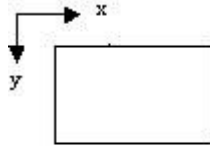


## Boletín 3- Introducción a las clases y objetos

- **Ejercicio 3.1.** Crear una clase **Rect** que represente un rectángulo. La clase tendrá como datos privados de tipo entero, las coordenadas de los vértices superior izquierdo ( $x_1, y_1$ ) y las del vértice inferior derecho ( $x_2, y_2$ )



La clase tendrá los siguientes constructores:

- i. Recibe como parámetros las coordenadas de los vértices ( $x_1, y_1, x_2, y_2$ ).
- ii. Recibe el ancho y el alto ( $w, h$ ) . Inicializa  $x_1$  e  $y_1$  a cero.
- iii. Constructor por defecto.

Tendrá los siguientes métodos:

- i. **mover**: Mueve el rectángulo  $\Delta X$  y  $\Delta Y$
- ii. **estaAdentro**: Recibe un punto ( $x, y$ ) y devuelve verdadero si el punto está dentro del rectángulo.
- iii. **union** Recibe un rectángulo  $r$  y devuelve el menor rectángulo que incluye a  $r$  y a este rectángulo. (este rectángulo es la instancia en la que se invoca el método).
- iv. **interseccion** Recibe un rectángulo  $r$  y devuelve la intersección de  $r$  y este rectángulo.
- v.  **$x_1, y_1, x_2, y_2, w, h$** : Métodos de acceso a los datos. Permiten leer un dato o setear un dato.

- 
- **Ejercicio 3.2** - Crear una clase **TesteaRect** que implementa una aplicación para testear la clase **Rect**. (**TesteaRect** tiene **main** como único método).

- i. Crear 2 objetos de la clase **Rect** :  $r_1$  y  $r_2$

- ii. Obtener los rectángulos u e i unión e intersección de r1 y r2.
  - iii. Imprimir los resultados en la consola.
- 

- **Ejercicio 3.3** - Se decide cambiar la estructura interna de Rect. En vez de guardar como datos las coordenadas de los vértices, se guardan ahora las coordenadas del vértice superior derecho, el ancho y la altura.

- i. Rescribir los métodos dada la nueva estructura.
  - ii. ¿Cómo cambia TesteaRect?
  - iii. Conclusiones.
- 

- **Ejercicio 3.4** –

Una Empresa desea mantener la información de sus proveedores por lo que nos solicita la realización de un programa.

Se nos pide:

- Cada proveedor debe quedar identificado por un número único de 5 cifras (desde el 10.000 en adelante - la cantidad de proveedores es mucho menor que 10.000)
- Se desea mantener la siguiente información:

Nombre del proveedor  
Dirección  
Teléfono

- Debe se posible saber el monto total de lo que se le ha comprado a cada proveedor desde el inicio de la relación comercial con el mismo.

Se pide:

1. Crear una clase proveedor que permita cumplir con lo anterior.

- a) Al momento de ingresar un nuevo proveedor se conoce en algunos casos su nombre y en otros se tiene además los datos de dirección y teléfono.
- b) El número de identificación y el nombre no deben poder modificarse una vez creado un nuevo proveedor.

2. El siguiente código pertenece a otra clase que utiliza la clase Proveedor.

```
...
Proveedor p = new Proveedor("La Divina Tuerca", "18 de
julio 1254", "9059322");
System.out.print(p);
...
```

y provoca la siguiente salida a pantalla:

```
Proveedor N° 13425 - La Divina Tuerca
```

```
18 de julio 12354
```

```
Tel. 9059322
```

- a) Agregue en Proveedor el método que falta para que esto sea posible
- b) Explique que pasa si no agregamos este método.

- 
- **Ejercicio 3.5** - Crear una clase StackDeEnteros que implemente un Stack de números enteros. Para la implementación se utilizará un array de enteros de largo fijo (largo 100).

La clase tendrá los siguientes métodos:

- i. **push** Introduce un entero en el stack
  - ii. **pop** Extrae un entero del stack y lo devuelve
  - iii. **peek** Devuelve el entero en el tope del stack sin extraerlo.
  - iv. **estaLleno** Devuelve verdadero si la capacidad del stack esta colmada.
-