Informática
Año 2014/2015
Grados en Matemáticas
Facultad de CC. Matemáticas

Práctica 4

Figuras geométricas en el plano

8 de mayo de 2015

En este ejercicio se realizarán tareas básicas para manejar figuras geométricas sencillas en el plano.

1 Clases para las figuras

Define clases para poder representar puntos (`Point`), rectas (`Line`), segmentos (`Segment`), círculos (`Circle`), triángulos (`Triangle`) y paralelogramos (`Parallelogram`) en el plano.

Todas estas clases deben tener dos métodos `intersects(self,other)` y `distance(self,other)`. siendo `other` una figura de cualquiera de las clases anteriores.

- `intersects(self,other)` debe indicar si `self` y `other` se cortan.
- `distance(self,other)` debe calcular la distancia entre `self` y `other`.

Te será muy útil definir una clase auxiliar `Vector` para facilitar los cálculos. Cada una de las clases debe tener constructores con los siguientes parámetros:

- `Vector`: dos coordenadas.
- `Point`: dos coordenadas.
- `Line`: dos puntos.
- `Segment`: dos puntos.
- `Circle`: un punto y el radio.
- `Triangle`: tres puntos.
- `Parallelogram`: un punto y dos vectores

Además cada clase debe definir el método `__str__`. De esta forma la sentencia de Python `print figura` debe escribir, dependiendo del tipo del objeto `figura`, una de las siguientes cadenas:

- `Point(x,y)`
- `Line(Point(x,y),Point(x,y))`
- `Segment(Point(x,y),Point(x,y))`
- `Circle(Point(x,y),r)`
- `Triangle(Point(x,y),Point(x,y),Point(x,y))`
- `Parallelogram(Point(x,y),Vector(x,y),Vector(x,y))`

donde debemos sustituir `x,y,r` por las cadenas de caracteres correspondientes a los valores concretos de los parámetros pasados al construir la figura.

Las clases también definirán el método `__repr__` (en nuestro caso tiene sentido que devuelva lo mismo que `__str__`). El objetivo de éste método es que tras evaluar código del estilo:

```
p = Point(1,2)
q = eval(repr(p))
```

`p` y `q` apunten a objetos idénticos.

2 Calculos exactos

Implementa los métodos solicitados de manera que, si las coordenadas con las que creamos los objetos de las clases anteriores son enteras o de alguno de los tipos del módulo sympy http://docs.sympy.org/latest/tutorial/intro.html, los resultados se calculen simbólicamente. Por ejemplo:

- `Point(0,0).distance(Point(1,1))` debe devolver `sqrt(2)`
- `Point(0.,0.).distance(Point(1.,1.))` debe devolver `1.41421356237310`

Fecha límite de entrega: 22 de Mayo.