constructor que crea una acción que recibe como parámetros el símbolo y nombre de la acción, los valores de cierre anterior y actual deben tomar el valor de cero (0).
- Los métodos de acceso y modificación para los atributos.

Dibuje el diagrama UML para la clase y luego implemente la clase. Escriba un programa de prueba que cree una acción con el símbolo ORCL y el nombre Oracle Corporation, establezca que el precio de cierre anterior es de 34.5, establezca un precio actual a 34.35 y muestre el porcentaje de cambio de precio.

3) Diseñar una clase llamada *Cuenta* que contenga:

- Un atributo denominado *id* de tipo int que almacena el nro. de cuenta (predeterminado 0).
- Un atributo denominado *saldo* de tipo double que almacena el saldo de la cuenta (valor predeterminado 0).
- Un atributo denominado *tasaInteresAnual* de tipo double que almacena la tasa de interés anual (valor predeterminado 0). Suponga que todas las cuentas tienen el mismo tipo de interés.
- Un atributo denominado *fechaCreacion* de tipo Date que almacena la fecha en la que se creó la cuenta.
- Un constructor sin argumento que crea una cuenta predeterminada.
- Un constructor que crea una cuenta con un determinado id y saldo inicial.
- Los métodos de acceso y modificación para los atributos *id*, *saldo* y *tasaInteresAnual*.
- El método de acceso para *fechaCreacion*.
- Un método denominado *getTasaInteresMensual* que devuelve la tasa de interés mensual.
- Un método denominado *getInteresMensual* que devuelve el interés mensual.
- Un método denominado *retirar* que retira una cantidad especificada de la cuenta.
- Un método denominado *depositar* que deposita una cantidad especificada en la cuenta.

- Un método denominado *toString* que devuelve una cadena de caracteres que contiene: el id, saldo, interés mensual y fecha en que se creó la cuenta.

Dibuje el diagrama UML para la clase y luego implemente la clase. Escriba un programa de prueba que cree un objeto *Cuenta* con un nro. de cuenta 1122, un saldo de \$ 20.000 y una tasa de interés anual del 4,5%; retire \$ 2.500, deposite \$ 3,000 e imprima: el id, saldo, interés mensual y fecha en que se creó la cuenta.

Sugerencias:

- El método *getInteresMensual()* debe devolver el interés mensual, no la tasa de interés. El interés mensual es igual a *saldo* \* la tasa de interés mensual. La tasa de interés mensual es igual a *tasaInteresAnual* / 12.
- Tenga en cuenta que *tasaInteresAnual* es un porcentaje, por ejemplo: 4,5%. (Debe dividir ese valor por 100).

### Opcionales:

4) Diseño de una clase denominada *Cronometro*. La clase contiene:

- Los datos *horaInicio* y *horaFinal* con sus métodos *get*.
- Un constructor sin argumentos que inicializa la hora de inicio con la hora actual.
- Un método denominado *comienza* que restablece la hora de inicio a la hora actual.
- Un método denominado *detiene* que establece la hora de finalización a la hora actual.
- Un método denominado *tiempoTranscurrido* que devuelve el tiempo transcurrido en milisegundos.

Dibuje el diagrama UML para la clase y luego implemente la clase. Escriba una prueba de programa que mida el tiempo de ejecución de la impresión por pantalla de los números del 1 al 100.000.

5) Diseñe una clase denominada *Ventilador* para representar un ventilador. La clase contiene:

- Tres constantes con nombre *LENTO*, *MEDIO* y *RAPIDO* con los valores 1, 2 y 3 para denotar la velocidad del ventilador.
- Un atributo denominado *velocidad* de tipo entero que especifica la velocidad del ventilador (el valor predeterminado es *LENTO*).
- Un atributo lógico denominado *encendido* que especifica si el ventilador está encendido (el valor predeterminado es falso).
- Un atributo de tipo real denominado *radio* en que especifica el radio del ventilador (el valor predeterminado es 5).
- Un atributo de tipo texto denominado *color* que especifica el color del ventilador (el valor por defecto es azul).
- Los métodos de acceso y modificación para los cuatro atributos.
- Un constructor sin argumentos que crea un ventilador por defecto.
- Un método denominado *toString* que devuelve una cadena que describe al ventilador. Si el ventilador está encendido, el método devuelve la velocidad del ventilador, el color, y el radio. Si el ventilador no está encendido, el método

devuelve el color del ventilador y el radio junto con la cadena *"el ventilador está apagado"*.

Dibuje el diagrama UML para la clase y luego implemente la clase. Escriba un programa de prueba que cree dos ventiladores. Asigne una velocidad máxima, un radio de 10, un color amarillo y un estado de encendido al primer ventilador. Asigne una velocidad media, un radio de 5, un color azul y un estado apagado al segundo ventilador. Muestre los objetos invocando al método *toString*.