# Tecnología y paradigmas de programación. Lab 1.

Follow these instructions, the methodology, tips and coding style shown in theoretical lectures and previous topics. Some areas are intentionally ambiguous, the student must discover by him/herself the need for certain language structures as the instructions are followed.

Create a solution with two projects, a console application and a class library.
In the class library define the enum Color with the values Transparent, Black and Red.

Define the class Point2d that represents bi-dimensional space points with the following properties, attributes and methods:

- Attributes: two double, x and y, the point coordinates, c the point color. Add corresponding r/w properties.
- Constructors: default constructor, from two double (creates a Transparent point) and finally from two double and a Color.
- Destructor.
- Methods:
    1. Euclidean distance between two points.
    2. ToString, a Point2d with x=1.0, y=2.0, c=Red is shown as (1.0,2.0):Red.

Define the class Line that represents straight lines such as $y=mx+n$ where m is the slope and n the intercept with the following properties, attributes and methods:

- Attributes: two doubles: m and n. Add the corresponding properties.
- Constructors: default constructor and from two doubles.
- Destructor.
- Methods:
    1. Perpendicular: needs a Point2d parameter, returns a Line perpendicular that contains the Point2d parameter. If the the point is $(x_0, y_0)$ and the straight line $y=m_0 x+n_0$ , the perpendicular straight line that contains the point is $y=(-1/m_0)x+(y_0+(1/m_0)x_0)$ : m attribute is $-1/m_0$ and n attribute is $y_0+(1/m_0)x_0$
    2. Line Intersection: needs a Line parameter, returns the Point2d where two lines met. Given two straight lines $y=m_1 x+n_1$ y $y=m_2 x+n_2$ the coordinates of the intersection are $y_0=(n_1/m_1-n_2/m_2)/(1/m_1-1/m_2)$ , $x_0=(y_0-n_2)/m_2$ . The color of the result is Transparent.
    3. Distance between a Line and a Point2d: needs a Point2d parameter, returns a double with the value of the shortest distance between the point and the line. Implement using the previously defined methods. The distance between a point and a straight line equals the distance between that point and the intersection of the perpendicular from the same point to the line with precisely this line.
    4. ToString, a Line with m=1.0 and n=1.0 is shown as $y=1.0x+1.0$ . It is not hard to achieve a more appealing representation including cases such as zero or negative intercept, etc.

Write a program in the console application project with calls to all the implemented methods. Do not forget: add suitable references and set start project. Add comments (regular ones and XML). Use the usual techniques to debug the code.
**Upload the mandatory activity before the next Lab.**