Fundamentos de Programación

Convocatoria de Febrero. Curso 2015/2016

9 de Febrero 2016

Tiempo: 2h 30m

1. (**2 puntos (sobre 7)**)
Defina una clase para trabajar con una secuencia de caracteres y proporcione métodos para obtener el carácter que hay en un índice determinado, obtener la longitud actual de la secuencia y añadir un carácter nuevo.

   Dada una secuencia de caracteres que contiene un mensaje cifrado, se pide definir un método para construir otra secuencia nueva con el mensaje descifrado. La forma de descifrado consiste en coger la primera y última letra de cada palabra. Las palabras están separadas por uno o más espacios en blanco o el final de la secuencia. Si la última palabra no tiene espacios en blanco a su derecha, se coge sólo el primer carácter.

   Por ejemplo, si denotamos el inicio y final de la secuencia con un corchete, entonces:

   `[  Hidrógeno limpia  ]` se descodificaría como `Hola`

   `[Hidrógeno limpia   ]` se descodificaría como `Hola`

   `[Hidrógeno limpia]` se descodificaría como `Hol`

   `[Hidrógeno]` se descodificaría como `H`

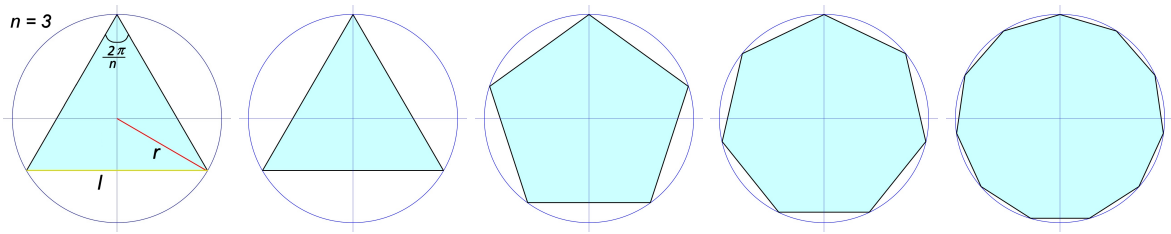
   `[Hidrógeno   ]` se descodificaría como `Ho`

   `[H]` se descodificaría como `H`

   `[H   ]` se descodificaría como `H`

   No hace falta incluir el programa principal.

2. (**3 puntos (sobre 7)**)
Un polígono regular de n caras tiene n lados de la misma longitud y todos los ángulos interiores son iguales. Su centro geométrico es el centro de la circunferencia circunscrita (la que lo envuelve). Supondremos que dos polígonos son distintos si se diferencian o bien en sus centros geométricos, o bien en el número de lados o bien en la longitud de cualquiera de ellos. Así pues, por ejemplo, no tendremos en cuenta las distintas posiciones en el plano que se podrían obtener girando el polígono sobre su centro.



   Si llamamos n al número de lados y l la longitud de cualquiera de ellos, tenemos que:

   - la longitud r del radio de la circunferencia circunscrita viene definida por $r = l/(2\sin(\pi/n))$
   - El área del polígono es $A = (n/2) \cdot r^2 \sin(2\pi/n)$
   - Si queremos construir un polígono inscrito en la misma circunferencia, pero multiplicando por un entero k el número de lados, la longitud de cada uno de los kn lados viene dada por $l' = r\sqrt{2(1-\cos(2\pi/(kn)))}$.

   Se quiere diseñar la clase `PoligonoRegular` para poder representar este tipo de polígonos y realizar las siguientes tareas:

   - Calcular el perímetro del polígono.
   - Calcular el área del polígono.
   - Calcular la diferencia entre el área del polígono y la del círculo correspondiente a la circunferencia circunscrita.
   - Comprobar si un polígono es mayor que otro (considerando al área de cada uno)
   - Construir un nuevo polígono que tenga la misma circunferencia circunscrita y con un número de lados que sea múltiplo del número de lados del polígono.

   Debe tener los siguientes constructores:

   - Un constructor sin parámetros en el que los valores a asignar por defecto sean: 3 para el número de lados (triángulo), 1 para la longitud y (0,0) para las coordenadas del centro.

- Un constructor que cree un polígono regular con una longitud y número de lados concretos y deje como centro el valor por defecto (0,0).
- Un constructor que cree un polígono regular con una longitud, número de lados y centro concretos.

Defina un programa que realice lo siguiente:

- Cree dos polígonos, `polígono1` con los valores por defecto y `polígono2` con 6 lados de longitud 4 y centrado en (0,0). El programa comprobará si `polígono1` es mayor que `polígono2`.
- Construya un nuevo polígono a partir de `polígono1`, con la misma circunferencia circunscrita y con el doble número de lados. El programa imprimirá en pantalla el área del nuevo polígono.
- Repita el proceso anterior generando polígonos con el doble número de lados en cada iteración, hasta que el polígono generado tenga un área *similar* a la del círculo delimitado por la circunferencia circunscrita. Consideraremos que las áreas son similares si no se diferencian en más de 10^{-5}.

  El programa mostrará el número de lados del polígono que aproxima a la circunferencia.

3. (**2 puntos (sobre 7)**)
Dada la especificación de `SecuenciaEnteros`, se quiere implementar un método llamado `SeleccionDireccional` dentro de la clase `TablaRectangularEnteros` que devuelva una secuencia de enteros que contenga una copia de los elementos de la matriz dada una posición de origen y una dirección. El método recibirá una posición origen (i, j) y un dirección dada por dos enteros (dx, dy). La secuencia a devolver contendrá como primer elemento el valor (i, j) de la matriz. Los siguientes elementos se obtendrán dando saltos (dx, dy) desde la posición origen hasta alcanzar el extremo de la matriz. Hay que realizar todas las comprobaciones necesarias antes de acceder a posiciones inválidas en la matriz. Por ejemplo, dada la matriz:

$$\begin{pmatrix} 5 & 8 & 2 & 9 & 3 \\ 1 & 0 & -2 & 9 & 2 \\ 0 & 1 & 8 & \text{-3} & 0 \\ 2 & -2 & 4 & 8 & -5 \\ -2 & 1 & 4 & 3 & 3 \end{pmatrix}$$

la selección direccional con $(i = 0, j = 0)$ y $(dx = 1, dy = 2)$ sería {5,1,4}, y con $(i = 1, j = 3)$ y $(dx = 0, dy = 1)$ sería {9,-3,8,3}.

Con estas indicaciones, y teniendo en cuenta que se va a emplear la clase `TablaRectangularEnteros`, se trata de realizar las siguientes tareas:

a) Defina los datos miembros de la clase `TablaRectangularEnteros`.

b) Implemente los métodos necesarios de `TablaRectangularEnteros`.

c) Implemente el método `SeleccionDireccional` de acuerdo a las indicaciones dadas.

d) Escriba el código de la función `main` que se encarga de llamar adecuadamente al método `SeleccionDireccional`, para obtener una `SecuenciaEnteros` a partir de una `TablaRectangularEnteros` (supondremos que la tabla desde la que se construye la secuencia se ha creado y llenado correctamente).

Métodos de `SecuenciaEnteros` que NO hay que implementar y se pueden usar
`TotalUtilizados`
`Capacidad`
`Aniade`
`Elemento`