**משימה – Design Patterns**

**מטרה**

במשימה זו נתרגל תבניות עיצוב ממשפחת ה - Creational

**דרישות**

* IDE כזה או אחר
* JAVA 11 או כל גרסה שאתם מעוניינים

**שלב א – בניית מחלקת בסיס**

הגדירו את מחלקת Person באופן הבא:

* Attributes
  + - id: int
  + - name: String
  + - city: String
  + - age: double
* Constructors
  + + Person(id: int, name: String, city: String, age: double)
  + + Person(name: String, city: String, age: double)
  + + Person(name: String, city: String)
* Methods
  + + Getters & Setters
  + + toString(): String

הגדירו את מחלקת Test באופן הבא:

מתודה ראשית main ובתוכה 3 מופעים ע"י 3 בנאים שונים.

**שלב ב – Builder Design Pattern**

מחקו לאלתר את הבנאים השונים ובמקומם ממשו Builder והביעו באמצעותו את האובייקטים ממחלקת Test שיצרתם קודם לכן

**שלב ג – Prototype Design Pattern**

דרסו (Override) את המתודה clone() והחזירו deep copy של האובייקט הנוכחי.

ניתן להיעזר ב-Copy Constructor או ב- Builder

**שלב ד – הגדרת Factory Method**

הגדירו את מחלקת Utils באופן הבא:

* Methods
  + + init(len: int,person:Person): static List<Person>
    - Generate a collection with {len} size clone objects

**שלב ה – הגדרת Singleton**

הגדירו את מחלקת Motorola באופן הבא:

את מחלקה זאת תממשו כ-Eagerly Loaded Singleton

* Attributes
  + - name: final String = “Motorola”
  + – ppl: List<Person>

Constructors

* + – Motorola()
    - Init the ppl list using the factory method you create before
* Methods
  + + Getters & Setters
  + + toString(): String
  + + display(): void
    - Print the list

**שלב ו – הגדרת Singleton**

הפכו את מחלקת Motorola ל-Lazily Loaded Singleton

**סיכום**

בתרגיל זה קודדתם את ה-Design Patterns הבאים:

* Prototype (Clone)
* Builder
* Factory Method
* Singleton

**בהצלחה!**