

פרויקט פת"מ תשפ"ה – תרגיל 3

רקע

כעת כאשר יש לנו את התשתית לייצר סוכנים ו Topics והודעות אנו צריכים ישות שתוכל לאגד את היצירה שלהם ולמעשה ליצור גרף חישובי. היא פשוט תייצר את הסוכנים, שבתורם ייצרו Topics ע"י הרשמה אליהם או פרסום דרכם, והנה לנו גרף חישובי מכוון. בתרגיל זה נגדיר ישות זו כ Config (עבור קונפיגורציה), נגדיר גרף חישובי פשוט, ונוודא שהוא אינו מכיל מעגלים.

תרגיל 3:

הפעם ניצור package בשם configs. צרו את המחלקה **Node** המייצגת קודקוד בגרף מכוון.

- השדות יהיו:
 - name – שם הקודקוד כמחרוזת
 - edges – רשימה של Node המייצגת למעשה את הקשתות היוצאות מקודקוד זה אל הקודקודים שברשימה.
 - message – ההודעה שקודקוד זה עשוי להכיל.
- הבנאי יקבל את שם הקודקוד
- צרו setters & getters לכל השדות
- המתודה addEdge() תוסיף Node לרשימה edges.
- המתודה hasCycles() תחזיר אמת או שקר האם הקודקוד הנ"ל שייך לרכיב קשירות בגרף שמכיל מעגלים
- רמז: בהינתן קודקוד ניתן לעבור רקורסיבית על כל הצאצאים שלו, אם אחד מהם זה הוא עצמו הרי שיש מעגל.
- כמובן, תוכלו להיעזר במתודת עזר פרטית.

כך למעשה נוכל להגדיר גרף כרשימה של Nodes. כל קודקוד יחזיק את רשימת הקודקודים שניתן להגיע אליהם ממנו, וכדי לוודא שאין מעגלים בכל הגרף, בכל רכיבי הקשירות שלו, פשוט שכל קודקוד מחזיר false במתודה hasCycles.

אופי הבדיקה:

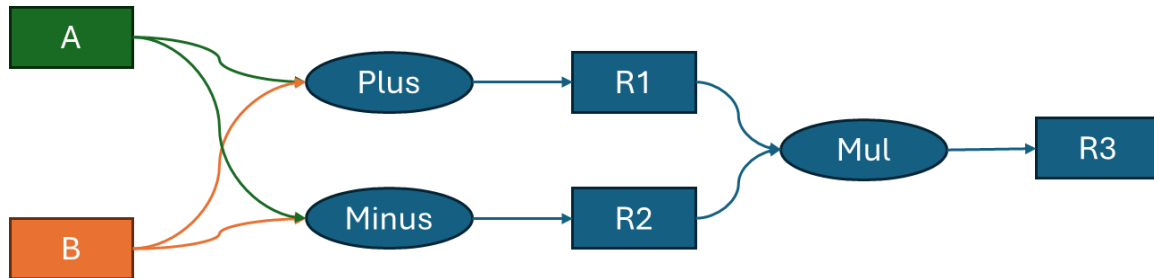
נבדוק שבהינתן כמה גרפים פשוטים (רשימות של Node המהווים גרפים) נקבל תשובה נכונה כאשר יש או כשאין מעגלים.

כעת, נתון הממשק Config:

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

המתודה create נועדה כדי ליצור מופעים של סוכנים והרשמות נוספות ל Topics (במידת הצורך).

נתונה גם דוגמה למימוש פשוט במחלקה MathExampleConfig שמייצרת את הגרף החישובי הבא:



כאשר A,B הם Topics המהווים קלט של הקודקודים Plus ו Minus שמפרסמים את התוצאות שלהם ל Topics של R1 ו R2 בהתאמה – שבתורם מהווים את הקלטים של הקודקוד Mul – שמפרסם את התוצאה ל Topic בשם R3.

ניתן לראות את החישוב הזה גם כמו $r3 = (a+b)*(a-b)$

```
public class MathExampleConfig implements Config {

    @Override
    public void create() {
        new BinOpAgent("plus", "A", "B", "R1", (x,y)->x+y);
        new BinOpAgent("minus", "A", "B", "R2", (x,y)->x-y);
        new BinOpAgent("mul", "R1", "R2", "R3", (x,y)->x*y);
    }

    @Override
    public String getName() {
        return "Math Example";
    }

    @Override
    public int getVersion() {
        return 1;
    }

}
```

כפי שניתן לראות, כדי שזה יעבוד, עליכם ליצור את המחלקה BinOpAgent המייצגת סוכן שמבצע פעולה בינארית. בבנאי נקבל:

- שם הסוכן
- שם ה Topic של הקלט הראשון
- שם ה Topic של הקלט השני
- שם ה Topic של הפלט

- ביטוי למבדה של פעולה בינארית.

תזכורת: ביטוי למבדה הם דרך נהדרת להשתמש בתבנית העיצוב Strategy – המאפשרת להעביר "פונקציה" כפרמטר בשפה מכוונת עצמים. ב Java, כדי לאפשר הזנה של ביטוי למבדה נצטרך לקבל פרמטר מסוג functional interface – כלומר ממשק בעל מתודה חסרת מימוש אחת בלבד. במקרה שלנו נוכל להשתמש בממשק הקיים BinaryOperator כאשר הטיפוס הפרמטרי שלו הוא כמובן Double.

מימוש: על הסוכן להירשם ל Topics של הקלט (כבר בבנאי), להפעיל את הפעולה הבינארית במתודה callback (אם אכן בכל אחד משני הקלטים יש הודעה מסוג Double) ולפרסם את התוצאה ל Topic הפלט.

במתודה reset תאתחל את הקלטים ל 0.

אופי הבדיקה:

ניצור גרפים חישוביים דומים ונוודא שאנו מקבלים את התוצאה הרצויה.

כעת עלינו לבדוק שקונפיגורציה מסוימת אינה מכילה מעגלים. לשם כך עלינו ליצור את **המחלקה Graph**.

- היות וגרף אצלנו הוא פשוט רשימה של Nodes, המחלקה Graph תירש ArrayList של Nodes.
- המתודה hasCycles תחזיר אמת או שקר בהתאמה האם הגרף, על כל רכיבי הקשירות שלו, מכיל מעגל או לא.
- המתודה createFromTopics תאתחל את הגרף מתוך הרשומות הקיימות ב TopicManager
 - כל Topic הוא קודקוד ממנו יוצאות קשתות לכל הסוכנים שנרשמו אליו (ע"פ רשימת subs)
 - כל Agent הוא קודקוד ממנו יוצאות קשתות לכל ה Topics שהוא עשוי לבצע להם publish (ע"פ רשימת pubs).
 - שם כל קודקוד הוא שם ה Topic או ה Agent שעבורו הוא נוצר, כאשר האות הראשונה בשמו תהיה T או A בהתאמה (כדי שנוכל מאוחר יותר להביל ולצייר אתם אחרת).
 - הקפידו על כך שאינכם מייצרים קודקודים מיותרים בתהליך.

אופי הבדיקה:

נייצר מופע של Graph מתוך קונפיגורציות שונות, ונבדוק את שמות הקודקודים, מבנה הגרף, והאם יש להן מעגלים או לא.

הגשה

- עליכם להגיש את המחלקה לעיל ע"פ ההוראות במודול \ במערכת הבדיקות
- זכרו לשנות את ה package ל test.
- יש להגיש את התרגיל במוד הגשה כדי שיחשב כמוגש

בהצלחה!