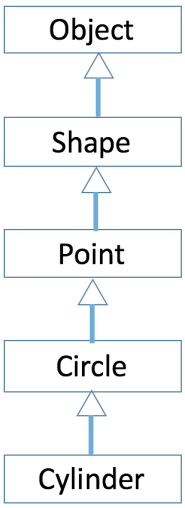
University of Science and Technology of Hanoi
Address: Building 2H, 18 Hoang Quoc Viet, Cau Giay, Hanoi
Telephone/ Fax: +84-4 37 91 69 60
Email: officeusth@usth.edu.vn
Website: http://www.usth.edu.vn

Lab Session 5 : Inheritance and Polymorphism

Task 1: Design and Implement the following multi-level inheritance structure:



- Object class is the parent class of all classes in Java
- Shape class is an abstract class and is a child of Object class. Shape class contains two regular methods and one abstract method
  - Regular method calArea(): return the area of the shape
  - Regular method calVolume(): return the volume of the shape
  - Abstract method getName(): return the name of the shape
- Point is a regular class and is a child of Shape class.
  - A point is defined by two coordinates (x & y)
  - Point class inherits/overrides regular methods of Shape class and implements abstract method of Shape class
- Circle is a regular class and is a child of Point class
  - A circle is defined by two coordinates (x & y) of the center and radius r
  - Circle class inherits/overrides regular methods of Shape/Point class and implements abstract methods of Shape/Point class
- Cylinder is a regular class and is a child of Circle class
  - A cylinder is defined by two coordinates (x & y) of the center, radius r and height h
  - Cylinder class inherits/overrides regular methods of Shape/Point/Circle class and implements abstract methods of Shape/Point/Circle class

Task 2: Develop a "ShapeTestDrive" Java program to check the inheritance relationship of Point, Circle, Cylinder with the Shape class.

- Use polymorphism concept to create an array of objects "Point, Circle and Cylinder"
- Browse created polymorphic array to perform the four following operations for each element of the array:
  - Get name of the object to see if it is a Point or a Circle or a Cylinder
  - Calculate the area of the object
  - Calculate the volume of the object
  - Display name, area and volume of each object to the screen

Task 3: Re-do the task 1 and 2 but using **Interface** instead of abstract classes as much as possible.