

כתיבת סוכן ממוחשב למשא-ומתן:

1. בעבודה זו עליכם לכתוב סוכן שמנהל מו"מ מול סוכן ממוחשב אחר. המו"מ יתנהל בתוכנת Genius. עליכם להכין אסטרטגיה שתגבור ככל הניתן על הסוכנים האחרים. המשא-ומתן יתנהל בסביבה (Domain). סביבה כוללת מספר נושאים (Issues) שצריכים להתדיין במו"מ. סביבה מכילה 2 פרופילים. כל פרופיל מייצג צד במו"מ, ובו כתובים הערכים הרצויים של הסוכן (Utilities).
 - לדוגמה: הסביבה היא "ציאה לבילוי". נניח שבסביבה הזו יש נושא אחד בלבד שיידון, שהוא "לאן יוצאים". ניתן להגיע לאחד מ-2 הסכמים – לראות הצגה או לראות סרט. הפרופיל של צד אחד נותן את הערך 1.0 ל"הצגה" ואת הערך 0.2 ל"סרט". הפרופיל של הצד השני נותן את הערך 1.0 ל"סרט" ו-0.5 ל"הצגה". אם במידה והצדדים יסכימו על "סרט", אז הסוכן שייצג את הצד הראשון יקבל Utility של 0.2, והסוכן שייצג את הצד השני יקבל Utility של 1.0. אם יש מספר נושאים שיידונו (לדוגמה, גם "לאן יוצאים" וגם "מתי יוצאים"), הערך הסופי שיינתן יהיה מכפלת המשקולות של הערכים (לכל ערך יש משקל בפרופיל – סך המשקולות בפרופיל שווה 1).
2. עליכם לקרוא את קובץ ה-PDF באנגלית שמסביר את העקרונות מאחורי התוכנה והסברים מפורטים כיצד לממש את הסוכן שלכם (בדגש על סעיפים 1 ו-5) למסמך שהוא בוצעו מספר עדכונים ושינויים, והם מובאים במסמך זה. **יש לעבור על 2 המסמכים לפני שמתחילים לכתוב את הסוכן.**
3. בתוכנת Genius המקורית, המתוארת בקובץ PDF המצורף, כל זוג סוכנים ביצעו מו"מ פעם אחת בלבד. בתרגיל זה, התוכנית שונתה קצת, כך שכל זוג סוכנים יבצע מספר סיבובים (Sessions) של מו"מ. לדוגמה, אם מספר הסיבובים הוא 3, אז שני הסוכנים יבצעו 3 מו"מ, שבכל אחד מהם הסוכנים יצטרכו להגיע להכרעה מחדש. הרעיון הוא, שבתחילת הסיבוב השני יהיה לסוכן שלכם מידע שהוא צבר מהסיבוב הראשון, שהוא יוכל לנצל כדי לבצע את המו"מ בצורה טובה יותר. עדיין – בכל אחד מהסיבובים על הסוכן להשיג את התוצאה הטובה ביותר עבורו.
 - דוגמה לאסטרטגיה: אם ראיתם בסיבוב הראשון שהסוכן שמו לכם היה מאוד מתפשר, אולי כדאי לכם להיות יותר קשוחים בסיבוב השני ולהשיג כך תוצאה טובה יותר עבורכם. (על השיטה לזיהוי "מתפשר" אתם צריכים לחשוב וליישם)
4. עליכם להגיש את הפרטים הבאים:
 - **סוכן** ← קובץ/קבצי Java המכילים את הקוד להרצה.
 - **אסטרטגיה** ← קובץ וורד / PDF המסביר את האסטרטגיה של הסוכן
 - **סביבה (Domain)** ← 3 קבצי XML המכילים את נתוני הסביבההניקוד שלכם יקבע לפי 2 קריטריונים: התחכום של האסטרטגיה והביצועים מול הסוכנים האחרים. הגשה תתבצע ל-FTP: <ftp.ise.bgu.ac.il/Agents/BGU>
5. בכתיבת הסביבה יש להיעזר במדריך שבקובץ השני ← סעיף 3. כמו כן, ניתן להוסיף לסביבה (ל-2 הקבצים שמייצגים את 2 צדדי המו"מ) עוד מאפיין הנקרא Reservation value. המאפיין הזה מתייחס ל-Utility של כל סוכן במידה והמו"מ נגמר ללא הסכמה (או בסיום הזמן החוקי שהוקצה למו"מ). לדוגמה, אם ה-Reservation value של הסביבה יהיה 0.4, אזי במידה והמו"מ יסתיים בכישלון, כל אחד מהצדדים יסיים עם $UtilityValue = 0.4$. את הסביבה אפשר לכתוב באמצעות התוכנה. פרטים על כך אפשר לראות [בסרטון הזה](#). יחד עם זאת, את ערך ה-RV וה-Discount factor יש לרשום בקבצי ה-XML **באופן ידני**, לדוגמה באופן הבא:
<reservation value="0.4" />

6. תיאור השיטות למימוש הסוכן (מודגשות – הפונקציות שנוספו לתוכנה הנוכחית ולא מוסברים בקובץ השני):

שם הפונקציה	תיאור
Void init()	אתחול הסוכן. פעולה זו מתבצעת בתחילת המשא ומתן הראשוני.
Void beginSession(int sessionNumber)	אתחול מו"מ חדש. בנקודת הזמן הזו ניתנת לכם ההזדמנות לשנות או לעדכן את האסטרטגיה, כתוצאה של היסטוריית הסיבובים הקודמים.
void endSession(Action LastAction)	סיום מו"מ. במצב זה תקבלו את תוצאת המו"מ. יכול להיות מצב של הסכמה או מצב של אי-הסכמה – במקרה הזה תקבלו Bid שמציין זאת. ניתן לבדוק זאת ע"י: <div style="text-align: center;"> LastAction instanceof Accept LastAction instanceof EndNegotiation </div> תוכלו במקרה זה לבדוק מהו ה-Bid שעליו הסכמתם. כמו כן, ייתכן שמו"מ יסתיים עקב Timeout (סיום הזמן המוקצה למו"מ) או אחרי כישלון טכני. במקרה זה, הקלט שתקבלו יהיה null ועליכם להתייחס אליו כאל כשלון במו"מ.
void ReceiveMessage(Action opponentAction)	קבלת הודעה במהלך המו"מ שמציינת את ה-Bid שנתן הצד השני.
Action chooseAction()	עליכם להחזיר את הפעולה שברצונכם לבצע. הפעולה יכולה להיות מתן Bid חלופי (Offer), הסכמה ל-Bid האחרון שנשלח לכם (Accept) או החלטה לסיים את המו"מ (EndNegotiation).
String getName()	עליכם להחזיר את שם הסוכן שלכם
static String getVersion()	מס' הגרסה של הסוכן שלכם (לשימושכם האישי, אין צורך לשנות)

7. הקפידו לשנות את שם קובץ ה-java לשם הסוכן שלכם (לדוגמה: AgentName.java). ניתן לצרף קבצים נוספים במידת הצורך.

8. הרצת התוכנה תתבצע ע"י כתיבה במסך ה-Command:
Java -jar Genius33.jar <number of sessions>
במידה ולא יכתבו כמות הסיבובים, ברירת המחדל היא 1.

9. טיפים, המלצות והערות:
- לסוכן אסור לקרוא או לכתוב למקור חיצוני (קובץ בדיסק, אינטרנט וכו').
 - על אף שבתוכנת ה-Genius אתם יכולים לראות את ה-Utility של כל הצעה עבור הסוכן השני, זכרו שבפועל בזמן המו"מ, אינכם יכולים לדעת את ה-Utility שלו! אתם יכולים רק לדעת עד כמה טוב ה-Bid שלו עבורכם.
 - חקרו את המשתנים שהסוכן שלכם מכיל. לדוגמה, UtilitySpace יכול להכיל המון מידע מועיל. עוד על כך בקובץ השני.
 - בחלק מהסביבות יש DiscountFactor – ככל שהוא יותר נמוך, כך ה-Utility של כל Bid יקטן לאורך זמן – כלומר, ה-Utility של ההצעה שהצעתם בזמן מאוחר יותר יהיה נמוך מה-Utility של אותה הצעה בדיוק בזמן מוקדם יותר (אולי כדאי לסגור על עסקה מוקדם יותר?). שימו לב שה-DF יכול להיות שונה לכל צד (אתם לא יודעים את ה-DF של השני)
 - דאגו לאתחל את כל משתני המחלקה שלכם בתוך שיטות (כמו init ו-StartSession) ולא בהצהרה החיצונית עליהם. שימו לב שאין לכם בנאי לסוכן שלכם – init הוא הבנאי מבחינתכם.
 - קימפול סוכן – אין צורך לקמפל אותו כפי שהוסבר בקובץ השני. אם הוא רץ בסביבת ה-Eclipse ביחד עם המערכת, זה מספיק טוב. יחד עם זאת, שימו לב – הקוד יתקמפל ב-JRE 1.6 ולכן יש להקפיד **להימנע** מכתיבה בסביבה 1.7.
 - כדי שהסוכן שלכם יופיע ברשימת הסוכנים בתוכנה, יש להכניס את השם שלו ואת המחלקה שאליה יש לגשת בקובץ agentrepository.xml.

- לצורך Debug, אם תרצו לטעון את הסביבה ל-Eclipse בוודאי תראו שקיימות הרבה מחלקות המכילות את השיטה `void main(String[] args)`.
- כדי להריץ את התוכנית, עליכם לקרוא לשיטה שנמצאת במחלקה `negotiator.genius`.
- לרשותכם קובץ של סוכן פשוט (נמצא תחת `samples\simpleAgent.java`), אשר מציע הצעות באופן רנדומלי. כמובן שסוכן כזה לא יצליח (בלשון המעטה) בעבודה..
- כמו כן, מצורפים 3 קבצי XML של סביבה + תפקידים במו"מ, הכוללים גם Reservation value ו-Discount factor, כדי שתתרשמו מאופן הכנת הקובץ.
- לנוחיותכם, עקב שאלות עבר, מצורף קטע קוד שתפקידו לעבור על כל הנושאים (Issues) בסביבה (Domain), ולבדוק עבור כל נושא מה הערך (Utility) של הסוכן עבור כל אפשרות שקיימת בנושא הזה. אני מאמין שחלקים מהקוד יסייעו לכם בכתיבת הסוכן, וכן להתמצא יותר במחלקות ובשיטות הרלוונטיות לכם בתוכנה:

```

Iterator<Entry<Objective, Evaluator>> issue = utilitySpace.getEvaluators().iterator();
while (issue.hasNext()){ // Iterate over every issue
    Entry<Objective, Evaluator> entry = issue.next();
    Evaluator e = entry.getValue(); // set evaluator for this issue.
    if (e.getType() == EVALUATOR.TYPE.DISCRETE){ //the same can be done for Integer and Real types
        Iterator<ValueDiscrete> v = ((EvaluatorDiscrete)e).getValues().iterator();
        while (v.hasNext()){ // Iterate over every option in this issue
            ValueDiscrete vd = v.next();
            double val = ((EvaluatorDiscrete)e).getEvaluation(vd);
            //this is the value of this option vd of issue e
        } //while
    } //if
} //while

```

בהצלחה!