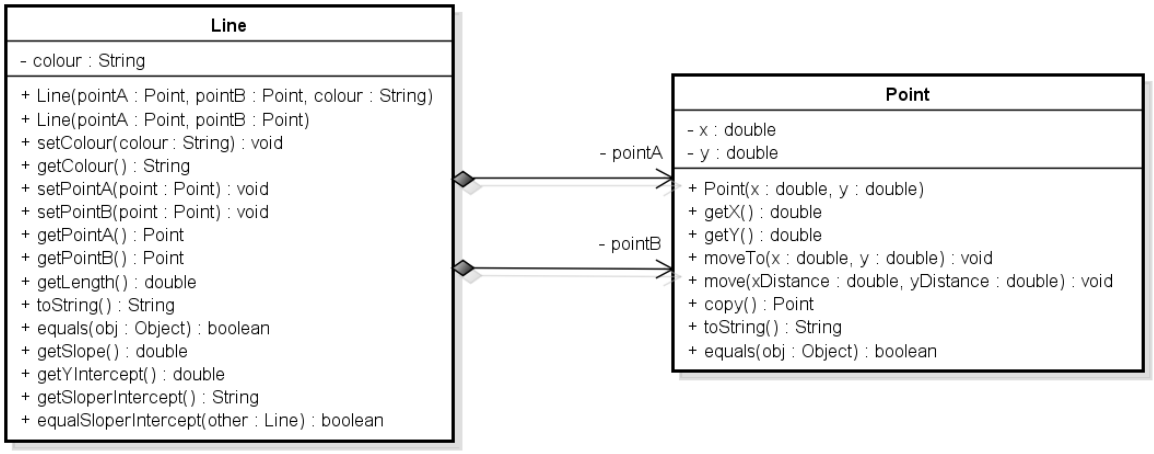
Exercise: Line



Create a class `Line` representing a line between two integer points (x_1, y_1) and (x_2, y_2) in a plane coordinate system. Note class `Point` is your solution to a previous exercise.

The class `Line` has:

a) Three instance variables, two of them representing the two points `pointA` and `pointB` both of type `Point` (from the exercise above) and the last one an instance variable `colour` of type `String` representing the colour of the line

b) Constructor with three arguments setting both points and the colour

c) Constructor with two arguments being the two points. The colour is assumed `"black"` or `"Not defined"` if not set.

d) Setters and getters for all instance variables

e) A method `length()` that return the length of the line segment bounded by the two points The length: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ where `Math.sqrt(value)` calculates the square root of `value` in Java

f) A method `toString()` that return a string with the information of the line. An example, calling the `toString` for a line with colour=”`red`” and the points (x_1, y_1) = (3, 1) and (x_2, y_2) = (7, 4) could return the string: `"A red line from (3, 1) to (7,4) with length equal to 5.0"`

g) A method `equals` returning `true` if the argument to the method is a `Line` object with the same information, i.e. same colour and same two set of points – otherwise returns `false`.

h) A method `getSlope()` that return the slope *m* for the slope-intercept form of the line (i.e. *m* from the form: y = *m*x + *b*). The slope is calculated as: $m = \frac{y_2 - y_1}{x_2 - x_1}$

i) A method `getYIntercept()` that return the y-intercept *b* for the slope-intercept form of the line (i.e. *b* from the form: y = *m*x + *b*). The y-intercept is calculated as: $b = y_1 - mx_1$

j) A method `getSlopeIntercept()` that return a sting with the slope-intercept form of the line (in the form: y = *m*x + *b*)

k) A method `equalSlopeIntercept(Line other)` that return a true if the argument other is a line with the same slope-intercept form of the line (both with the same values for *m* and *b* from the form: y = *m*x + *b*)

Implement a test class, `TestLine`, with a main method and test your solution