

1. Identificar las principales propiedades (atributos y métodos) de los siguientes objetos:

- a) Triángulo.
- b) Reloj.
- c) Alumno.
- d) Ciudad.
- e) Libro.
- f) Canción.
- g) Televisor.
- h) Mueble.
- i) Película.

Ejemplo:

Rectángulo.

Clase: Rectangulo.

Atributos: base, altura.

Métodos: calcular la superficie, calcular el perímetro, cambiar la base, cambiar la altura, dibujar, pintar, etc.

2. Según la clase Empleado que modela a los empleados de una empresa, se cuenta por cada uno de ellos, las diferentes unidades de información: Número de DNI, Nombre y apellidos, Edad, Sueldo base, Pago por hora extra, Hora de inicio y de finalización de trabajo para cada día del mes, Tipo (porcentaje) de IRPF, Si está casado o no y el Número de hijos. Los servicios que proporciona la clase, teniendo en cuenta que será utilizada en un programa de gestión de información de la plantilla de la empresa, poseen las siguientes funcionalidades:

- a) Contratar un nuevo empleado.
  - b) Despedir a un empleado.
  - c) Actualizar la información de un empleado dado.
  - d) Obtención de diferentes informaciones acerca del sueldo de un empleado escogido: Sueldo base, Sueldo bruto (incluye el base y el complemento correspondiente a las horas extra), Complemento correspondiente a las horas extra realizadas por el empleado y Cálculo de las retenciones (IRPF) a partir del tipo del empleado. Para ello hay que tener en cuenta que el porcentaje de retención que hay que aplicar es el tipo menos 2 puntos si el empleado está casado y menos 1 punto por cada hijo que tenga; el porcentaje se aplica sobre todo el sueldo bruto.
  - a) Visualización de los datos personales de un empleado elegido.
  - b) Visualización de toda la información del empleado elegido: los datos personales más el sueldo base, el complemento por horas extra, el sueldo bruto, la retención de IRPF y el sueldo neto.
- ¿Cambiarías algo en la resolución del ejercicio si todos los empleados cobrasen lo mismo por cada hora extra realizada?

3. Resolver los siguientes ejercicios respecto al ámbito de los identificadores:

- a) Escribir en las líneas punteadas a qué hace referencia.

```
class ClaseAmbito
{
    entero ident1
    caracter ident2
    booleano ident3
}
```

```
ClaseAmbito()  
{  
    ident1... //.....  
    ident2... //.....  
    ident3... //.....  
}
```

```
metodoUno()  
{  
    booleano ident3  
    ident1... //.....  
    ident3... //.....  
    retornar  
}
```

```
metodoDos(entero ident1, caracter ident2)  
{  
    ident1... //.....  
    ident2... //.....  
    retornar  
}
```

```
metodoTres(booleano ident3)  
{  
    caracter ident2  
    ident1... //.....  
    ident2... //.....  
    ident3... //.....  
    retornar  
}
```

```
metodoCuatro()  
{  
    entero ident1  
    caracter ident2  
    booleano ident3  
    ident1... //.....  
    ident2... //.....  
    ident3... //.....  
    retornar  
}  
}
```

b) Teniendo en cuenta la clase ClaseAmbito, escribir en las líneas punteadas el resultado de los mensajes, según la ejecución del programa AmbitoIdentificador.

```
class ClaseAmbito
{
    entero id1
    entero id2
    entero id3

    ClaseAmbito()
    {
        id1 ← 0
        id2 ← 1
        id3 ← -1
    }

    metodoUno(entero id1)
    {
        id1 ← id2 + 5
        escribir id1
        retornar
    }

    metodoDos()
    {
        entero id2
        id2 ← 33
        id1 ← id3 + id2
        escribir id1, id2, id3
        retornar
    }

    metodoTres()
    {
        escribir id1, id2, id3
        retornar
    }

    metodoCuatro(entero id2, entero id3)
    {
        id1 ← id1 + 1
        id2 ← id2 + 1
        id3 ← id3 + 1
        escribir id1, id2, id3
        retornar
    }
}
```

```
{  
principal AmbitoIdentificador  
  {  
    ClaseAmbito objetoAmbito  
    objetoAmbito = nuevo ClaseAmbito()  
    objetoAmbito.metodoTres() .....  
    objetoAmbito.metodoCuatro(1, -1) .....  
    objetoAmbito.metodoDos() .....  
    objetoAmbito.metodoTres() .....  
    objetoAmbito.metodoUno(5) .....  
    objetoAmbito.metodoTres() .....  
    objetoAmbito.metodoCuatro(1, 0) .....  
    objetoAmbito.metodoDos() .....  
    objetoAmbito.metodoTres() .....  
  }  
}
```

**Aclaración: En este práctico se incluyen pocos ejercicios donde se solicite hacer el programa (PRINCIPAL), ya que se le da prioridad a la creación de las clases. Pero, sería recomendable que por cada clase, prueben crear y manipular objetos a través de los mensajes. No olvidar los testeos o validaciones correspondientes.**

4. En base a las propiedades obtenidas en el primer ejercicio, implementar la clase Televisor. Con el constructor, los *set* y *get* que sean necesarios, y con los métodos para encender, apagar, subir y bajar el canal, subir y bajar el volumen, y volver al canal antes seleccionado. Nota: no olvidar que el televisor debe estar encendido para realizar ciertas todas las tareas.

5. Una hora se compone de 3 elementos: hora, minutos y segundos. Codificar la clase Hora con los siguientes métodos:

- Cuando se crea el objeto de la clase Hora, los valores de hora, minutos y segundos deben ser válidos (es decir, una hora correcta en Argentina). Si alguno es inválido, generar la hora 00:00:00.
- Permitir cambiar la hora, los minutos y los segundos por separado.
- Devolver la hora en minutos.
- Devolver la hora en segundos.
- Cambiar la hora aumentándola en X minutos.
- Mostrar la hora de la siguiente forma: hh:mm:ss.

6. Una fecha se compone de 3 elementos: día, mes y año. Codificar la clase Fecha con los siguientes métodos:

- Cuando se crea el objeto de la clase Fecha, los valores de día, mes y año deben ser válidos (es decir, una fecha correcta). Si alguno es inválido, generar la fecha 01/01/2002.
- Devolver el día; devolver el mes; devolver el año.
- Cambiar el día; cambiar el mes; cambiar el año.
- Obtener el nombre del mes de la fecha (ejemplo: si el mes es 1, obtener Enero).
- Mostrar a qué estación del año pertenece.
- Mostrar la fecha de la siguiente forma: 12 de Octubre de 2002.

g) Calcular la cantidad de días que hay entre una fecha cargada y la fecha creada, del mismo año. Suponer que la fecha cargada es menor que la fecha creada.

7. Escribir un programa en el que se creen dos objetos de tipo Fecha. Probar los métodos más importantes creados anteriormente. Mostrar en formato (dd/mm/aaaa) aquella fecha, de ambas, que se encuentre primero en el calendario.

8. En un billetera pueden guardarse billetes de 2, 5, 10, 20, 50 y 100 pesos. Escribir la clase Billetera que permita guardar y quitar billetes (de a un billete o a partir de un monto determinado). Ver el contenido total de la billetera.

9. Completar la implementación de la clase CajaDeAhorro, que representa a una caja de ahorros en pesos, bancaria.

```
clase CajaDeAhorro
{
    entero numero
    real saldo
    caracter [] titular
    caracter [] domicilio
    entero telefono

    CajaDeAhorro(...)
    {
        .
        .
        .
    }

    depositarDinero(entero CANTIDAD)
    {
        .
        .
        .
    }

    extraerDinero(entero CANTIDAD)
    {
        .
        .
        .
    }

    consultarSaldo()
    {
        .
        .
        .
    }
}
```

Tener en cuenta que:

- a) los depósitos siempre son cantidades positivas mayores que 0.
- b) nunca se puede hacer una extracción mayor al saldo.
- c) dado que el cliente puede mudarse, debe ser posible la modificación del domicilio y teléfono.
- d) debe ser posible la consulta de todos los datos de la caja.
- e) si cree conveniente agregar algún otro método, incorpórelo.

**10.** Crear una clase Cadena, que represente una palabra. Alguno de los métodos podrá ser: buscar un caracter en la cadena, reemplazar un caracter por otro, devolver el primer carácter, contar la cantidad de ocurrencias de un determinado caracter, etc.

**11.** Agregar a la clase Vector métodos para:

- a) Devolver el elemento de una determinada posición.
- b) Devolver la cantidad de elementos.
- c) Calcular la suma de los elementos.
- d) Calcular el promedio de los elementos.
- e) Modificar el elemento de una determinada posición.
- f) Buscar y obtener el elemento más grande.
- g) Buscar y obtener el elemento más chico.
- h) Ordenar los elementos en otro vector.
- i) Devolver en otro vector los elementos invertidos.

**12.** Con la clase Vector resultante, hacer un programa que use dos vectores de clase Vector de diferentes dimensiones y muestre el promedio más alto, muestre el mayor elemento del primer vector y menor elemento del segundo vector y devuelva un vector ordenado y el otro invertido.

**13.** Crear una clase Matriz, que represente una matriz de  $M \times N$  elementos enteros. Escribir métodos para:

- a) Agregar elementos.
- b) Devolver el elemento de una determinada posición (fila, columna).
- c) Mostrar la matriz.
- d) Calcular la traspuesta (generar otra matriz). Ejercicio 13 del Práctico 6.
- e) Mostrar los elementos de la diagonal.
- f) Mostrar los elementos de una determinada fila.
- g) Mostrar los elementos de una determinada columna.
- h) Calcular la sumatoria de los elementos del contorno. Una opción del Ejercicio 9 del Práctico 6.
- i) Verificar si la matriz es la matriz identidad.
- j) Modificar el elemento de una determinada posición (fila, columna).

**14.** Crear la clase MatrizRala cuyos métodos serán: generar una matriz; verificar si conviene o no reducirla; si se justifica, efectuar la transformación generando la matriz reducida; mostrar la matriz original a partir de la matriz reducida. Agregarle a la clase los siguientes métodos: determinar si existe al menos una fila con todos elementos 0 (decir cuál es) y determinar si existe al menos una columna con todos elementos 0 (decir cuál es). Si cree que falta algún método, agréguelo.

**15.** Suponga que no existe el tipo dato entero para ser usado desde un Principal. Codifique una clase Entero que tenga el valor y los siguientes métodos:

- a) Asignar el valor.
- b) Devolver el valor.
- c) Es par.
- d) Es múltiplo de X.
- e) Es divisible por X.
- f) Operación de suma.
- g) Operación de resta.
- h) Devolver el valor elevado a X. Si X es un valor negativo, devolver 0.
- i) Devolver el factorial. Ejercicio 8 inciso r del Práctico 4.
- j) Es primo. Ejercicio 2 del Práctico 7.
- k) Es un número perfecto. Ejercicio 5 del Práctico 7.
- l) Devolver en número binario. Ejercicio 7 del Práctico 7.