Assignment 4 420-PA3-AS Deadline: 13-March-2022 at 23:55

1. Create a java project with a basic package called basicClasses contains the **Person, Candidate, DoCertificationExam, CertificationExam**.
   Also, in the same package create an interface names **ServiceCertification.**

2. Create the class Person with the following characterises:

   - Attributes and Constants:
     - Last Name type String
     - First Name type String
     - Phone type String
   - Constructor:
     - Create a parameterized constructor
   - Methods:
     - The class provides access methods to all attributes and modifies the attributes phone, last name and first name
     - The class overrides the method **toString()** to return the following output :

     First Name: …….. Last Name : ……. Phone : ……

3. Create the class CertificationExam with the following characterises:

   - Attributes and Constants:
     - id type String
     - title type String
     - successMark type float
     - number of days wait type int
   - Constructor:
     - Create a parameterized constructor
   - Methods:
     - The class provides access methods to all attributes and modifies the attributes number of days wait, successMark
     - The class overrides the method **toString()** to return the following output :

   ID: …….. Title: ……. Success Mark: …… Number of Days to Wait: ……

4. Create the interface ServiceCertification with the following characterises:

   - Attributes and Constants:
     - Two final objects from CertificationExam class
     - An array of CertificationExam with these objects

- Methods:
  - serviceSuccess that takes a grade type String (A+, A , B+, B, C+, C, D+, D)
  - serviceFaillure that takes number of days type int

5. Create the class Candidate that inherits from Person and implements from ServiceCertification with the following characterises:

- Attributes and Constants:
  - certificationExam type CertificationExam
  - examDate type String
  - examMark type float
  - grade type String
  - nbDaysToWait type int

- Constructor:
  - Create a parameterized constructor and initialize super class with all attributes of super class and also examDate and certificationExam. Hint: create an object of DoCertificationExam in this constructor (explain this class at step 6)

- Methods:
  - The class provides access methods and modifies the attributes for examDate, examMark and certificationExam
  - serviceSuccess takes grade and initialize the grade attribute
  - serviceFaillure takes number of days and initialize the nbDaysToWait attribute
  - The class overrides the method **toString()** to check whether the candidate passed or failed the exam and display a message for success and failure with a message return the following output :
  -

Pass Certification exam: …….. Certification Exam id: …….. Exam Title: …… Mark: ……

Fails Certification exam: …….. Certification Exam id: …….. Exam Title: …… Mark: …… Number Of Days to Wait ……

6. Create the class DoCertificationExam with the following characterises:

- Attributes and Constants:
  - candidate type Candidate
  - serviceCertification type ServiceCertification

- Constructor:
  - Create a parameterized constructor with all attributes and call the method validateCertification.

- Methods:
  - The class provides access methods to all attributes and modifies the attributes number of days wait, successMark
  - getGrade takes examMark type float
  - getMark takes examId and Candidate Last name. the method reads from a text file named marks.txt and retrieve the exam mark from file from the last name and exam ID.
  - validateCertification takes no argument and check whether if the mark is passed, then check the grade according to the mark and then set it as the parameter of serviceSuccess. If not passed, set the number days for waiting for serviceFaillure.
  - The class overrides the method **toString()** to return the following output :

(examMark >=95) → A+
(examMark >=90 && examMark<95) → A
(examMark >=85 && examMark<=90) → B+
(examMark >=80 && examMark<=85) → B
(examMark >=75 && examMark<=80) → C+
(examMark >=70 && examMark<=75) → C
(examMark >=65 && examMark<=70) → D+
(examMark >=60 && examMark<=65) → D
(examMark <60) → E

Deliverable:
1) Create one zip file, containing the necessary source-code files (java)

You must name your file using the following convention:
A#_studentID, where # is the number of the assignment. studentID is your student ID number.
2) Assignments must be submitted in the assignment section (Léa) by 13-March– 2022 at 23:55.