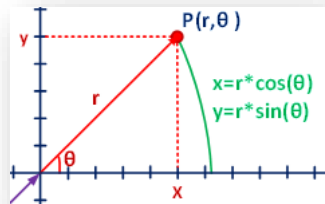


PROJECT

SUBJECT & BASIC INFORMATION

➡ WRITE A C# PROGRAM WITH FOLLOWING REQUIREMENTS

- Define a **Point2D** class:
 - ✚ Add data members of cartesian coordinates (**x** and **y**) and related properties for these fields
 - ✚ Define a default constructor with no parameters
 - ✚ Define a constructor setting initial 2D coordinates with random **x** and **y** values
 - ✚ Define a **printCoordinates()** method that prints the coordinates of the 2D point
 - ✚ Define a **calculatePolarCoordinates()** method that calculates polar coordinates (**P(r,θ)**) of this 2D point according to the figure below:



$$x^2 + y^2 = r^2$$

$$r = \text{sqrt}(x^2 + y^2)$$

$$\theta = \tan^{-1}(y/x)$$

- ✚ Define a **calculateCartesianCoordinates()** method that calculates cartesian coordinates (**P(x,y)**) of the 2D point (vice verse of converting to polar coordinates)
- ✚ Define a **printPolarCoordinates()** method that prints the pre-calculated polar coordinates of this 2D point.
- Define a **Polygon** class
 - ✚ Add **center** data member composed of **Point2D** class
 - ✚ Add **length** data member and related property for this field
 - ✚ Add **numberOfEdges** data member and related property for this field
 - ✚ Define a default constructor with no parameters
 - ✚ Define a second constructor gets initial center and length as parameter
 - ✚ Define a **calculateEdgeCoordinates()** method that calculates the vertex points of the polygon.
 - First vertex should start with a random point calculated depending on the **center** and **length** values.
 - ✚ Define a **rotatePolygon()** method that recalculates the vertex points of the polygon (rotation direction will be clockwise)

- Create a form interface including these form elements below :

- Two **textBoxes** to enter the **center** of the polygon

- range of random values for x is [0,3] and for y is [0-3]
 - set default value as **(0,0)**

- A **textBox** to enter the **length** of the polygon

- range of random values is [3-9]
 - set default value as **4**

- A **textBox** to enter **numberOfEdges** of the polygon

- range of random values is [3-10]
 - set default value as **5**

- A **textBox** to enter the angle of rotation (the first draw value should be zero)

- range of random values is [0-359]
 - set default value as **30**

- A **listBox** to write the edge coordinates in order

- A **pictureBox** to draw the graphics depending on the textboxes

- get the center point as the midpoint of the pictureBox

- A **button** that will start drawing graphics

- create a polygon object depending on the the values of text boxes except rotation angel (angel will be zero for first draw)
 - call the required functions to (re)calculate the edge coordinates
 - draw the polygon on the pictureBox and list the edge coordinates in the listBox

- A **button** that will rotate the drawn graph depending on the entered angle

- if no drawn graph exist do nothing or give warning

- A **button** that will set random value on all textboxes

NOTE: These default and random values are given to make it easier for you but, If you can't cope, you can use other values .

RULES & EVALUATION

- ➡ Name of the project should be the student number (without dot)
- ➡ To optimize the size of the assignment folder, the project should be cleaned (to clean your Solution/Project, use **Build-> Clean Solution**)
- ➡ The beginning of all .cs files should include this comment lines below

```
//*****
//**
//**          STUDENT NAME.....:          **
//**          STUDENT NUMBER.....:         **
//*****
```

- ➡ There should be comment lines for some operations (methods, specific calculations, etc.)
- ➡ **Deadline:** Control SABIS system
- ➡ You should upload **your zipped project file(s)** before deadline.
- ➡ Evaluation Criteria
 - ✚ Comment lines (student information, explaining operations like variable names, if statements, loops, etc.)
 - ✚ Obeying the variable declaration rules
 - ✚ Being readable (intendation, comments, etc.)
 - ✚ Correct compilation of the code
 - ✚ The evaluation of projects will be competitive and copied assignments will be evaluated as 0.