**משימה – JAVA API**

**מטרה**

במשימה זו נתרגל תוכנה מודרנית עם APIs נפוצים בשפת Java

**דרישות**

* עליכם ליצור מחלקה עם מתודה ראשית main()
* במתודה main יש להדפיס אובייקטים מהמחלקות השונות
* אין צורך לבצע בקרת קלט, הניחו כי הקלט תקין

**שלב 1**

בשלב זה נתרגל את האלמנטים הבאים:

* תכנות מונחה עצמים בסיסי
* עבודה עם Factory Method
* עבודה עם Collections
* עבודה עם Enum

ממשו את המחלקה **Employee** באופן הבא:

1. Attributes – מאפיינים
   * - name: String
   * - salary: double
2. Constructors – בנאים
   * + Provide constructor that takes all the arguments
3. Methods – מתודות
   * + Provide getters/setters methods
   * + toString(): String
   * + Override equals()
     + return true for two employees with the same **name** and **salary**

ממשו את המחלקה **Secretary** היורשות מ- **Employee** באופן הבא:

1. Attributes – מאפיינים
   * - office: String
2. Constructors – בנאים
   * + Provide constructor that takes all the arguments
3. Methods – מתודות
   * + Provide getters/setters methods
   * + toString(): String

ממשו את המחלקה **Engineer** היורשות מ- **Employee** באופן הבא:

1. Attributes – מאפיינים
   * - specialty: String
2. Constructors – בנאים
   * + Provide constructor that takes all the arguments
3. Methods – מתודות
   * + Provide getters/setters methods
   * + toString(): String

ממשו את המחלקה **Manager** היורשות מ- **Employee** באופן הבא:

1. Attributes – מאפיינים
   * - department: String
2. Constructors – בנאים
   * + Provide constructor that takes all the arguments
3. Methods – מתודות
   * + Provide getters/setters methods
   * + toString(): String

ממשו את המחלקה **Director** היורשות מ- **Manager** באופן הבא:

1. Attributes – מאפיינים
   * - group: String
2. Constructors – בנאים
   * + Provide constructor that takes all the arguments
3. Methods – מתודות
   * + Provide getters/setters methods
   * + toString(): String

ממשו את ה-Enum **Name** באופן הבא:

1. Attributes – מאפיינים
   * MOSHE
   * DAVID
   * JACK
   * TAL
   * DANIEL
2. Constructors – בנאים
   * - Default empty private constructor
3. Methods – מתודות
   * + static getRandName(): String
     + return random name in lower case

ממשו את המחלקה **FactoryUtils** באופן הבא:

1. Methods – מתודות
   * + static randSalary(employee: Employee): double
     + Return a random salary **depends on Subclass Type**
       - Secretary – 7000 – 8500
       - Engineer – 9000 – 11000
       - Manager – 12000 – 15500
       - Director – 18000 – 35000
   * + static initEmployess(): List<Employee>
     + Return list of 10 random Employees as the following description:
       - 2 secretaries
       - 2 managers
       - 3 employees
       - 1 director
       - 2 engineers

ממשו את המחלקה **Helper** באופן הבא:

1. Methods – מתודות
   * + static avgSalary(employees: List<Employee>): double
     + Return the average salary
   * +static avgManagementSalary(employees: List<Employee>): double
     + Return the average salary of managing level (Managers and Directors)
   * + static display(employees: List<Employee>): void
     + Print the collection using **foreach loop**
   * + static report(employees: List<Employee>): void
     + **"conducts the orchestra”** and call of the above methods

ממשו את המחלקה **Test** באופן הבא:

1. צרו מבנה נתונים בשם employess
   * יש לבחור במבנה נתונים המסודר **ע"פ סדר ההכנסה**
   * מאותחל וממומש לעבודה **כרשימה מקושרת**
2. הוסיפו לרשימה באמצעות המתודה addAll את הערך החוזר של ה-Factory method שיצרתם קודם
3. הדפיסו את מבנה הנתונים
4. הדפיסו את הדו"ח עבור מבנה הנתונים

**בהצלחה!**