**Week 4 assignment**

Ex1)

* **Point2D** class that represents a point with x and y coordinates that contains:

1. Private double data fields x and y with their respective getters and setters
2. No-args constructor that initializes an object with x=0 and y=0
3. Constructor that initializes x and y with the given arguments
4. Public method called getDistance that returns the distance from this point to another Point2D type. Method signature is the following:  
   public double getDistance (Point2D);
5. Public method called getDistance that returns the distance from this point to another point specified by x and y primitive values. Method signature is the following:  
   public double getDistance(double, double);
6. Static method named getDistance that returns the distance between two Point2D objects. Method signature is the following:  
   public static double getDistance(Point2D, Point2D);

* Triangle class that contains:

1. 3 private data fields of type Point2D named p1, p2, p3
2. Getters and setters for each point (p1, p2, p3)
3. No args constructor that creates a triangle at points (0, 0), (0, 1) and (2, 0)
4. Constructor that takes 3 Point2D arguments and initializes p1, p2, p3
5. Public method getPerimeter()
6. Public method getArea()  
   \*You can use Heron’s formula to find the area or any algorithm you want
7. Public method contains(Point2D p) that returns true if point p is inside the triangle

EX2) Regular Expressions

To solve this exercise, have a look at this class in GitHub repo <https://github.com/iglidraci/cen215-java/blob/main/Week4/src/RegExp.java>

There you can find some simple regular expressions that we can use in Java to match some patterns. These are enough for you to solve the exercise.

If you want to have a deeper understanding of regex follow the Oracle documentation <https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>.

The exercise is as following:

* Create a class named Student with these properties: firstName, fathersName, lastName and creditCard. All these properties are of type String
* Create getters and setters for each property
* Create a constructor that initializes these 4 properties of class Student
* Create another test class named TestStudent
* Inside TestStudent class create the main method to execute the program (as we have done so far in all lab sessions we have had together)
* Inside the main method create an array with 10 students, initialize them with any values you want
* Create a method inside TestStudent class called filterByFullName that returns another array of Student but with those students that their full name (firstName + fathersName + lastName) contains a certain substring. This substring will be the argument to the method filterStudentsByFullName. Method signature is as following:  
  public static Student[] filterStudentsByFullName(Student[] students, String substring);  
  Make sure to ignore lower or uppercase characters. Ignore special characters, that means if someone’s name is “Hëna”, I should be able to find him/her with substring ”hena”. Basically you have to replace “ç” with “c” and “ë” with “e” before matching. In that way you are making a proper filtering. Use regular expressions to match the patterns.  
  For example: if we have an array with students {“Arjan Bob Çela”, “X Y Z” and “X Y Çela” } and we call our filter method with substring “cela” than our method should return an array with 2 elements only {“Arjan Bob Çela”, “X Y Çela” }.
* Create a method inside TestStudent class called filterCorrectCreditCards that returns another array of Student but with those students that their credit card is in the correct format. The method signature is as following:  
  public static Student[] filterCorrectCreditCards (Student[] students);  
  Correct credit card format is considered : 4digits-4digits-4digits-4digits  
  If student’s credit card is in this format than include it in the returning array otherwise don’t.