CESUR
Tu Centro Oficial de FP

CASO PRÁCTICO 8

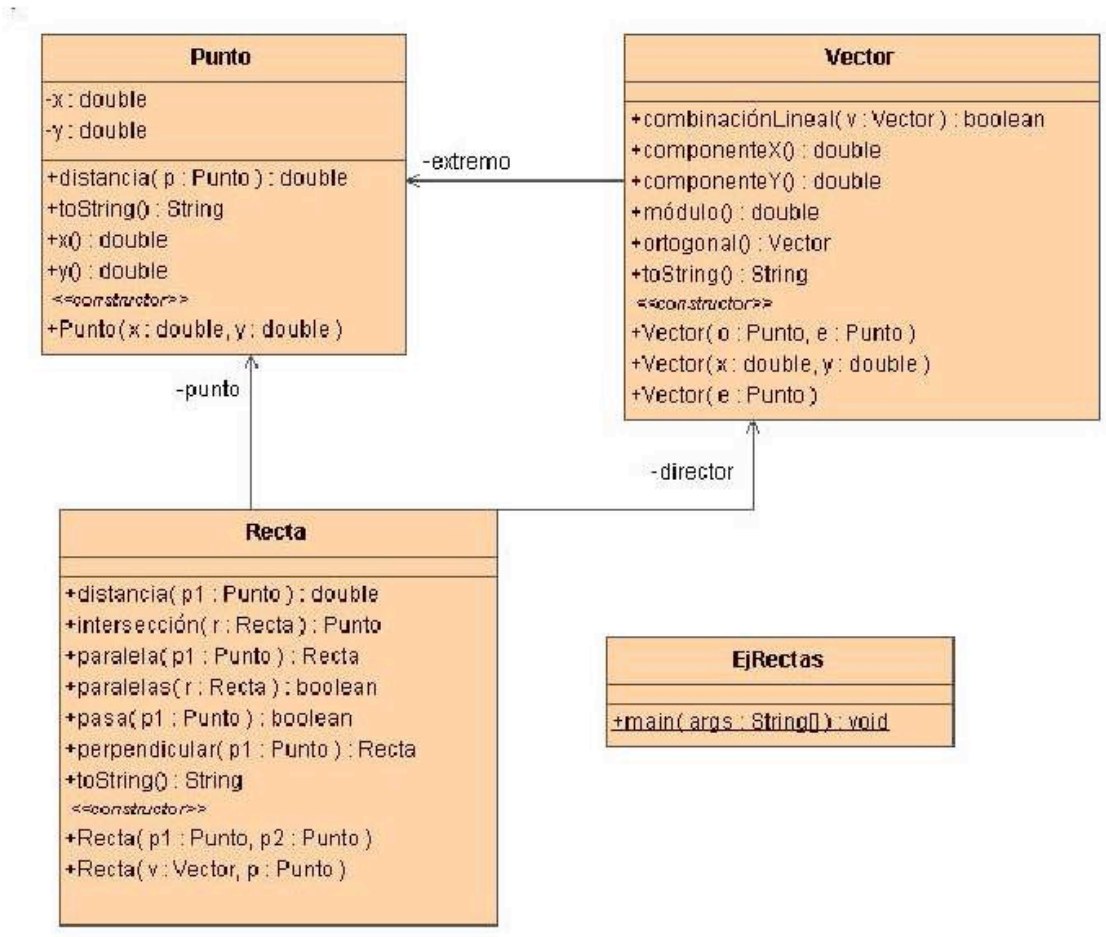
- **TÍTULO: Programación orientada a objetos con Java**

- **SITUACIÓN**

  Tenemos que resolver los siguientes problemas para la empresa de programación para la que trabajamos.

- **INSTRUCCIONES**

En esta práctica implementaremos clases que manipulan puntos, vectores y rectas del plano (clases Punto, Vector y Recta).



Hay que tener en cuenta que, en el diagrama de clases en las flechas de relación de herencia entre clases aparecen unas etiquetas. Estas etiquetas hacen referencia a "atributos" de la clase hija que **no vamos a declarar**.

Por ejemplo, en la relación entre Vector y Punto aparece la etiqueta "-extremo", esto nos indica que extremo es un atributo de la clase Vector que es private (signo -) y su tipo de datos es Punto, pero que no vamos a declarar.

Por otra parte, para elevar un número a otro en Java utilizaremos: **`Math.pow(double n, double e);`** , que devolverá un double.

Para calcular la raíz cuadrada utilizaremos **`Math.sqrt(n);`**

a) Un punto viene determinado por una ordenada y una abscisa (de tipo double). Su representación será P(x,y).

b) La distancia de un punto a otro se calcula de la siguiente forma:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

c) Suponemos que un vector siempre tiene el origen en el origen de coordenadas, por lo que conociendo su extremo un vector queda perfectamente delimitado. La clase Vector tendrá tres constructores: el primero creará un vector conocido su ordenada y su abscisa; el segundo lo hará conociendo el punto extremo; y el último lo creará conociendo un punto origen y un punto extremo (en este caso se realizarán los cálculos para almacenar simplemente el extremo cuando el origen es el origen de coordenadas). Un vector se representará por V(x,y). El módulo de un vector es la distancia del punto (0,0) al extremo del vector.

*Un vector ortogonal a V(x,y) es el vector V(-y,x). Dos vectores V(vx,vy) y V(ux,uy) forman una combinación lineal si verifican vx*uy == vy*ux. Esto quiere decir que los vectores tienen la misma dirección (aunque pueden tener diferente sentido).*

d) Una recta queda determinada por un vector que marca su dirección (vector director) y un punto por donde pasa. Se proporcionarán dos constructores: el primero que genere la recta conociendo un punto por donde pasa y un vector que determina su dirección; y el segundo que genere la recta conociendo dos puntos por donde pasa. Una recta se representa por R(vector,punto).

Dos rectas son paralelas si sus vectores directores forman una combinación lineal. Un punto p pertenece a una recta si el vector formado por p y un punto de la recta junto con el vector director de la recta forman una combinación lineal.

Las rectas R(V(vx,vy),P(px,py)) y R(V(ux,uy),P(qx,qy)) se cortan en el punto de ordenada y abscisa determinados de la siguiente manera:

d = uy * vx – vy * ux

d1 = vx * py – vy * px

d2 = ux * qy – uy * qx

abscisa = (d1 * ux – vx * d2) / d

ordenada = - (vy * d2 - uy * d1) / d

Como vemos, sólo está definido si d no es cero.

Una recta perpendicular a nuestra recta se obtiene pasándole un punto de nuestra recta y su vector director será el vector ortogonal a nuestro vector director de nuestra recta.

e) El siguiente programa calcula el área de un triángulo conociendo los tres puntos del plano que lo delimitan.

```
public class EjRectas {
	public static void main(String[] args) {
		// Calcular el área del triángulo
		// definido por sus tres vértices
		Punto p1 = new Punto(0,0);
		Punto p2 = new Punto(4,0);
		Punto p3 = new Punto(2,3);
		// Se calcula la distancia entre p1 y p2
		double base = p1.distancia(p2);
		// Se calcula la recta que pasa por p1 y p2
		Recta r1 = new Recta(p1,p2);
		// se calcula la distancia entre p3 y r1
		double altura = r1.distancia(p3);
		// El area es base*altura/2
		double area = base*altura/2;
		System.out.println("Puntos: "+p1+","+p2+","+p3);
		System.out.println("Área = "+area);
	}
}
```