1. Object Oriented Part (Java) [8% of the final course grade]

Since this solution must follow the object-oriented paradigm, your program must be composed of a set of classes. Specifically, it must include, among others, the following classes:

- The `Point3D` class that includes
  - getX, getY and getZ methods returning double
    - `public double getX()`
    - `public double getY()`
    - `public double getZ()`
- The `Plane3D` class that includes
  - A constructor from 3 points
    - `public Plane3D(Point3D p1, Point3D p2, Point3D p3)`
  - A constructor from parameters
    - `public Plane3D(double a, double b, double c, double d)`
  - A getDistance method that returns the distance from a point to the plane
    - `public double getDistance(Point3D pt)`
- The `PointCloud` class that includes
  - A constructor from a xyz file
    - `PointCloud(String filename)`
  - An empty constructor that constructs an empty point cloud
    - `PointCloud()`
  - A addPoint method that adds a point to the point cloud
    - `public void addPoint(Point3D pt)`
  - A getPoint method that returns a random point from the cloud
    - `Point3D getPoint()`
  - A save method that saves the point cloud into a xyz file
    - `public void save(String filename)`
  - An iterator method that returns an iterator to the points in the cloud
    - `Iterator<Point3D> iterator()`
      - This iterator should include hasNext, next and remove methods (e.g. the iterator from an ArrayList)
- The `PlaneRANSAC` class that includes
  - A constructor that takes as input a point cloud
    - `public PlaneRANSAC(PointCloud pc)`

- setter and getter for the epsilon value
  - `public void setEps(double eps)`
  - `public double getEps()`
- a method that returns the estimated number of iterations required to obtain a certain level of confidence to identify a plane made of a certain percentage of points
  - `public int getNumberOfIterations(double confidence, double percentageOfPointsOnPlane)`
- a run method that runs the RANSAC algorithm for identifying the dominant plane of the point cloud (only one plane)
  - `public void run(int numberOfIterations, String filename)`
    - filename being the xyz file that will contain the points of the dominant plane
    - this method will also remove the plane points from the point cloud

In addition to the source code of your solution, you must also submit a document that includes a UML diagram of all your classes (showing attributes, associations and methods). Do not use static methods, except for the `main` *function. This document must also include all references used to build your solution.*

You must use the object-oriented paradigm.