UNIVERSIDAD DE GRANADA

Departamento de Ciencias de la Computación e Inteligencia Artificial

DECSAI

FUNDAMENTOS DE PROGRAMACIÓN
Grado en Ingeniería Informática y Matemáticas

(Diciembre de 2020)

Actualización del ejercicio 23
de la Relación de Problemas IV
(array de circunferencias)

En el ejercicio mencionado puede leer:

> *Escribir un programa que lea las coordenadas que definen un rectángulo y guarde en una array una serie de datos de tipo* `Circunferencia`*, todas centrada en en el punto de corte de las diagonales del rectángulo.*
>
> *Las circunferencias en las que estamos interesadas serán todas las circunferencias inscritas en el rectángulo. Para ello comience con una circunferencia de radio radio=0.5 y vaya incrementando su valor 0.25 en cada iteración.*
>
> *Muestre cuántas circunferencias se han generado y a continuación, sus propiedades (centro y radio).*
>
> *Por ejemplo, dado el rectángulo de la figura 1, el array de datos de tipo* `Circunferencia` *contendrá tres casillas con datos válidos (la circunferencia con trazo discontinuo no se guardará porque no está inscrita en el rectángulo).*

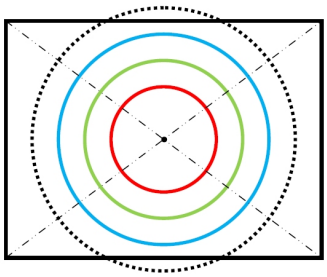

Figure 1: *Cuatro circunferencias concéntricas centradas en el punto de corte de las diagonales del rectángulo. Tres de ellas (trazo contínuo) están inscritas; la cuarta (trazo discontínuo) no lo está.*

En este momento puede modelar las entidades que intervienen en el problema (punto, circunferencia y rectángulo) con **clases** (`Punto2D`, `Circunferencia` y `Rectangulo`, respectivamente). Las circunferencias que formarán parte de la solución se guardarán en una *secuencia*, para lo que tendrá que escribir la clase `SecuenciaCircunferencias`. Las clases mencionadas se proporcionan en el fichero `prueba.cpp` y su estructura se muestra en la tabla 1.

1. Las clases `Punto2D`, `Circunferencia` y `Rectangulo` suministradas están prácticamente finalizadas, aunque debe **completar algunos métodos** (indicado explícitamente en `prueba.cpp`) y es posible que deba **añadir nuevos métodos**.
2. Los métodos de la clase `SecuenciaCircunferencias` debe escribirlos **todos**. Use como referencia una clase `Secuencia`.
3. Deberá completar también parte de la función `main`.

En la figura 2 puede ver un ejemplo de ejecución de este programa.

Punto2D
- double x - double y
+ Punto2D (double abscisaPunto, double ordenadaPunto) + SetCoordenadas (double abscisaPunto, double ordenadaPunto) + double GetX (void) + double GetY (void) + bool EsIgual (Punto2D otro) + double DistanciaEuclidea (Punto2D otro) + string ToString (void)

Circunferencia
- static const double PI - Punto2D centro - double radio
+ Circunferencia (Punto2D el_centro, double el_radio) + Punto2D GetCentro (void) + double GetRadio (void) + double GetDiametro (void) + double Longitud (void) + double Area (void) + string ToString (void)

Rectangulo
- Punto2D esquina_si - double long_ladoX - double long_ladoY
+ Rectangulo (Punto2D la_esquina_si, double la_long_ladoX, double la_long_ladoY) + Punto2D GetEsquinaSI (void) + Punto2D GetEsquinaSD (void) + Punto2D GetEsquinaII (void) + Punto2D GetEsquinaID (void) + double Area (void) + double Perimetro (void) + string ToString (void)

SecuenciaCircunferencias
- static const int TAMANIO - Circunferencia vector_privado[TAMANIO] - int total_utilizados
+ ColeccionCircunferencias (void) ..... + int NumCircunferencias (void) ..... + string ToString (void)

Table 1: Clases que colaboran para la solución del problema

Descargue el fichero `prueba.cpp` y complételo. Deberá entregar en PRADO únicamente el fichero `prueba.cpp`

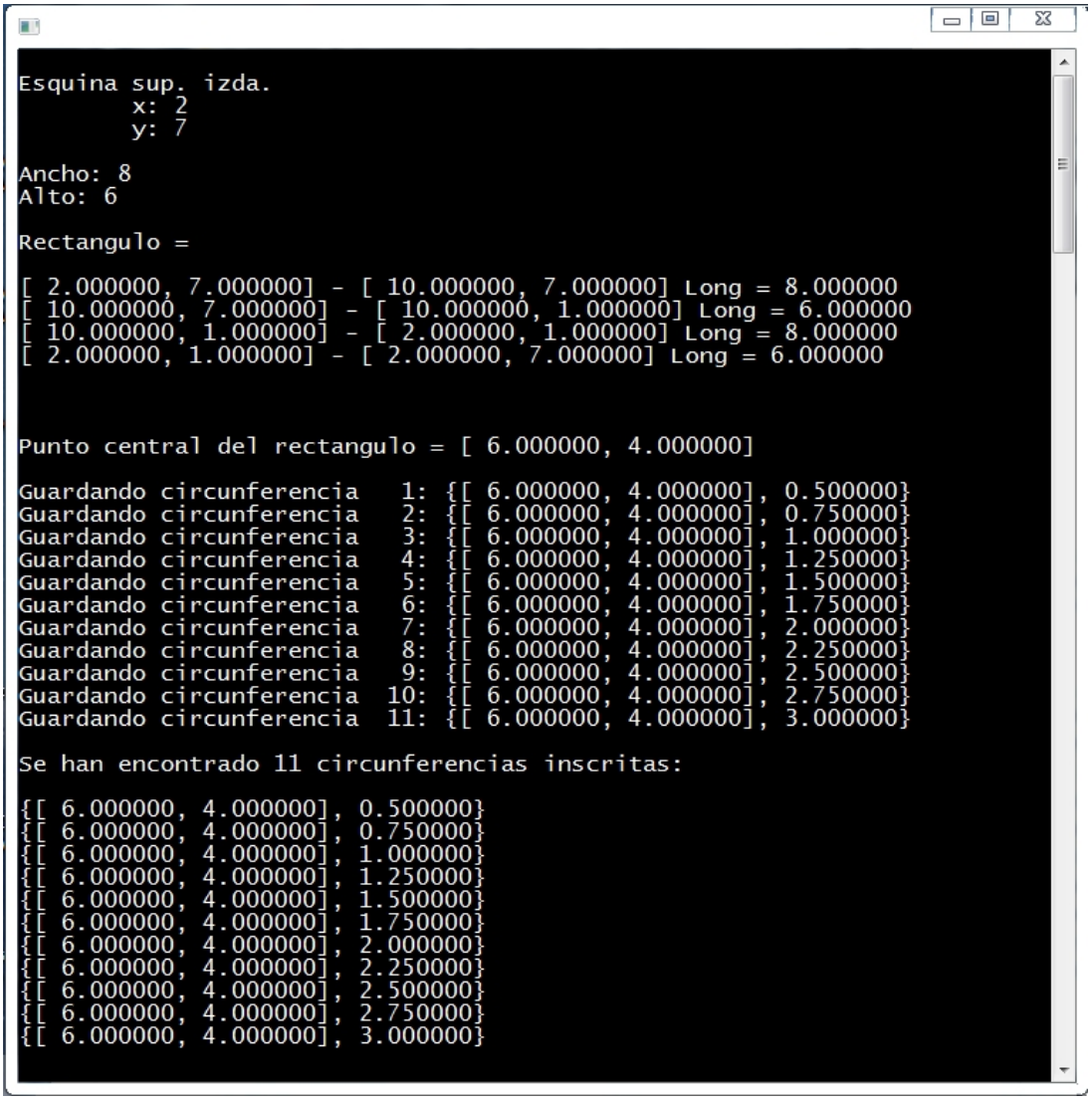

Figure 2: *Ejemplo de ejecución. Circunferencias inscritas en un rectángulo de dimensiones 8 (ancho) × 6 (alto) cuya esquina superior izquierda es* $(2, 7)$.