

## פרויקט פת"מ תשפ"ה – תרגיל 2

### רקע

כזכור מהתרגיל הקודם, ברגע שפורסמה הודעה ל Topic מסוים, ה Topic הריץ בזו אחר זו את מתודות ה callback אצל כל אחד מהסוכנים שנרשמו לנושא זה. אולם, אם אחד מהם מבצע חישוב אורך, זה יעכב את שליחת ההודעה לסוכנים הבאים אחריו בתור.

תרגיל זה מיועד לקורס פת"מ 2 (בלבד). בתרגיל זה אנו נממש concurrency pattern הנקראת Active Object שתפקידה להפריד בצורה חכמה בין הת'רד היוזם קריאה למתודה לבין הת'רד שמריץ אותה בפועל. את התבנית הזו נשלב במחלקה ParallelAgent אשר תהווה אף Decorator ל Agent.

### תרגיל 2:

בפרויקט, ב package בשם graph שיצרנו בתרגיל הקודם, צרו את המחלקה ParallelAgent כ Decorator של Agent. כלומר עליו להיות סוג של Agent מחד, ולהכיל מופע של Agent מאידך (נקרא לו מעטה agent). את המופע הזה הוא יאתחל בבנאי, ובאמצעותו יממש את כל המתודות של הממשק Agent שעליו לממש.

אולם, ללא תרומה כלשהי מחלקה כזו תהיה חסרת ערך. ולכן כעת נממש את תבנית Active Object.

המחלקה תכיל תור מסוג BlockingQueue של Message.

- הוא thread safe
- בעל המתודה take אשר תחזיר לנו את ה Message שבתחילת התור, או שתמתין כל עוד התור ריק.
- בעל המתודה put אשר תכניס Message לתור או שתמתין כל עוד התור הגיע לקיבולת המקסימלית שלו.

את התור יש לאתחל בבנאי להיות מסוג ArrayBlockingQueue. תור זה דורש כפרמטר את הקיבולת (capacity) של התור שאותה תבקשו בבנאי של ParallelAgent כפרמטר שני (בנוסף ל agent).

### **פעולת התבנית:**

- במתודה callback, במקום להריץ את ה callback של agent - הסוכן המוכל, נכניס פשוט הודעה לתור (באמצעות המתודה put).
  - כך לא משנה כמה זמן עלול לקחת ה callback של agent, ה callback של ParallelAgent לוקח כלום זמן, שכן הוא רק מכניס אובייקט לתור וזהו.
  - ההודעה שנכניס לתור צריכה להכיל גם את ה Topic וגם את ה Message שקיבלנו כפרמטרים במתודה callback (של ParallelAgent).
- בבנאי רץ ת'רד שמריץ לולאה ששולפת הודעות בזו אחר זו מהתור באמצעות המתודה take
  - נשים לב שהת'רד הזה למעשה נח כאשר התור ריק
  - מכל הודעה נחלץ את ה Topic ואת ה Message המקוריים ונפעיל את ה callback של הסוכן המוכל agent.
  - נשים לב שכך הפרדנו בין הת'רד שיזם את הפעולה לזה שמריץ אותה. למעשה כל הפעולות ירוצו בזו אחר זו בת'רד הזה שפתחנו בבנאי.

- המתודה close צריכה גם להביא לסיום פעולתו של הת'רד לעיל, כלומר להביא לסוף הלולאה. גם אם היא תקועה כרגע על take כאשר התור ריק.

### אופי הבדיקה:

בבדיקה נבדוק שאכן מופע של ParallelAgent מייצר ת'רד אחד נוסף בו הוא מריץ את הסוכן שעטף כמו שצריך וכן שקריאה ל close ביצעה יציאה מסודרת ולא נותרו ת'רדים פתוחים.

### הגשה

- עליכם להגיש את המחלקה לעיל ע"פ ההוראות במודול \ במערכת הבדיקות
- זכרו לשנות את ה package ל test.
- כמו קודם, תקבלו רק אינדיקציה שאין שגיאות קומפילציה או ריצה, אך לא תקבלו פלט של טסטים שנכשלו. וזאת כדי לעודד אתכם לכתוב בדיקות משלכם.
- יש להגיש את התרגיל במוד הגשה כדי שיחשב כמוגש

בהצלחה!