VIETNAM NATIONAL UNIVERSITY HCMC **School of Comp. Science and Engineering**
INTERNATIONAL UNIVERSITY **Course: Data Structures and Algorithms**

LAB No. 1

Question I: Create a Java program to do the tasks below

(TASK 1) Create a data type to support storing and working with points in 2D-space, the name of the data type is Point2D. You are required to design the data type so that users are able to:

[Constructor]

1) Create a point by passing the coordinates of the point, e.g.: new Point2D(1.5, 2.0)
2) Create a point by copying the coordinates from an old point, e.g.: new Point2D(oldPoint)
3) Create a point with default values x=0, y=0, and its real coordinates will be assigned later.
4) Create a point by calling to method named “clone” of an existing point (without any parameter), e.g.: existingPoint.clone()

[Setter/Getter]

5) Get/Set current coordinates with getters and setters. This means that you are forced to enscapsulate the data fields of points.

[Conversion to String]

6) Override toString method to represent a point as follows: “(1.50,   2.00)”

[Services]

7) Generate a list of N points randomly, the coordinates (x and y) for each point must in [-10.0, 10.0]
   - N is passed as parameter
   - This method can be invoked without using any reference to a point object.
   - You are forced to use Java array to store item in the list (do not use ArrayList or any other kind of list)
8) Compute the distance between two points given in parameters, method’s name” distanceAB
   - Parameter list has two points called a and b.
   - This method can be invoked without using any reference to a point object.
9) Compute the distance to a point specified in parameter
   - Parameter list has a point
   - This method must be invoked with a reference to point object
   -

(TASK 2) Create a data type to support storing and working with vectors in 2D-space, the name of the data type is Vector2D. You are required to design the data type so that users are able to:

[Constructor]

1) Create a vector by passing the coordinates of the vector, e.g.: new Vector2D(2.0, 3.0)
2) Create a vector by copying the coordinates from an old vector, e.g.: new Vector2D(oldVector)
3) Create a vector with default values x=1, y=0, and its real coordinates will be assigned later by setters and getters.
4) Create a vector by calling to method named “clone” of an existing vector (without any parameter), e.g.: existingVector.clone()

[Setter/Getter]

5) Get/Set current coordinates with getters and setters. This means that you are forced to enscapsulate the data fields of vectors.

[Conversion to String]

6) Override toString method to represent a vector as follows: “(2.00,   3.00)”

[Services]

7) Create a vector from point A to point B, A and B specified in parameter
   - Method name: A2B(Point A, Point B) ➔ create a vector AB (from A to B)
   - This method can be invoked without using any reference to a vector object.
8) Compute the length of vector
   - Method name: length
9) Normalize a vector. Change the vector’s coordinates so that its length equal to 1 (unit) after normalization.
   - Method name: normalize
10) Compute dot-product between two vectors specified in parameters.
    - This method can be invoked without using any reference to a vector object.
    - Method name: dot
11) Compute dot-product with a vector specified in parameter
    - This method must be invoked with a reference to vector object.
    - Method name: dotWith
12) Determine the angle between two vectors specified in parameter
    - This method can be invoked without using any reference to a vector object.
    - Method name: angle
    - Angles are measured with degree (not radian)
13) Get a vector that is perpendicular with “this” vector.
    - This method must be invoked with a reference to vector object.
    - Method name: getPerp

(TASK 3) Create a program with method main and do the following subtasks. Code each of subtasks in a separate method and call them in main.

1) Generate a list of N points and show the generated points on screen, each point on a line
   - N is assigned to 10, for example.
   - You are required to use solutions in (TASK 1) to complete this task

   a) Determine the space complexity and the computational complexity of this subtask. You are required to put a comment preceeding this method to specify the complexity, the example as following:

```
/*
Method: xxx
Objective: Generate a list of N points and show the generated points on screen, each
point on a line:
---
1/ Space complexity: O(nlogn), for example
Reasons:
        + put the reason here (many line possible, must be short explanation)
Computational complexity: O(n^2), for example
        + put the reason here (many line possible, must be short explanation)

*/
```

2) Generate two list points randomly, each has N points. The name of two lists are: listA and listB
   - N is assigned to 10, for example.
   - You are required to use solutions in (TASK 1) to complete this task

   a) Compute and return the distance between pairs of points in listA and listB
      - Pair = <point at index i in listA, point at index i in listB>, i = 0,.., N-1

   b) Show on screen pairs of points and its distance, as follows:
      (x1, y1)   (x2, y2): distance

   c) Determine the space complexity and the computational complexity of this subtask. You are required to put a comment preceeding this method to specify the complexity.

3) Generate a list of N points randomly
   - N is assigned to 10, for example.
   - You are required to use solutions in (TASK 1) to complete this task

   a) Find the pair of points in this list that has the minimum distance
   b) Find the pair of points in this list that has the maximum distance
   c) Determine the space complexity and the computational complexity of this subtask. You are required to put a comment preceeding this method to specify the complexity.

4) Generate a list of N points randomly
   - N is assigned to 10, for example.
   - You are required to use solutions in (TASK 1) to complete this task

   a) Compute the center point of the list, the center means “center of mass”. This center points named as “centerPoint”
   b) Find the point in the list that is farthest to “centerPoint”, named this point “farthestPoint”.
   c) Find the point in the list that is nearest to “centerPoint”, named this point “nearestPoint”.
   d) Compute the angle between vector vF and vN, where, vF is vector from “centerPoint” to “farthestPoint”, and vN is vector from “centerPoint” to “nearestPoint”
   e) Show on screen: centerPoint, farthestPoint, nearestPoint, vF, vN, and angle <vF, vN>
   f) Determine the space complexity and the computational complexity of this subtask. You are required to put a comment preceeding this method to specify the complexity.