

# פרויקט פת"מ תשפ"ה – תרגיל 4

## רקע

בתרגיל הקודם יצרנו תשתית לאגד יחדיו סוכנים ו topics כסוג של Config. בתרגיל זה ניצור קונפיגורציה גנרית שנטענת מתוך הגדרות בקובץ. המטרה בהמשך היא שהמשתמש יוכל להיכנס דרך הדפדפן לאפליקציית רשת שרצה על מכונה מרוחקת כלשהי, ואז לטעון קובץ קונפיגורציה. המכונה תטען את ההגדרות מתוך הקובץ ותריץ את הסוכנים בהתאם.

כדי לשמור על היבטים של אבטחה, המשתמש לא יוכל להעלות קוד (למשל קובץ class מקומפל) של סוכן חדש, אלא אך ורק קובץ הגדרות שמחווט מחדש את הסוכנים שכבר קיימים במכונה.

## תרגיל 4:

תחילה נעדכן את הממשק Config, כך שיכיל מתודות סגירה.

```
public interface Config {
    void create();
    String getName();
    int getVersion();
    void close();
}
```

לתלמידי פת"מ2: הקונפיגורציה הגנרית שלנו עתידה להריץ מספר סוכנים במקביל (מופעים של ParallelAgent) והמתודה close תאפשר לנו לבצע סגירה מסודרת של כולם.

כעת עליכם ליצור 2 סוכנים פשוטים לדוגמה (ב configs package).

כדי לשמור על כלליות, הבנאי של כל סוכן יקבל שני פרמטרים:

- מערך של מחרוזות בשם subs המציין את שמות ה Topics שאליהם ירשם סוכן זה
- מערך של מחרוזות בשם pubs המציין את שמות ה Topics שאליהם יפרסם סוכן זה

הסוכנים הם:

- PlusAgent – סוכן שמבצע פעולת +.
  - נרשם לשני ה Topics הראשונים מתוך subs, ומעדכן על פיהם ערכים פנימיים x,y בהתאמה. (ערך ברירת המחדל ל x,y הוא 0)
  - בכל עדכון, אם גם x וגם y הינם מספרים חוקיים, אז הוא יחשב את x+y ויפרסם את התוצאה ב Topic הראשון של pubs.
- IncAgent – סוכן שמגדיל ערך כלשהו ב 1.
  - נרשם ל Topic הראשון של subs.
  - כאשר מגיע לשם ערך מספרי חוקי, הוא מגדיל אותו ב 1 ומפרסם את התוצאה ב Topic הראשון של pubs.

כעת צרו תיקייה מחוץ לתיקיית הפרויקט, קראו לה config\_files, וצרו בתוכה קובץ טקסט בשם simple.conf שזהו תוכנו:

```
project_folder.configs.PlusAgent
A,B
C
project_folder.configs.IncAgent
C
D
```

\* עליכם להחליף את "project\_folder" בשם תיקיית הפרויקט

במציאות היינו משתמשים בפורמט ידוע כמו JSON, אך לשם הפשטות הקובץ שלנו מוגדר כך:

- כל סוכן מוגדר ע"י 3 שורות
  - השם המלא של טיפוס הסוכן.
  - רשימת שמות ה Topics מופרדים בפסיק שעל הסוכן להירשם אליהם.
  - רשימת שמות ה Topics מופרדים בפסיק שהסוכן עשוי לפרסם אליהם.

הקובץ מוגדר מאוסף הגדרות של סוכנים.

בדוגמה לעיל שם הטיפוס של הסוכן הראשון הוא project\_folder.configs.PlusAgent וזאת משום שהוא נמצא בפרויקט ששמו project\_folder (תחליפו ע"פ הצורך) בתוך package בשם configs, ושם המחלקה הוא PlusAgent. ניתן לראות שהוא נרשם ל Topics הקלט שלו A,B ומפרסם את התוצאות ל C. ואילו הסוכן של IncAgent מאזין ל C ומפרסם את התוצאה ל D.

כעת עליכם ליצור את המחלקה GenericConfig.

- היא סוג של Config
- באמצעות המתודה setConfFile היא תגדיר את שם קובץ הקונפיגורציה
- במתודה create היא תיצור את הקונפיגורציה ע"פ ההגדרות בקובץ. עוד על כך בהמשך.
- במתודה close היא תקרא לסגירה של כל הסוכנים שנוצרו.

כיצד ניצור את המופעים ע"פ ההגדרה בקובץ?

- רצוי תחילה לקרוא את כל שורות הקובץ לתוך רשימה. אם גדולה מתחלק ב 3 ללא שארית כנראה שהקלט תקין ואפשר להמשיך.
- כעת נעבור על כל 3 שורות ונחלץ את הנתונים בהתאם לפורמט.
- משימת למידה עצמאית:
  - קראו על `Class<?>` וכיצד מיצרים מופע שלה בהינתן שם המחלקה.
  - קראו בדוקומנטציה של Java על המתודה `getConstructor`. היא תאפשר לכם להגדיר את טיפוס הפרמטרים שהבנאי מקבל ותחזיר לכם אפשרות ליצור מופע (של סוכן) באמצעות בנאי זה.
- את הסוכן שיוצרתם יש לעטוף באמצעות `ParallelAgent`, וכדאי לזכור אותם ברשימה כדי שתוכלו לסגור אותם אח"כ במתודה `close`.
  - פת"מ2: עליכם להגיש את מימוש ה `active object`
  - פת"מ1: עליכם להגיש אותו כ `decorator` שעוטף סוכן ומריץ בפשטות את המתודות שלו ללא כל מניפולציה (כהכנה לקוד שתממשו בפת"מ2).

### **בדיקה:**

זכרו לשנות את שם ה package ל test עבור הבדיקה במערכת הבדיקות.

מוד האימון כולל בדיקה פשוטה עם הסוכנים inc i plus המוגדרים לעיל, שמקבלים באמצעותם תוצאה נכונה עבור קלט אקראי וכן שכל הת'רדים נפתחים כראוי ונסגרים בסוף.

מוד ההגשה כולל בדיקה חסויה של קונפיגורציה אחרת עם סוכנים נוספים.

נא להגיש את הקבצים ע"פ ההוראות בדף המודול.

בהצלחה!