



שפת ג'אוה ומבוא לפיתוח ווב

אביב 2020 - Super Duper Market (SDM)

תקציר

צרכנות נבונה כדרך חיים !

התרגיל מנוסח בלשון זכר, אך מכון לכל המגדרים האפשריים בצורה שווה

מרצה: **אביעד כהן** | aviadco@mta.ac.il

תוכן העניינים

4.....	דרישות הקורס
4.....	כללי
5.....	איך להגיש תרגילים באיחור, ולהישאר בחיים
6.....	הנחיות כלליות לכתובת התרגיל
8.....	תרגיל reflection – תרגיל עצמאי ביחידים (5%). מועד הגשה: 31.7.2020
8.....	כללי
8.....	מבנה התרגיל
8.....	ניקוד
9.....	איך בודקים ?
9.....	מה מגישים ?
9.....	מקרה בדיקה לדוגמא
11.....	Super Duper Market – מע' צרכנית נבונה
11.....	מטרת התרגיל(ים) בקורס
11.....	כללי
11.....	מיקומים
11.....	חנויות ומוצרים
12.....	לקוחות וביצוע קנייה
12.....	הנחיות ספציפיות למימוש מערכת Super Duper Market
13.....	תרגיל 1 – מימוש Super-Duper-Market כאפליקציית Console (25%) - הגשה: 22.8.2020
13.....	פרטים יבשים
13.....	דרישות
17.....	חלוקה למודולים
18.....	איך מתחילים ? (המלצה...)
18.....	בונוסים
18.....	סבבה, סיימתי. מה ואיך להגיש ?
19.....	שאלות ותשובות
20.....	תרגיל 2 – מימוש Super-Duper-Market כאפליקציית JavaFX (30%) - הגשה: 24.9.2020
20.....	פרטים יבשים
20.....	דרישות
24.....	חלוקה למודולים
25.....	איך מתחילים ? (המלצה...)
25.....	בונוסים
26.....	סבבה, סיימתי. מה ואיך להגיש ?
26.....	שאלות ותשובות
27.....	תרגיל 3 – מימוש Super-Duper-Market כאפליקציית WEB (40%) - הגשה: 31.10.2020
27.....	פרטים יבשים
27.....	דרישות
32.....	חלוקה למודולים

32	איך מתחילים ?
33	בונסים
34	סבבה, סיימתי. מה ואיך להגיש ?
34	שאלות ותשובות
35	נספח א' – תיאור מבנה המערכת באמצעות קובץ XML
35	סכמת קובץ XML – גרסה I
36	סכמת קובץ XML – גרסה II
38	סכמת קובץ XML – גרסה III
39	נספח ב' – קבצי סכמה לזיהוי קובץ ה XML

כללי

1. בקורס אין בחינה אך חובה להגיש תרגילים (סה"כ 4).
2. את מרבית התרגילים (3) ניתן להגיש בזוגות, אך לא בשלישיות / רביעיות / חמישיות או יותר (כן, גם אם מדובר בשלישייה / רביעיה / חמישייה הצועדת יחדיו לאורך שנים מאז גיל הגן והגישה עד עכשיו את כל הפרוייקטים ביחד).
את תרגיל ה- [reflection](#) חובה להגיש ביחידים.
3. בעבודה משותפת על תרגיל יש להקפיד על מעורבות אקטיבית של כלל המגישים בכל חלקי התרגיל.
4. במידה והוגדר בנוס לתרגיל מסוים, ציון הבונוס יתווסף לציון התרגיל בלבד (ולא לציון הסופי של הקורס כולו).
5. התרגילים יוגשו דרך מערכת Mama. מוגדר רכיב 'מטלה' נפרד לכל תרגיל.
6. לפני שליחת התרגיל יש לבדוק שהוא עובד ומכיל את הקבצים המעודכנים ביותר, על מערכת "נקייה".
בצעו את סט הפעולות שאתם מצפים מן הבודק לבצע וודאו כי הכל מתנהל כראוי וכסדרו.
7. ניתן להחליף את השותפ/ה בכל תרגיל, ללא צורך באישור או הודעה למרצה.
הניקוד על כל תרגיל נזקף לזכות הסטודנטים שבצעו אותו בלבד.
8. טרם הגשת התרגיל המתגלגל הראשון, תידרשו לשלוח מייל שבו מידע על צוות מגיש/י התרגיל.
במידה ויחול שינוי בציוות בתרגילים הבאים – יש לשלוח עדכון מתאים במייל לקראת ההגשה של התרגיל הבא.
9. יש להעלות את התרגיל רק עבור אחד מבני הזוג ולהוסיף את שם בת/בן הזוג ומספר תעודת הזהות שלה/ו גם באתר וגם בקובץ ה readme המצורף (פרטים בהמשך).

1. ניתן להגיש תרגיל עד שבוע איחור, כאשר עבור כל 24 שעות איחור – תורד נקודה אחת מציון התרגיל; תרגיל שיוגש באיחור של יותר משבוע (ללא סיבה מוצדקת) – **פשוט לא יבדק**.
2. במידה והגשתם תרגיל אולם הבודק נתקל בבדיקתו במצב שפשוט לא מאפשר את המשך הבדיקה (למשל כישלון בטעינת קובץ הבדיקה) – הרי שאתם מוגדרים כתקלת level 0. במקרה של תקלה שכזו איידע אתכם ואפשר לכם לבדוק, לתקן ולהגיש מחדש את התרגיל כדי שאפשר יהיה לבדוקו אחרי הכל.
שימו לב כי **בכל במקרה של הגשה חוזרת** בגלל level 0 – הציון לתרגיל יתחיל מ 90, ללא שום קשר לאופי הבעיה ו/או התיקון (גם אם התיקון היה "קטן". גם אם התיקון היה בגלל בלבול בהגשה של גרסה קדומה יותר של הקבצים. גם אם הכלב אכל לכם את שיעורי הבית)
3. בתרגילים השונים ניתן לממש בונוסים (פרטים בהמשך).
המטרה של הבונוס היא לעזור לכם להעלות את הציון ולא להורידו!
רוצה לאמר: אל תגישו באיחור רק בשביל להספיק לפתח בונוס.
בונוס מפתחים **אם ורק אם** סיימתם את כל דרישות הבסיס להגשה, יש לכם עוד מספר ימים, וברצונכם לנסות ולהגדיל את הציון ע"י בונוס.
היות וכך, ולמען הסר כל ספק: לא ייבדקו הבונוסים עבור תרגילים שהוגשו באיחור (שאינו מוצדק).
בהתאם לכך, אני שומר לעצמי חירות רבה יותר בשינוי כזה או אחר של מי מסעיפי הבונוס, גם במהלך התרגיל עצמו.
4. עומס בלימודים, בעבודה, בחיים, בגלל הילדים או ההורים, שכנים וחברים (או בכל תחום אחר) **אינו נחשב** כסיבה לגיטימית להארכה.
5. במקרה של מחלה או מילואים יש להוסיף סריקה של אישור המחלה או המילואים להגשת התרגיל; בהגשת תרגיל באיחור בשל שירות מילואים יש לצרף להגשת התרגיל קובץ סרוק של **טופס השחרור מהמילואים** (ולא את צו הקריאה).
6. במקרה של בקשה להארכה (מכל סיבה שהיא, לרבות מילואים ומחלה) יש לפנות למרצה מראש על מנת לקבל אישור (במייל או פורנטלית בכיתה)
7. לאנשי הקבע – הישארות של שבת בבסיס אינה נחשבת כסיבה לגיטימית להארכה (מכיוון שזה חלק מהסדר העבודה בצה"ל); יציאה לאבט"ש כן נחשבת כמילואים ויש להגיש אישור ממפקד הבסיס.
8. סטודנט המגיש תרגיל באיחור של **עד שבוע** עם סיבה מוצדקת ישלח את הנימוקים לאיחור ביחד עם התרגיל.
9. ניתן לערער על ציון של תרגיל לכל היותר שבוע מיום פרסום המשוב והציון במע' המאמא.
כדי לערער יש לשלוח מייל למרצה בצירוף כל הסיבות והטענות שלכם.

- במהלך הקורס יוצגו דוגמאות והסברים מבוססים על כתיבה בסביבת הפיתוח (IDE) – IntelliJ IDEA. אתם מוזמנים (ומעודדים בזאת) לפתח גם כן את התרגיל בסביבת העבודה intellij. ניתן לקבל רישיון חינם לשימוש בגרסת ultimate, רק בשל היותכם סטודנטים במכללה (כבר שווה!) יחד עם זאת, כל אחד רשאי לבחור לעבוד בסביבת העבודה הנוחה והמוכרת לו. כך או אחרת הגשת התרגיל אינה כוללת את סביבת הפיתוח אלא אך ורק הרצה ידנית מ cmd (כמו פעם...). שימו לב: מבחינתכם, לבודק פשוט אין intellij (או כל ide אחר) ולכן זו אפילו לא אופציה. חיסכו ממני (ומכם) את כתיבת המייל המבקש זאת.
- יש להגיש את התרגילים בתור קובץ rar/zip (לא 7Z!) הקובץ יכיל:
 1. כל הקבצים הרלבנטיים להפעלת התרגיל (war/jar – פרטים בגוף התרגיל).
 2. קובץ אצווה (batch ==) שיכיל את הפקודה שמריצה את התרגיל.
 3. קובץ readme שיכיל את פרטי המגיש/ים, כמו גם הנחיות כלליות להרצה התרגיל וכל הנחות שלקחתם במהלך התרגיל ואתם סבורים שחשוב כי הבודק יכיר. דמיינו כי בכל שאלה/תקלה שיתקל בהן הבודק, יעמוד לרשותו רק קובץ ה readme שלכם. דאגו להבהיר ולהסביר את כל הדברים שיכולים להשתבש ו/או שבעטיים ייתכנו בעיות/שאלות/תהיות וכיוצא בזה.
 כמו כן, כל הנחה שאתם מניחים בעצמכם לגבי אופן מימוש התרגיל (בין אם בלוגיקת התרגיל ובין אם בהנחה טכנולוגית) צריכה להיות רשומה בקובץ. על קובץ ה readme להיות בפורמט word או pdf (לא notepad!). חי נפשי – אם מישהו מגיש readme כקובץ טקסט פשוט תרד לו נקודה...
- דווקא בגלל שאין זהו קורס שבו יכנסו לנבכי הקוד ויבדקו כל שורה ושורה, יש להקפיד ביתר שאת על קוד נקי, מסודר, קריא ויעיל. בפרט:
 - הימנעו משכפול קוד
 - פונקציות ארוכות מדי (יותר מגודל עמוד)
 - בחירת שמות גרועים למחלקות, לפונקציות ולמשתנים
 - הזחה (אינדנטציה) נכונה
 - imports מיותרים
 - יש להקפיד להשתמש ב-modifiers בצורה נבונה:
 - מחלקה שלא אמורים לבנות אובייקטים שלה אמורה להיות מוגדרת כ-abstract
 - קבועים יש לסמן כ-final
 - משתנים של המחלקה רצוי להגדיר כ-private אלא אם יש סיבה לגיטימית לבחירה אחרת.
 - יש להקפיד על מוסכמות בסגנון הכתיבה – שמות מחלקות יתחילו באות גדולה, שמות חבילות, משתנים ופונקציות באות קטנה, שמות קבועים יהיו מורכבים רק מאותיות גדולות וכו'. ראו מסמך java coding conventions שהועלה למאמא.
- התמודדות עם קלט שאינו תקין (במקומות הרלבנטיים) היא חלק בלתי נפרד מחווית המפתח (לטוב ולרע...). יש לוודא קלט תקין מהמשתמש בכל שלב ולהחזיר הודאות שגיהא קריאות, אינפורמטיביות במידה והקלט אינו תקין. (למשל: לא להגיד שהקובץ לא תקין – אלא מה לא תקין בקובץ בצורה מפורטת...)

- כל הקלט והפלט בתרגילים השונים יהיה באנגלית בלבד.
- אין להציג או לתמוך בקבלת קלט ו/או הצגת פלט בעברית או בכל שפה אחרת.
- כל הקלטים באנגלית יהיו case insensitive, כלומר אין חשיבות ל capital case. דוגמא: MoMo=mOmO
- יש לדאוג שבאף מקרה לא ייזרק Exception שלא טופל ושיגרום לתעופה של התוכנית; יש לטפל ב-Exceptions בנקודה שבה ניתן לעשות כן.
- הוראות שגויות שייגרמו לאפליקציה שלא לרוץ יורידו נקודות, ולכן רצוי מאוד שתנסו להתקין את האפליקציה בעצמכם לפי ההוראות שתכתבו.
- זהו תרגיל מתגלגל. המטרה היא לבנות בסיס ראשוני בתרגיל הראשון, ולהמשיך ולהשתמש בו, ככל האפשר (ואפשר!) במהלך התרגילים הבאים. השקיעו חשיבה ותכנון בעיצוב הפתרון תוך מחשבה על איך **מה שתעשו היום ישרת אתכם מחר**. (זה כלל נכון לחיים, לא רק לתרגיל זה).
- חלק מהעבודה בתרגילים היא קבלת החלטות בנושאים שאינם מפורטים במדויק. המטרה היא לתרגל את הנושאים המרכזיים הנלמדים בקורס, ולא לתפור מוצר לפי דרישות של לקוח. על כן, בכל מקום שלא מופיעה דרישה מדויקת – מוטל עליכם לבחור בדרך ההגייונית ביותר שנראית לכם ולציין את בחירתכם בקובץ ה Readme אשר מוגש עם התרגיל. אם יש ספק לגבי אופן פעולתכם אתם מעודדים לשאול האם הפתרון שאתם חושבים לתת לסוגיה מסוימת הוא קביל ולגיטימי (שאלות בפורום, מייל למרצה וכו)
- ווידאו הגשת התרגיל טרם הגשתו:
 - יש לוודא כי ההגשה שלכם רצה היטיב על מע' נקייה, באופן שבו גם הבודק יריץ אותה, על מערכת נקייה וללא תוצרי לוואי אחרים של הפעלות קודמות שלכם.
 - הבודק יבצע את הבדיקה על מע' windows 10. כל מי שמפתח על גבי linux/mac – זכותכם – אבל גם חובתכם לוודא כי אתם רצים היטיב על windows 10. למען הסר ספק, לא תתבצע בדיקה על מע' הפעלה אחרת. כמו כן לא תהיה התחשבות בתקלות שמקורם רק בשל עבודה על מע' הפעלה שונות (ולא שאמורות להיות תקלות כאלה..)
 - יש לוודא כי כל קבצי הבדיקה השונים שהועלו ל mama נטענים בהצלחה ע"י המע' שלכם טרם ההגשה. בדיקת הבודק תתחיל מבדיקה בסיסית המבוססת בצורה גסה על קבצים אלה. חבל ליפול 0 Level על שטות שיכולתם לעלות עליה בשנייה עוד בשלב הפיתוח.
 - כאמור, הגשה חוזרת בשל תקלות level 0 תתחיל מראש מציון של 90. בלי שום יוצא מן הכלל. הקדימו תרופה למכה.
- בחלק מהתרגילים ניתנת אפשרות למימוש דרישות בונים.
 - ישנם 2 סוגי בונים:
 1. בונים בתוך טווח התרגיל - יכול להביא אתכם לכל היותר לציון 100, ולחפות במקרה והורדו לכם נקודות בשל תקלות.
 2. בונים מחוץ לטווח התרגיל – יכול להעלות את ציונכם אף מעבר ל 100 (וכן, יש כפל מבצעים לטובת הלקוח).
 - בכל מקרה יש לבצע את הבונים **אם ורק אם** סיימתם את כל דרישות הבסיס ההכרחיות לתרגיל.
 - חלק מהבונים בתרגילים השונים הם כאלה שנועדו "להקדים תרופה למכה" – מימוש דרישה בתרגיל n אשר בכל מקרה תגיע כדרישה חובה בתרגיל n+1.
 - הדבר נועד לעודד אתכם להוריד את העומס הצפוי בתרגיל n+1, מתוך הנחת יסוד שתרגיל n הוא קל יותר ומרווח יותר. תכננו את עבודתכם בהתאם ושאלו "להקדים תרופה למכה", במידת האפשר. (וגם זה כלל חשוב לחיים, בלי קשר לתרגיל ולקורס).
 - פירוט הבונים, משקלם ונקודותיהם מפורט בגוף התרגיל הספציפי.
 - אם כבר מממשים בונים, יש לממש את כולו, עפ"י דרישתו כדי לזכות במלוא הניקוד שהוא מקנה. בכל מקרה ההחלטה על ניקוד הבונים היא בידי הבודק/מרצה בלבד (אל דאגה, המגמה היא להיות נדיבים ככל האפשר...)
 - כאמור, ולמען הסר כל ספק – ניקוד הבונים מתווסף לניקוד התרגיל הספציפי שבו הוא מומש ולא לניקוד הסופי של הקורס. לא ניתן לקבל ציון סופי בקורס שהוא מעל ל 100 (גם אם בזכות הבונים הגעתם לציון כזה).

כללי

מועד הגשה: 31.7.2020

צורת הגשה: ביחידים בלבד

בתרגיל זה תזכו להכיר ולתרגל את מנגנון ה Reflection בג'אווה.

המצגת מכילה ידע בסיסי המהווה נק' פתיחה בלבד לנושא זה, ולכן כחלק מהתרגיל תדרשו גם ללמוד לבד נושאים נוספים הקשורים לעולם ה reflection שייתכן ואינם מכוסים (או אינם מכוסים כהלכה) במצגת. שימו לב כי במצגת ישנם 3 נושאים. התרגיל הוא על הנושא הראשון בלבד של Reflection.

מהות התרגיל היא לכתוב investigator שיודע לקבל מופע (instance) של איזה שהוא אובייקט, ואז יודע "לחקור" אותו ולענות על מספר שאלות בהקשרו.

במהלך התרגיל תצטרכו להשתמש אך ורק ביכולת ה Reflection של ג'אווה כפי שמוסברות במצגת. **אין להשתמש (ואין שום צורך) בשום ספריית צד שלישית כדי לבצע את המטלות השונות!**

צפי העבודה על התרגיל, בהינתן שקראתם והבנתם את המצגת, הוא 3-4 שעות. צפי אורך הקוד שעליכם לכתוב הוא לא יותר מ 250 שורות (אני עשיתי זאת ב 160 שורות מרווחות היטב..).

הבדיקה לתרגיל תבוצע בצורה אוטומטית, ע"י קוד שיטען את הקובץ שלכם ויריץ את סט הבדיקות שנגזרות ממנו, כלומר יקרא לכל השיטות המוגדרות ב interface שמימשתם, תוך השוואת הערך המוחזר מהקריאה אל ערך מצופה.

מבנה התרגיל

התרגיל מכיל ממשק בשם Investigator, אותו עליכם לממש. מהות השיטות בממשק היא לחקור instance של class אחר.

כל שיטה בממשק מתועדת היטב מבחינת מה היא צריכה לעשות, מה הפרמטרים שהיא מקבלת ומה היא צריכה להחזיר. חלק מהתרגיל כולל גם התמודדות עם התייעוד והבנה בעזרתו בלבד מה עליכם לעשות בכל שיטה ושיטה.

הערות חשובות:

1. שימו לב כי ה class שאתם מממשים חייב להכיל default public constructor !
2. שימו לב כי עליכם למקם את הממשק שקיבלתם (Investigator.java) בדיוק תחת ה package שנקרא reflection.api (ובהתאם לכך גם היררכיית הספריות כמובן).
3. במידה ויש מטודות הדורשות תפיסת exception, יש לתפוס אותו אולם אין להדפיסו! (זה יוצר אי-סדר בהדפסות הפלט). במידה ומתרחשת תקלה או שתזרקו הלאה את ה exception ותוכנית הבדיקה תתמודד איתו בדרכה (הורדת ניקוד על הסעיף המדובר) או שתחזירו ערך כלשהוא כתוצאה ממטודה (גם אם הוא לא נכון).
4. התוכנית מתחילה מקריאה לפונקציה load (אחת הפונקציות המתוארות בממשק). בפונקציה זו תקבלו את ה instance אותו עליכם לחקור. אפשר להניח כי זו הפונקציה הראשונה שתקרא, וכי היא תיקרא בדיוק פעם אחת.
5. כל מטודה בתרגיל עובדת בפני עצמה ופועלת על ה instance שקיבלתם במטודה load. אין להניח או להסתמך על סדר בקריאת המטודות (למען האמת בכל בדיקה הסדר הוא רנדומלי).
6. אין שום צורך (מבחינת התרגיל) לכתוב פונקציית main בקובץ התוכנית שלכם. אני לא אפעיל את התוכנית שלכם, אלא אפעיל את תוכנית הבדיקה שלי (שם יש main) והיא, בתורה, תטען את התוכנית שלכם.

ניקוד

התרגיל שווה עד 5 נקודות **מהציון הסופי** (!!)

(לא רע ל 3 - 4 שעות עבודה ו 200 שורות קוד...)

איך בודקים ?

קיבלתם את תוכנית הבדיקה עצמה, אותה הבודק הולך להפעיל לטובת בדיקת התרגיל.

התוכנית מקבלת כפרמטר חיצוני את שם הקובץ המקומפל שלכם (.class) ובודקת אותו על מספר מקרי בדיקה.

בעותק התוכנית שקיבלתם כרגע, יש דוגמא למקרה בדיקה פשוט עליו תוכלו לנסות, להתנסות ולוודא אם אתם בכיוון הנכון או אם לאו.

כדי להפעיל את תוכנית הבדיקה עליכם לנווט לספרייה בה נמצאת תוכנית הבדיקה, ולהקליד ב CMD:

```
...\> RunTester <your .class file name>
```

אם הכל עובד כשורה, תקבלו את הפלט הבא:

Test name: Rectangle Class

Testing Rectangle Class basics...

Testing [getTotalNumberOfMethods]: expecting answer [6] and got [6]

...

Test Score: 100

מה מגישים ?

עליכם להגיש קובץ zip, הכולל **בדיוק** 2 קבצים (ו 2 קבצים בלבד !):

1. קובץ התוכנית שלכם בלבד, בגרסתו המקומפלת (.class).

2. קובץ קוד המקור (למקרה של בעיות חמורות בלבד)

שם קובץ הזיפ צריך להכיל את שמכם ואת הת.ז. (אין להגיש קובץ readme בתרגיל זה...)

מה לא מגישים ?

1. את קובץ הממשק שקיבלתם

2. תיקיית פרויקט...

3. כל זבל אחר שבמקרה יושב לכם ליד הקוד...

מקרה בדיקה לדוגמא

במקרה זה מתואר האובייקט **Rectangle** היורש מאובייקט שנקרא **Polygon**.

מימוש זה כבר מוטמע בתוכנית הבדיקה שקיבלתם ומופע שלו יינתן כקלט לתוכנית שלכם.

המופע יאותחל בצורה הבאה:

```
rectangle = new Rectangle(4,6);
```

(המימוש הוא חלקי ולא תמיד הגיוני – אז אל תתפסו לקטנות...)

```
public class Polygon {  
    private Set<Point> points;  
  
    public Polygon() {  
        points = new HashSet<>();  
    }  
  
    public int getTotalPoints() {  
        return points.size();  
    }  
  
    protected void addPoint(int x, int y) {  
        points.add(new Point(x, y));  
    }  
}
```

```

    }
}

public class Rectangle extends Polygon implements Comparable, Serializable {

    private int x;
    private int y;
    private final int SCALE = 2;

    public static void PRINT_SOMETHING() {
        System.out.println("this is a static method");
    }

    public Rectangle() {
        x = -1;
        y = -1;
    }

    public Rectangle(int x, int y) {
        this.x = x;
        this.y = y;
        updateParent();
    }

    private void updateParent() {
        addPoint(0, 0);
        addPoint(x, 0);
        addPoint(0, y);
        addPoint(x, y);
    }

    public int calcArea() {
        return x * y;
    }

    public int calcPerimeter() {
        return twice(x) + twice(y);
    }

    private int twice(int num) {
        return 2 * num;
    }

    @Override
    public int compareTo(Object o) {
        return this.calcArea() - ((Rectangle)o).calcArea();
    }
}

```

מטרת התרגיל(ים) בקורס

- חלק א' – מימוש מנוע מע' הקניות המציע פוקציונליות בסיסית ותפעול המע' כאפליקציית console.
- חלק ב' – הרחבת המע' במספר פיצ'רים ופיתוח תצוגה גרפית כאפליקציית Java FX (Desktop).
- חלק ג' – מימוש אפליקציית web המציעה מע' מקוונת המעודדת תחרות ומאפשרת צרכנות נבונה.

כללי

(הדרישות "הכלליות" המובאות בהמשך תקפות לכל תיאור המע' בכללותה. בכל תרגיל נממש רק (עוד) חלק מהדרישות.)

יוקר המחיה מהווה משקולת כלכלית מהותית על יותר ויותר אנשים בעולם. אחד האמצעים להקלה ביוקר המחיה הוא היכולת לבדוק ולהשוות מחירים בצורה נוחה וקלה, דבר המאפשר קנייה צרכנית נבונה של סל מוצרים כלשהוא.

התפתחות עולם הטכנולוגיה מאפשרת לקרב ולהפגיש בין מוכרים (חנויות) לקונים ולייצר זירה דיגיטלית בה יכול הצרכן, מחד, לבצע השוואת מחירים קלה ומהירה, ומאידך, לייצר חשיפה עבור חנויות לצרכנים שונים שקודם לכן לא היו מודעים להן, כמו גם לעודד תחרות בין החנויות על ליבו של הצרכן.

בתרגיל זה תפתחו מע' אשר תאפשר להגדיר חנויות המוכרות פריטים שונים, ותאפשר לצרכן לבצע הזמנה של פריטים אלה, תוך שהמע' מבצעת את השוואת המחירים האידיאלית לפי דרישות הצרכן.

מיקומים

המע' תתמוך במע' קורדינטות דו מימדית. המע' מתחילה מנקודה 1 ; 1 הממוקמת בצד שמאל למטה (כראשית הצירים). לכל בר-מיקום במע' (חנות, לקוח) יש לציין את מיקומו באמצעות קורדינטת ה X וה Y שלו. הקורדינטות (Y ו X) ימצאו בין הטווח של 1 ו 50, כולל: [1,50]

בהינתן 2 מיקומים במע' (כלומר 2 זוגות קורדינטות), ניתן לחשב ביתר קלות באמצעות משפט פיתגורס את המרחק ביניהן.

דוגמא:

מיקום 1: x = 2 ; y = 4 < -- > מיקום 2: x = 5 ; y = 8

מרחק:

$$D = \text{SQRT}((|X1-X2|)^2 + (|Y1-Y2|)^2)$$

חנויות ומוצרים

- המע' תאפשר להגדיר מוצרים שונים שנמכרים על ידי החנויות השונות.
- עבור כל מוצר/פריט יש לשמור את המידעים הבאים:
- מספר סידורי ייחודי המתאר אותו בצורה חד ערכית. מספר שלם.
 - שם (באנגלית) המתאר את המוצר. מחרוזת תווים. יכולה להכיל רווחים.
 - אופן הקנייה: האם זהו מוצר שקונים לפי כמות (למשל קטשופ) או לפי משקל (למשל פטרוזיליה)

בנוסף, המע' תאפשר להגדיר 'חנות', דרכה יכולים למכור את כל/חלק מהמוצרים השונים.

כל חנות תתמחר כרצונה את המחיר שהיא מעניקה לכל מוצר שהיא מוכרת.

אם אופן קניית המוצר הוא לפי כמות (למשל קטשופ) – המחיר הוא עבור יחידה אחת של המוצר.

אם אופן קניית המוצר הוא לפי משקל, אזי המחיר הנקוב הוא עבור 1 ק"ג של המוצר.

כל חנות תמוקם בקורדינטה במע'.
לא ייתכנו 2 חנויות הממוקמות באותן הקורדינטות.

כל חנות תציע מחיר לביצוע משלוח של הזמנה ממנה.
עלות המשלוח נקבעת לפי המרחק שבין הלקוח לחנות. החנות תנקוב בסכום עבור כל ק"מ מרחק. יחידה זו נקראת PPK – Price Per Kilometer.
כך, אם בחנות מסויימת $PPK = 20$ והלקוח מרוחק ממנה כדי 2 ק"מ – אזי עלות המשלוח היא 40.

כל חנות יכולה להציע "מבצע" על מוצר(ים) מסויימים.
כל מבצע מורכב משני חלקים:
1. אם קונים מוצר מסויים (יש לספק מספר סידורי של מוצר) בכמות מסויימת (יש לספק את הכמות) אז
2. אפשר לקבל מוצר(ים) אחרים (יש לתת מספרם הסידורי של המוצר(ים)), בכמות מסויימת (יש לציין את הכמות) בתוספת תשלום כזו או אחרת עבור כל יחידה.

לקוחות וביצוע קנייה

לכל לקוח במע' יהיה מספר סידורי מזהה חד חד ערכי ושם (באנגלית, היכול להכיל רווחים)

הלקוחות במע' יכולים לייצר הזמנה, המכילה את הפריטים (מתוך המלאי הכללי שהמע' מציעה) שהם מעוניינים לרכוש ובכמות מסויימת. כחלק מביצוע ההזמנה הלקוח יגדיר את מיקומו על גבי רשת הקורדינטות, וכך אפשר יהיה לחשב את עלות המשלוח של המוצרים אליו בהתאם למיקום של החנות(יות) מהן ההזמנה שלו תתבצע. לקוח (או הזמנה) לא יופיעו על קורדינטה של חנות ולהיפך. לכל לקוח/חנות יש את הקורדינטה הייחודית לו.

כחלק מהגדרת ההזמנה, הלקוח יבחר האם הוא רוצה לקנות מחנות מסויימת את כל סל הקניות שלו (בחירה סטטית, ידועה מראש) או לתת למע' להרכיב בשבילו את ההזמנה האופטימלית, בצורה דינמית, כך שכל פריט ייקנה מהחנות המציעה את המחיר הזול ביותר ובכך למעשה אפשר יהיה להוזיל את סל המוצרים של הלקוח.

בזמן ביצוע ההזמנה, במועד כזה או אחר (ראו פירוט בהמשך התרגיל) המע' תדע להציע ללקוח את המבצעים האפשריים העומדים לרשותו בהתאם לחנות(יות) מהן הוא בוחר לקנות ולהרכיב המוצרים שבחר. במידה והלקוח יתעניין – הוא יוכל לבחור לאמץ את אחד המבצעים (או יותר) ובכך להנות ממוצרים נוספים במחירים אטרקטיביים יותר.

למחיר הסופי של ההזמנה יתווסף מחיר המשלוח(ים) לפי חישוב המרחק של הלקוח מהחנות(יות) ממנה מתבצעת ההזמנה.

כל לקוח יוכל גם להעניק פידבק לחנות(יות) מהן הוא ביצע את ההזמנה. הפידבק יכול ערך מספרי שלם המעיד על שביעות הרצון (מספר בין 1 ל 5) כמו גם, אופציונלית, פידבק מילולי המעיד על חווית הקנייה.

הנחיות ספציפיות למימוש מערכת Super Duper Market

- המטרה היא לבנות מנוע מערכת גנרי, כזה שידע לקבל את הפרטים לגבי מבנה החנויות, והמוצרים המגיעים מתוך קובץ נתונים בפורמט XML (עבודה עם XML'ים תילמד במהלך הקורס כמובן).
מנוע המערכת הגנרי ילך וישתכלל מתרגיל לתרגיל, בהתאם לפיצ'רים השונים. כך תוכלו לחוות מהלך שלם של מוצר החל מרעיון קטן במימוש בסיסי וכלה במנוע מע' המניע אפליקציית ווב שלמה.
- במהלך התרגיל תידרשו לכתוב אלגוריתם שיבוצ' הזמנות לחנויות. זהו לא קורס אלגוריתמים ואינכם נבחנים על איכות וטיב האלגוריתם. הוא לא חייב להיות היעיל או הנכון ביותר. הוא פשוט צריך לעבוד בזמן סביר ולתת תוצאות נכונות (גם אם במקרי קצה מסויימים לא תדייקו – זה בסדר). הקלטים לאלגוריתם (כמות המוצרים, החנויות, הלקוחות, פריטים בהזמנה) יהיו יחסית קטנים כך שגם אלגוריתם שאינו יעיל (מבחינת חישוב סיבוכיות) לא אמור להיתקע או להימשך יותר מדי זמן.

פרטים יבשים

תאריך הגשה: **22.8.20**

ציון אפשרי מקסימלי: 110
קושי: סביר

צפי תחילת עבודה: **31.7.20**

צפי זמן לביצוע: 3 שבועות
משקל התרגיל: 25%

מטרת התרגיל העיקרית

- הקמת מנוע המע' הבסיסי
- יצירת ממשק console לתפעול המערכת

דרישות

- בתרגיל זה תממשו את מנוע המערכת, אשר יודע לקבל נתונים על המוצרים והחנויות השונות, ויאפשר ללקוח בודד (המשתמש) לבצע הזמנות במע'. המערכת תכיל סט פיצ'רים מינימלי שיאפשרו תפעול בסיסי של המע'.
את מנוע המע' תלווה שכבת UI מסוג של console application אשר תאפשר למשתמש "לדבר" עם המע' באמצעות הצגת סדרה של תפריטים ופעולות אפשריות. המשתמש יבחר פעולה לביצוע, יכניס קלט נדרש (במקרים הנדרשים) ויקבל חזרה את הפלט מהמע' וחוזר חלילה.
- בתרגיל זה תתמכו בגרסה "קלת משקל" של המע' המציעה:
 - כל המוצרים והחנויות יוגדרו במלואם בקובץ ה XML. אין צורך לתמוך בהגדרה דינמית שלהם.
 - אין "לקוחות" בתרגיל זה. כל הלקוחות הם לקוח אחד ויחיד שהוא המשתמש. אין לו שם או מספר סידורי.
 - בביצוע הזמנה המשתמש יבחר ידנית ומיידית את החנות ממנה הוא מעונין לרכוש את המוצרים (בחירה סטטית).
 - אין תמיכה ב"מבצעים"; אין תמיכה באלגוריתם לחישוב עלות סל אופטימלית; אין תמיכה במתן פידבקים.
- פרטיה הטכניים של המערכת יינתנו מקובץ XML (כמפורט [בנספח](#)).
עליכם לוודא בדיקת קלט לקובץ ה XML ולוודא כי הקובץ מכיל מידע תקין ואמין.
(מובטח כי הקובץ יהיה תקין schema-wise אבל לא בהכרח תקין application-wise...)
במידה והקובץ מתגלה כתקול, יש להודיע על כך בצורה מסודרת ונאותה למשתמש. יש לפרט מהי סיבת התקלה (ולא להסתפק במשפט סתום-בלום כמו: "הקובץ תקול...").

בפרט יש לוודא את הפרטים הבאים:
 - 3.1. הקובץ אכן קיים, והוא מסוג XML (די לבדוק לשם כך כי הוא נגמר בסימנת xml)
 - 3.2. בתוך קבוצת המוצרים, אין 2 מוצרים בעלי id זהה
 - 3.3. בתוך קבוצת החנויות, אין 2 חנויות בעלי id זהה
 - 3.4. בתוך חנות אחת, בהגדרת מכירת מוצר – ההפניות הן רק למוצרים קיימים
 - 3.5. כל מוצר נמכר לפחות ע"י חנות אחת
 - 3.6. בתוך חנות אחת, מכירת מוצר לא מוגדרת פעמיים
 - 3.7. מיקומי החנויות לא חורגים מרשת הקורדינטות של 50 ; 50 (x,y: [1,50])

4. יש לוודא תקינות קלט כחלק מכל אינטרקציה עם המשתמש בכל מקום שבו זה רלבנטי: אם אתם מצפים לקבל מספר – לא לקרוס כי הכניסו לכם בטעות (או בכוונה) טקסט וכו'. בכל מקרה של תקלה יש להיות מאוד ברורים במסר שמעבירים חזרה למשתמש: מה קרה? מה הייתה מהות התקלה? היכן שזה רלבנטי, איך לתקנה וכו'. חישבו איך להיות ידידותיים למשתמש ולעולם אל תניחו כי מי שמשתמש באפליקציה שלכם הוא מתכנת בעצמו או מישו שמגיע מהתחום ו"מכיר" איך דברים עובדים לבד. (זה הזמן לחשוב על...)

5. **אין צורך** להשתמש בצבעים שונים במהלך תרגיל זה בעת ההדפסה ל console. יתרה מזאת, ישנו צפי רב (ניסיון מהסמסטרים הקודמים) כי ניסיון לעשות כן תוך שימוש בספריות צד שלישי קורס אצל הבודק, מעוות את כל תצוגת המסך וגורם לחוסר יכולת לבדוק את ההגשה. גם אם בדקתם את זה אצלכם וזה עבד. גם אם בדקתם במחשב של השכנה וזה עבד.

כמו כן **אין צורך** לנקות את המסך בין פקודה לפקודה.

6. עליכם לכתוב ממשק משתמש בתצורת console. ממשק המשתמש יכול סט סופי של פקודות שדרך ניתן יהיה להפעיל את המערכת. אחרי הצגת תפריט הפקודות יש לחכות לקלט מהמשתמש באשר לפעולה אותה הוא רוצה לבצע. לאחר ביצוע הפעולה (שאוילי תגרור בקשת קלט נוסף מהמשתמש) יש להציג את הפלט החוזר ממנה (לכל פקודה יש פלט החוזר ממנה) ואז להציג שוב את התפריט וחוזר חלילה. **שימו לב:** ישנן פקודות שאין הגיון לבצע אותן אם לא קדמו להם פקודות אחרות. במידה וזה קורה יש להציג הודעת שגיאה רלבנטית למשתמש ולאפשר את המשך מהלך פעילות המע'.

רשימת הפקודות שיש לתמוך בהן:

1. קריאת קובץ פרטי המע'

פקודה זו טוענת את פרטי המערכת מתוך קובץ נתונים בפורמט XML. קבצי דוגמא מתאימים הועלו מבעוד מועד לאתר הקורס ואתם מוזמנים להורידם ולבחון אותם בהתאם. (אתם מעודדים לייצר לעצמכם קבצי בדיקה נוספים כדי לבדוק את המע' בצורה יסודית וטובה יותר כאוות נפשכם). יש לבקש מהמשתמש נתיב מלא לקובץ ה XML אותו הוא רוצה לטעון למע'. הנתיב יכול להכיל רווחים בתוכו (למשל "program files") ויש לוודא כי הדבר לא מכשיל אתכם (ולא שהוא אמור) הקובץ יהיה קובץ XML שפרטיו וחוקיו המפורטים מובאים ב**נוספה** לתרגיל זה. אתם מצופים לעבור על פרטים אלה ולהתייחס אליהם כחלק מהגדרת התרגיל. יש לוודא תקינות הנתונים המופיעים בקובץ כפי שהוגדרו בסעיף 3.

במידה והקובץ לא תקין יש לדווח זאת למשתמש בצורה ברורה כך שניתן יהיה להבין מה לא תקין בקובץ. אין לקרוס על exception במידה וקובץ מתגלה כאינו תקין; יש לאפשר למע' להמשיך לפעול במצב זה. (כחלק מבדיקת התרגיל יטענו למערכת קבצים שאינם חוקיים כדי לבדוק מהי התגובה).

במידה והקובץ נמצא תקין – יש לטעון את פרטיו למע' ולדווח על כך למשתמש.

דגשים:

1. יש לאפשר למשתמש לטעון כמה קבצים אחד אחרי השני (כלומר להפעיל את הפקודה כמה פעמים רצוף). כל קובץ תקין "דורס" לחלוטין את כל פרטי הקובץ (התקין) שהיה טעון לפניו במע' (ככל שהיה כזה). כל טעינה של קובץ תקול לא דורסת את פרטי הקובץ (התקין) האחרון שהיה במע' (ככל שהיה כזה)

2. פקודה זו מוצגת ומאפשרת תמיד. אפשר לבחור בה בכל רגע נתון במע'.

2. הצגת פירוט חנויות

פקודה זו תציג למשתמש את הפרטים השונים של כל חנות המוגדרת במע'.

עבור כל חנות יש להציג את המידע הבא:

1. מספר סידורי של החנות
2. שם החנות
3. רשימת הפריטים שהחנות מוכרת.
עבור כל פריט יש לכלול את המידעים הבאים:
 - a. מספר סידורי
 - b. שם הפריט
 - c. אופן הרכישה (לפי כמות או לפי משקל)
 - d. מחיר הפריט ליחידה אחת בחנות
 - e. סה"כ פריטים שנמכרו עד עכשיו מחנות זו
4. הזמנות שבוצעו מהחנות (ככל שיש כאלה. ייתכן ואין יש רק לציין זאת)
עבור כל הזמנה יש לכלול את המידעים הבאים:
 - a. תאריך
 - b. סה"כ פריטים (כמות כוללת, לא סוגים)
 - c. סה"כ עלות הפריטים
 - d. עלות המשלוח
 - e. עלות הזמנה כוללת (c + d)
5. PPK
6. סה"כ תשלום שקיבלה החנות רק על משלוחים עד עכשיו

דגשים:

אין לאפשר פקודה זו אם לא טעון קובץ תקין במע'.

3. הצגת כל הפריטים במע'

פקודה זו תציג למשתמש את כל הפריטים המוגדרים במע'

עבור כל פריט יש להציג:

1. מספרו הסידורי
2. שם הפריט
3. אופן קניית הפריט (לפי כמות או לפי משקל)
4. כמה חנויות מוכרות את הפריט
5. מחיר ממוצע של הפריט (כפונקציה של כל החנויות המוכרות אותו)
6. כמה פעמים סה"כ נמכר הפריט במע' (מסך ההזמנות שבוצעו) – כמות כוללת של הפריט על פני כל ההזמנות שבוצעו עד כה במע'. אם מדובר על פריט לפי המשקל – התוצאה יכולה להיות גם עשרונית.

דוגמא:

אם בהזמנה 1 נקנו 2.3 ק"ג בננות ובהזמנה 2 נקנו 1.4 ק"ג בננות, אזי בפירוט המוצר "בננות" יוצג סה"כ 3.7 ק"ג

אם בהזמנה 1 נקנו 3 קטשופים ובהזמנה 2 נקנו 1 קטשופ, אזי בפירוט המוצר "קטשופ" יוצג סה"כ 4

דגשים:

אין לאפשר פקודה זו אם לא טעון קובץ תקין במע'.

4. ביצוע הזמנה וקנייה במע'

פקודה זו מאפשרת לבצע הזמנה חדשה במע'.

היא כוללת כמה שלבים שיש להציגם למשתמש בצורה מדוייקת וללא סטיה:

1. יש להציג למשתמש את רשימת החנויות הפעילות במע'. עבור כל חנות יש להציג את מספרה הסידורי ; את שמה ; (PPK).

המשתמש יבחר מתוך הרשימה את החנות המבוקשת על ידי הכנסת מספרה הסידורי.

2. יש לבקש מהמשתמש את תאריך ההזמנה המבוקש. יש להכניס זאת בפורמט של dd/mm-hh:mm

3. יש לבקש מהמשתמש את מיקומו על גבי המפה. הקורדינטות הן של x ו y ויש לוודא כי הן בתחום המותר (בין 1 ל 50 כולל). שימו לב לקוח לא יכול להימצא בקורדינטה שבה יש כבר חנות. יש לוודא זאת ולהתריע על כך במקרה הצורך

4. יש לבקש מהמשתמש להכניס את הפריטים והכמויות של ההזמנה המדוברת.

לשם כך יש להציג לו את כל רשימת הפריטים המוגדרת במע'. עבור כל פריט יש להציג:

1. מספרו הסידורי

2. שם הפריט

3. אופן קניית הפריט (לפי כמות או משקל)

4. מחיר הפריט ליחידה (במידה והחנות מוכרת פריט זה. אם לא – פשוט מציינים שהחנות לא מוכרת)

המשתמש יבחר את הפריט המבוקש לקנייה על ידי מספרו הסידורי.

עתה, יש לשאול את המשתמש לגבי כמות רכישה עבור הפריט הנבחר.

אם אופן קניית הפריט הוא לפי כמות – אזי המשתמש יכול להכניס רק מספר שלם.

אם אופן קניית הפריט הוא לפי משקל – אזי המשתמש יכול להכניס כמות עשרונית (למשל 1.5 קילו עגבניות...).

אם המשתמש בחר לקנות פריט שלא נמכר ע"י החנות המבוקשת – יש להציג לו הודעת שגיאה מתאימה.

יש לחזור על (תת) סעיף זה עד שהמשתמש מביע את רצונו לסיים את ההזמנה ע"י הקשה על q

5. לקראת סיכום הגשת ההזמנה, יש להציג למשתמש את סיכום רשימת הפריטים שהזין להזמנה זו:

1. מספר סידורי של פריט

2. שם הפריט

3. אופן קניית הפריט (לפי כמות או משקל)

4. מחיר הפריט

5. כמות הפריט בהזמנה

6. סה"כ מחיר לפריט (מחיר הפריט X כמות הפריט בהזמנה)

יש להציג את עלות המשלוח: לשם כך יש לחשב את המרחק הליניארי ("קו אווירי") בין מיקום החנות המבוקשת

למיקום הלקוח ולהכפיל ב PPK של החנות.

יש לציין את המרחק מהחנות (דיוק של עד 2 ספרות אחרי הנקודה), את ה PPK ואת סך עלות המשלוח.

לבסוף יש לבקש מהמשתמש לאשר את פרטי ההזמנה.

אם אישר – ההזמנה יצאה לפועל ועכשיו כבר נחשבת למבוצעת (אין חשיבות למימד הזמן בתרגיל זה) – כלומר

השלכתה על כמות הפריטים שנמכרה, וההזמנות שבוצעו בכל חנות (פקודות 2 ו 3) כבר יכולות לבוא לידי ביטוי

כאשר הזמנה מבוצעת היא מקבלת מהמע' מספר סידורי חד חד ערכי (אפשר ספרור רץ)

אם המשתמש ביקש לוותר על ההזמנה – חוזרים לתפריט הראשי, ואין לפעולה זו שום השלכה על המע'.

דגשים:

1. ייתכן והמשתמש ייבחר לקנות את אותו הפריט פעמיים באמצעות 2 בחירות שונות. בסיכום אפשר להציג זאת בנפרד או כאיחוד, לבחירתכם.
2. אין לאפשר למשתמש להכניס מיקום הזהה למיקומה של החנות הנבחרת במע'.
3. אין צורך לתמוך באפשרות של "עריכת" הזמנה. יש לעקוב על כל השלבים המתוארים על פי הסדר במדויק.
4. ייתכן והמשתמש יבחר בפריט שהחנות המבוקשת לא מוכרת (לא כל חנות מוכרת כל פריט). במקרה זה יש להתריע בפניו עם הודעה מנומקת ולהמשיך את תהליך ביצוע ההזמנה !
5. אין לאפשר פקודה זו אם לא טעון קובץ תקין במע'.

5. הצגת הסטורית הזמנות במע'

פקודה זו תציג את כל ההזמנות שבוצעו עד כה במע'.
עבור כל הזמנה יש להציג:

1. מספר סידורי
2. תאריך
3. מספר סידורי ושם החנות ממנה בוצעה ההזמנה
4. סך סוגי הפריטים ; סך כמות הפריטים
5. עלות כוללת של הפריטים בהזמנה
6. עלות המשלוח
7. עלות כוללת של ההזמנה (סעיף 5 + 6)

דגשים:

1. במידה ומזמינים פריטים לפי משקל, יש להחשיב אותם בכמות של 1 בלי קשר למשקל הנרכש (למשל 4.8 קילו עגבניות נחשבים ככמות של 1. גם 0.3 קילו של בננות ייחשבו ככמות של 1)
2. אין לאפשר פקודה זו אם לא טעון קובץ תקין במע'.

6. יציאה מהמע'

פקודה זו יוצאת מהמערכת.

חלוקה למודולים

בתרגיל זה חובה לייצר (לפחות) 2 מודולים (מהם תפיקו בהמשך 2 jar'ים):

1. ממשק ה ui, המציג את התפריטים השונים, אחראי על קליטת קלט מהמשתמש והחזרת הפלט למשתמש. שימו לב לזה המודול "האקטיבי", המניע את כל המע'. הוא זה האחראי על פנייה ותפעול מנוע המערכת. כפועל יוצא, כל ההדפסות של מידעים למשתמש (System.out.println) מתבצעות אך ורק מתוך מודול זה ; במודול זה יושבת מטודת ה main ; מודול זה אחראי על לולאת תפעול המע' העיקרית, הצגת התפריטים, איסוף הקלט מהמשתמש, הצגת הפלטים למשתמש וכו'.
2. מנוע המערכת, האחראי על קבלת הפקודות (ממודול ה ui), ביצועם והחזרת פלטים מתאימים. שימו לב שמודול זה "פסיבי", והוא רק מגיב לבקשות ולפקודות המתקבלות ממקורות בלתי ידועים לו (בתרגיל זה מודול #1). בתרגילים הבאים מקורות נוספים יפנו אליו לקבלת מידע וחשוב מאוד להקפיד על כך שמודול זה אינו מכיר/מודע למי פונה אליו.

איך מתחילים ? (המלצה...)

ככלל, מומלץ להתחיל משלד ראשוני שמאפשר קליטת קלט מהמשתמש, העברתו למנוע המע' והחזרת התשובה למשתמש. ברגע שתעשו זאת מההתחלה במצב שבו אתם כבר עובדים עם 2 מודולים – יקל עליכם להרחיב את המע' עוד ועוד ולתמוך בפקודות הנדרשות.

התחילו בהקמת מבני הנתונים של המע' המתארים פריט, חנות, הזמנה. תוכלו להתחיל למלא את מבני הנתונים האלה עצמאית אפילו לפני שתממשו את קריאת המידע מה XML. התחילו בתמיכה בפקודות 2 ו 3 המציגות את המידעים הסטטים שכבר מגיעים מקובץ התרגיל. המשיכו למימוש פקודה 1 שמאפשרת ממש טעינה של מידעים מקובץ (והשתמשו בפקודות 2 ו 3 כדי לוודא את נכונות המידעים והקשרים בין מבני הנתונים שלכם).

רק לאחר מכן תימכו באפשרות לבצע הזמנה חדשה במע'. בעקבותיה תוכלו לבדוק כמובן את תקינות המע' באמצעות השתנות המידע הדינמי בפקודות 2 ו 3.

לבסוף, סיימו בפקודה 5 המציגה את הסטוריית ההזמנות שבוצעו במע'.

בונסים

#	סוג	מהות	למה שווה לי ?	כמה שווה לי ?
1	הקדמת תרופה למכה	ממשו כבר עכשיו את אלגוריתם הקצאת החנויות הדינמי כפי שהוא מוגדר בתרגיל 2 סעיף 6. בביצוע בונס זה יש לשנות את פקודה 4 כך שבחלקה הראשון המשתמש יישאל אם הוא רוצה הקצאה סטטית (האפשרות הרגילה) – ואז הוא ממשיך לבחור חנות אחת ממנה ייקנו הפריטים או אם הוא רוצה הקצאה דינמית – ואז אין צורך לבחור שום חנות – שכן המע' תבחר את החנויות בשבילו.	כי גם ככה הולכים לממש זאת בהמשך – אז למה לא ??	10 נקודות (עד ל 100)
2	הקדמת תרופה למכה	עדכון מוצרים בחנות. ממשו כבר עכשיו את סעיף 7 מתרגיל 2, על פי הנחיותיו השונות. יש להוסיף פקודה חדשה לתפריט, שממנה ניתן יהיה לבחור את מי מתתי הפקודות המוגדרות בסעיף 7 תרגיל 2.	פיסת עוגה... ממש מתנה לי זה עולה יותר	7 נקודות (עד ל 100)
3	הגדלת ראש	שמירת הסטוריית הזמנות אפשרו למשתמש 2 פקודות נוספות בתפריט: א. שמירת כל ההזמנות שבוצעו במע' לקובץ נתונים חיצוני. פורמט הנתונים נתון לבחירתכם. עליכם לבקש מהמשתמש נתיב מלא אליו יישמר קובץ הנתונים המדובר. ב. טעינת קובץ נתונים על הזמנות שבוצעו (מסעיף א'). פקודה זו תיקרא אחרי שכבר נטען קובץ ה XML המכיל את המידע של כל החנויות/מוצרים. בעליית מע', לאחר טעינת קובץ ה XNL ולאחר ביצוע פקודה זו, אפשר יהיה להפעיל את פקודה 2,3,5 ולראות את כל הנתונים המעודכנים לפי התוכן האגור בקובץ	טעימה קלה מדרכי שמירת נתונים	10 נקודות (מעל ל 100)

סבבה, סיימתי. מה ואיך להגיש ?

יש להגיש קובץ zip המכיל:

1. 2.jar'ים (לפחות) שהם כל הקוד שלכם, בצירוף קובץ אצווה (batch) שהפעלתו תריץ את התוכנית (כלומר תבצע `java -jar <class name>` וכו').
2. קבצי קוד המקור של הפרויקט שלכם.
3. קובץ ההגשה יכיל גם קובץ **readme** שיכיל הסבר על המערכת, בחירותיכם השונות במקומות שבהם היו לכם בחירה, כמו גם כל דבר נוסף העולה על דעתכם שחשוב שהבודק תדע.
4. יש לכלול בקובץ ה **readme** גם תיעוד והסבר כללי (וממצה) של המחלקות העיקריות ותפקידם.
5. יש לכלול בקובץ ה **readme** גם פירוט של המגישים שם, ת.ז. ואי מייל זמין ורלבנטי (!!) – במידה ויהיה צורך ליצור קשר.
6. במידה ומימשתם את מי מהבונסים לעיל, ציינו זאת בקובץ ה readme כדי שהבודק ידע לבדוק בהתאם.

בונס שימושי אבל לא יתועד – לא ייבדק !

#	תאריך	שאלה	תשובה
1	3.8.2020	אם קונים פריט לפי משקל בצורה עשרונית, איך לסכום את הכמות שלו בהצגת פירוט ההזמנה (פקודה 5.4)	כל כמות של פריט הנמכר <u>לפי משקל</u> תחשב ככמות של 1 בלבד. (מעודכן עכשיו גם בקובץ התרגיל עצמו בעמוד 17)
2	8.8.2020	האם לקוח יכול להופיע בקורדינטה של חנות, גם כזו שהוא אינו רוכש ממנה ?	לא . כל לקוח נמצא בקורדינטה משלו שאינה זהה לקורדינטה של שום חנות אחרת. כלומר לקוח לעולם לא יהיה בקורדינטה של כל חנות המופיעה בקובץ/במע'. שימו לב: מספר לקוחות (הזמנות) יכולים להימצא באותה הקורדינטה.

פרטים יבשים

תאריך הגשה: **24.9.20**

ציון אפשרי מקסימלי: 132
קושי: מאתגר

צפי תחילת עבודה: 28.8.20

צפי זמן לביצוע: 4 שבועות
משקל התרגיל: 30%

מטרות התרגיל העיקרית

1. מימוש האפליקציה כ desktop application באמצעות Java FX
2. הרחבת מנוע המערכת לתמיכה בפיצ'רים נוספים (מבצעים, הקצאת חנויות דינמית)

דרישות

1. בתרגיל זה תממשו את המערכת כ desktop application באמצעות ארכיטקטורת JavaFX כפי שנלמד בכיתה. בתרגיל זה אין יותר סדרה של "תפריטים" כפי שהיו בתרגיל 1. כל ממשק המשתמש הוא ממשק גרפי בלבד, המבוסס על JavaFX. כל האינטראקציה עם המשתמש מתרחשת דרך פעולותיו על גבי הפקדים השונים שתציגו לו (כפתורים, תיבות טקסט וכו').

2. העבודה במע' תחל בטעינת קובץ XML (כמו שהיה בתרגיל 1). לבד מהמידע לגבי הפריטים/חנויות אפשריות, הקובץ יכיל גם מידע לגבי זהות של לקוחות רשומים במע'. קבצי ה XML יהיו מוגדרים על פי סכמה חדשה של תרגיל 2 (ראו [נספח XML](#) כדי להבין מה התוספות/הבדלים יחסית לסכמה של תרגיל 1)

את קובץ ה XML יש לטעון באמצעות FileChooser Dialog בלבד, כפי שהוצג בדוגמא המסכמת. הטעינה מאחורי הקלעים תתבצע באמצעות task כפי שנלמד בכיתה תוך הצגת חיווי דינמי למשתמש לגבי אופן התקדמות הטעינה (במידת הצורך, הטמיעו השהיה מלאכותית כדי לסמלץ טעינה מורכבת וארוכה)

יש לבצע בדיקת תקינות על תוכנו של קובץ ה XML. כל הבדיקות כפי שבוצעו בתרגיל 1 ובנוסף:

1. בתוך קבוצת הלקוחות, אין 2 לקוחות בעלי id זהה.
2. בכל קורדינטה יש בדיוק חנות אחת, או לקוח אחד, או כלום.
3. במילים אחרות: לא ייתכנו לקוחות באותה הקורדינטה של חנות, או 2 חנויות באותה הקורדינטה וכו'.
3. בהגדרת מבצע (ראו סעיף 3), יש לוודא כי המספרים הסידוריים של המוצרים השונים קיימים במע', והם אכן נמכרים במסגרת החנות שהמבצע מוגדר עבורה.

3. בתרגיל זה תתמכו ביכולת של חנות להגדיר מבצע על המוצרים שבה. המידע על המבצעים יגיע מקובץ ה XML. עליכם לעבור על [נספח זה](#) כדי להבין את ההגדרות השונות של האלמנטים הכרוכים בהגדרת מבצע. למבצע יש שם ייחודי רק לו, מחרוזת תווים פשוטה באנגלית בלי מגבלות מיוחדות (כמובן בלי רווחים בהתחלה או בסוף) כל מבצע מחולק ל 2 חלקים:

1. אם הלקוח קונה מוצר מסוים ובכמות מסויימת **אז**
2. הוא רשאי לקבל סט של מוצרים אחרים, בכמות מסוימת, בתוספת תשלום (התשלום הוא עבור כל מוצר נוסף). במידה והוא רשאי לקבל יותר ממוצר אחד כחלק מהמבצע, בעל החנות יכול להחליט אם צריך לבחור אחד מבין המוצרים המשתתפים במבצע או האם הוא מקבל את כולם. (AND | OR)

דוגמאות:

אם חנות רוצה להגדיר מבצע של 1+1, הרי שהיא תגדיר שאם הלקוח קונה יחידה אחת של מוצר מסויים (נגיד קטשופ) אז הוא רשאי לקבל יחידה אחת של אותו המוצר (קטשופ) בתוספת תשלום של 0

אם חנות רוצה להגדיר מבצע: "קנה 2 מסטיקים וקבל 2 שעונים ב 20 או כוס חד פעמית ב 30" – אזי המבצע יוגדר כך:
אם הלקוח קונה 2 יחידות של מסטיקים, אם ירצה, הוא רשאי לבחור לקבל 2 שעונים במחיר של 20 כל אחד, או כוס חד פעמית אחת במחיר של 30

אם חנות רוצה להגדיר מבצע: "קנה 3 אנטנות וקבל מארז סוללות מהודר ב 10 וגם מטען לנוקיה 3100 ב 20" – אזי המבצע יוגדר כך:
אם הלקוח קונה 3 יחידות של אנטנה, אם ירצה, הוא מקבל אוטומטית את מארז הסוללות המהודר וגם את המטען לנוקיה 3100 בתוספת תשלום של 30 סה"כ

דגשים:

1. לחנות יכול להיות יותר ממבצע אחד באופן כללי, ויותר ממבצע אחד על אותו המוצר (גם במסגרת אותה החנות).
 2. עבור מוצרים שקילים, ניתן להגדיר כמות עשרונית גם לחלק הראשון של המבצע (if-you-buy) וגם לחלק השני של המבצע (then-you-get)
 3. בכל המע' יש כפל מבצעים.
דוגמא: אם יש מבצע 1+1 על קטשופ, וקניתי במקור 2 קטשופים, ניתן לממש את המבצע פעמיים ולצאת סה"כ עם 4 קטשופים.
 4. בכל המקומות שבהם מוגדר מוצר מסויים הכוונה היא לספק את ה id שלו
 5. הלקוח לא חייב לממש את המבצע. הוא יכול לוותר עליו
4. בתרגיל זה תתמכו ברשימת לקוחות רשומה במע'. רשימה הלקוחות תגיע מתוך קובץ ה XML.
לכל לקוח יש מספר סידורי ייחודי, שם ומיקום המוגדר בדמות צמד קורדינטות בטווח שבין 1 ל 50, כולל: [1,50]
מובטח כי בקובץ ה XML יופיע לפחות לקוח אחד.

5. הצגת מידעים:

- יש לאפשר למשתמש לקבל את המידע על הלקוחות הרשומים במע'. על כל לקוח יש להציג:
 - מספר סידורי
 - שם
 - מיקום (אפשר בתצורה של X ; Y)
 - סך הזמנות שביצע עד עכשיו
 - ממוצע מחיר ההזמנות (ללא המשלוח) שביצע עד עכשיו
 - ממוצע התשלום על המשלוחים בלבד שביצע עד עכשיו
- יש לאפשר למשתמש לקבל את המידע על כל החנויות שבמע', כולל כל המידעים של כל חנות, בדומה לפקודה מס' 2 של תרגיל 1.
במידה ומחנות זו נקנו גם מוצרים כחלק מהזמנה דינמית (ראו סעיף 6), יש להציג את המידע של המוצרים הרלבנטיים מחנות זו בלבד, תוך מתן שיוך להזמנה המקורית.
בנוסף יש להציג מידע על כל המבצעים שיש לחנות (אם יש לה מבצעים). חישבו לבד איך יהיה נכון עבורכם להציג את המידע הכולל את כל פרטי המבצע (מה צריך לקנות ומה מקבלים בתמורה, באיזה כמות ובאיזה מחיר..)
- יש לאפשר למשתמש לקבל את כל המידע על כל הפריטים שבמע', כולל המידע על כל פריט, בדומה לפקודה מס' 3 של תרגיל 1
- יש לאפשר למשתמש לקבל את כל המידע על הסטוריית ההזמנות שבוצעה במע', פירוט של אופן הצגת סיכום הזמנה בודדת מוצג בצורה מלאה בסעיף 8.G

6. בתרגיל זה תתמכו באלגוריתם איתור סל מינימלי (או הקצאת חנויות דינמית).
יש לכתוב אלגוריתם שבהינתן רשימת מוצרים, ימצא את סט החנויות שרכישת המוצרים מהן תביא את מחיר הסל למינימום האפשרי. במילים אחרות, אם מוצר קיים ביותר מחנות אחת – יש לבחור את זו שבה מחירו זול יותר.
כדי להקל, אין להתחשב בעלויות המשלוח כחלק מחישוב ערכו המינימלי של הסל. כלומר המטרה היא להביא למינימום את מחירו של סל הקנייה כפונקציה של הפריטים שנבחרו בו בלבד.

דגשים:

- זהו לא קורס אלגוריתמים, ועל כן האלגוריתם לא צריך להיות יעיל בשום צורה ואופן. הוא יכול להיות נאיבי, טיפש שרץ בסדרי גודל (n^2 , n^3 – מה שיוצא אני מרוצה) – כל עוד מדובר על זמן סביר הקלט (מוצרים, חנויות) לא צפוי להיות גדול מדי כך שגם אלגוריתם "טיפש" ירוץ בזמן סביר.
- ייתכן כי יש מוצרים שאינם נמכרים ע"י כל החנויות, ובפרט אף נמכרים ע"י חנות אחת בלבד. במקרה זה אין כמובן בחירה וחנות זו חייבת להיבחר לפחות עבור מוצר זה

7. עדכון מחירים/מוצרים של חנות:

המשתמש יוכל לבחור חנות, ולעדכן את הפריטים השונים שהיא מוכרת. הפעולות שיש לתמוך בהן בהקשר זה:

א. מחיקת מוצר שחנות מוכרת.

שימו לב:

- אם זו החנות היחידה המוכרת מוצר זה – יש למנוע את מחיקת המוצר, כדי שלא ייווצר מצב שבו יש מוצר שלא נמכר ע"י אף אחת מהחנויות. יש לספק הודעה מתאימה למשתמש במקרה זה.
- אם זהו המוצר האחרון שהחנות מוכרת – יש למנוע את מחיקת המוצר, כדי שלא ייווצר מצב שבו יש חנות שלא מוכרת אף מוצר. יש לספק הודעה מתאימה למשתמש במקרה זה.
- אם על המוצר שנמחק מהחנות מוגדר מבצע, הרי שעכשיו המבצע הזה כבר לא תקף וצריך למחוק את המבצע מרשימת המבצעים הקיימת בחנות. יש לספק ללקוח הודעה מתאימה במקרה זה.

ב. הוספת מוצר חדש שהחנות מוכרת (מסך המוצרים האפשריים המוגדרים בקובץ).
אין לאפשר להוסיף מוצר פעמיים לאותה החנות.

ג. עדכון מחיר של מוצר בחנות.

8. תהליך ביצוע ההזמנה:

באמצעות כפתור ייעודי תיכנסו לתהליך של ביצוע הזמנה חדשה במע'.
יש לעקוב על השלבים הבאים, לפי הסדר:

A. יש לבחור את הלקוח בעל ההזמנה (מרשימת הלקוחות האפשריים כמובן – הציגו את מספר הלקוח ואת שמו)
עצה שלי: קומבו-בוקס אידיאלי לזה !

B. יש לבחור תאריך להזמנה. השתמשו ב date picker

C. יש לבחור האם מעוניינים בבחירת חנות סטטית או בהקצאת חנויות דינמית

- אם נבחר סטטי – יש להציע למשתמש לבחור את החנות הנדרשת מרשימת החנויות המוגדרת במע' (יש להציג את מספר החנות, שם החנות, מיקום החנות). ברגע שנבחרה חנות יש להציג למשתמש את עלות המשלוח ממנה (שכן כבר ידוע מיקום הלקוח, מיקום החנות, PPK וכו'). **עצה שלי: קומבו-בוקס אידיאלי לזה !**
- אם נבחר דינמי – לא מציעים למשתמש כלל לבחור חנות, ממשיכים לסעיף הבא.

D. יש להציע למשתמש לבחור פריטים אותם הוא רוצה לרכוש.

במידה ומדובר על בחירת חנות סטטית – ליד כל פריט יופיע מחירו בחנות ;
במידה ומדובר על בחירת חנויות דינמית – ליד כל פריט לא יופיע מחיר, שכן עדיין לא יודעים מאיזו חנות הוא ירכש.
סעיף זה חוזר על עצמו עד שהלקוח לוחץ על כפתור ייעודי שבו הוא מסיים את הבחירה.

עצה שלי: קלאסי ל table view...

E. אם הבחירה בסעיף 3 הייתה דינמית, זו העת להפעיל את אלגוריתם איתור הסל המינימלי האפשרי, ולהחזיר למשתמש מידע על סך החנויות מהן תתבצע ההזמנה. עבור כל חנות יש להציג:

- מספר סידורי
- שם
- מיקום (קורדינטות)

- מרחק, PPK ועלות המשלוח ממנה
 - כמה סוגי מוצרים נרכשים ממנה (לא כמות)
 - סך עלות המוצרים הנרכשים ממנה
- אם הבחירה בסעיף 3 הייתה סטטית – ממשיכים לסעיף הבא.

F. יש להציג למשתמש את סך המבצעים אשר הוא זכאי להם כפונקציה של החנות(יות) מהן ההזמנה מתבצעת.

משתמש זכאי למבצע אם הוא קנה את המוצרים בכמות הנדרשת על פי הטריגר של כל מבצע. ייתכן ומשתמש יהיה זכאי ליותר ממבצע אחד בשל קניה רבה של אותו המוצר המהווה טריגר למבצע. במקרה זה יש לאפשר לו לממש את כמות המבצעים המתאפשרת:

דוגמאות:

- אם משתמש קנה 2 קטשופים, ויש מבצע של 1+1 על הקטשופ, הרי שהוא יכול לממש את המבצע פעמיים – עבור כל קטשופ שהוא קנה במקור.
- אם משתמש קנה 3 ק"ג בננות, ויש מבצע במידה וקונים 1.4 ק"ג בננות, הרי שהמשתמש יוכל לנצל פעמיים את המבצע הנ"ל ($3 > 2.8 = 2 \times 1.4$).

משתמש זכאי למבצעים על המוצרים המקוריים אותם הוא הזמין במסגרת סעיף D ורק עליהם. מוצרים נוספים הנרכשים במסגרת המבצע (כלומר במסגרת סעיף זה, F) לא מזכים את הלקוח במבצע(ים) נוספים.

על כל מבצע יש להציג למשתמש מה הוא זכאי לקבל, באיזה מחיר וכמה יחידות.

המשתמש צריך לבחור האם הוא רוצה לממש את המבצע (הוא לא חייב!).
במידה ומימש את המבצע, המוצרים (בתוספת מחירם) מצטרפים להזמנה.

במידה ובמבצע יש סט מוצרים לבחירה עם ONE-OF – הרי שהלקוח יכול לבחור בדיוק מוצר אחד.
במידה וסט המוצרים מוגדר לבחירה עם ALL-OR-NOTHING – הרי שהלקוח בוחר אם לקבל את כל המוצרים או אף אחד מהם.

עצה שלי: לממש רכיב המתאר מבצע, ולהציג למשתמש מספר מופעים של רכיב זה בתוך VBOX עם scroll pane...

G. לפני סיום ההזמנה יש להציג למשתמש את סיכום פרטי ההזמנה כולה (כל המידעים שנאספו בסעיפים הקודמים).

את הסיכום יש להציג עבור כל חנות שמשתתפת בהזמנה:
במידה ומדובר בהזמנה סטטית – הרי שיש רק חנות אחת.

במידה ומדובר בהזמנה דינמית הרי שיהיו כמה "בלוקים" של פרטי ההזמנה מכל חנות.

עצה שלי: הכינו רכיב המתאר את תוכן הקנייה מחנות מסוימת, ותוכלו להשתמש בו מספר פעמים כראות עינכם

עבור כל חנות בהזמנה יש להציג מידע של החנות:

1. מספר חנות
2. שם חנות
3. PPK
4. מרחק אל הלקוח
5. עלות המשלוח אל הלקוח

כמו כן יש להציג את הפריטים שנקנו ממנה. עבור כל פריט:

1. מספר
2. שם
3. אופן הרכישה (כמות או משקל)
4. כמות
5. מחיר ליחידה
6. סה"כ מחיר (סעיף 4 X סעיף 5)
7. האם המוצר נרכש כחלק ממימוש מבצע או לאו.

שימו לב: ייתכן והמבצע "מוסיף" מוצרים דומים שכבר נקנו.

במקרה זה יש להפריד בין המוצרים השונים: אלו שקנינו במקור, ואלו שהגיעו בזכות המבצע. דוגמא: אם קנינו 2 קטשופ, ואז בזכות מבצע כזה או אחר קיבלנו עוד 3 קטשופ (אולי אף במחיר אחר ליחידה) – אזי יש להראות 2 שורות של קטשופ בפירוט ההזמנה: אלו שהגיעו בקנייה המקורית (2 יחידות במחיר X) ואלו שהגיעו דרך המבצע (3 יחידות במחיר Y).

בסיכום פירוט ההזמנה (אחרי "הבלוקים" של כל החנויות) יש להציג:

1. סך עלות ההזמנה (מחירי הפריטים) – כולל המבצעים.
2. עלות המשלוח הכוללת של ההזמנה (סך עלויות המשלוחים של כל "בלוק" של חנות).
3. עלות סופית של ההזמנה (סעיף 1 + סעיף 2)

H. הלקוח צריך לאשר באמצעות כפתור ייעודי את פרטי ההזמנה. במידה ואישר – ההזמנה מבוצעת מיידית. כל הנתונים הופכים להיות משוקפים במבני הנתונים הרלבנטיים וניתנים לצפייה באמצעות הפקודות השונות כפי שמוגדר בסעיף 5.

8. בתרגיל זה עליכם לתמוך בהצגת מפת החנויות וההזמנות (== הלקוחות)

על גבי המפה ניתן יהיה לראות חיווי לחנויות השונות כמו גם ללקוחות השונים.

לחיצה על חנות תציג פרטים בסיסיים עליה (ID, שם, PPK, סך הזמנות שבוצעו ממנה) ; לחיצה על לקוח תציג פרטים בסיסיים עליו (ID, שם, כמה הזמנות ביצע).

זהו רכיב דינמי, שכן גודל המפה לא ידוע מראש! (אמנם טווח הקורדינטות הוא בין 1 ל 50, אולם **אין לסרטט מפה בגודל זה בצורה קבועה.**)

על המפה להיות מסורטטת כפונקציה של פריסת האלמנטים המופיעים עליה (חנויות, לקוחות), בתוספות "שוליים" של 1 סביבותיהם (כדי שהחנויות/לקוחות לא יהיו ממש על הסף). לשם כך עליכם לחשב מהם ה X וה Y הגדולים ביותר מבין כל מיקומי הלקוחות/חנויות, להוסיף 1 גם ל X וגם ל Y וזה מגדיר את גודל המפה שעליכם לסרטט.

דוגמא:

אם יש 2 חנויות במיקום (x,y) של 3,5 ו 8,9 וכן 2 לקוחות במיקום 4,23 ו 9,12 אזי ה X וה Y המקסימליים שבם יש לסרטט אלמנטים הם 9,23. נוסף 1 ל 2 הקורדינטות ונקבל שגודל המפה הוא 10,24.

(שימו לב: אם יש אלמנטים על המפה ב $X = 1$ או $Y = 1$ – הם יהיו על השול ואין צורך לרפד אותם ב קורדינטה 0)

אמנם המפה יכולה להתמצות באוסף של אלמנטים במרחב במיקומים היחסיים שלהם, אולם, חישובו איך ניתן להציג את המידע בצורה "מעניינת".

9. כחלק מבדיקת המערכת ישונה גודל המסך (resize) ותיבדק המע' שלכם במסך בגודל שונה. עליכם לדאוג לסידור נכון של רכיבי ה ui ולוודא את תקינותם גם בגודל קטן. מומלץ מאוד להשתמש ב scroll pane (יש ללמוד עליו לבד) כדי להציג תוכן גדול בתוך שטח מסך קטן.

למען הסר כל ספק ומניעת כל התחכמות שהיא, אין "לטפל" ב resize ע"י כך שפשוט תמנעו מהמסך להיות resizable.

במידה ומימשתם בונוסים בתרגיל הקודם אין הכרח לגרום להם לפעול גם בתרגיל זה, אולם אם זה מתאפשר זה יחמם את ליבי (למען הסר ספק, חימום ליבי אינו מתורגם להעלאת נקודות).

חלוקה למודולים

בתרגיל זה יש לממש את מודול ה Java FX כ jar עצמאי שעובד מול מודול מנוע המערכת (שגם הוא משתכלל כפונקציה של הדרישות השונות).

מומלץ לפתח את המודול של JavaFX באותו פרוייקט ב intellij, כמודול נוסף ליד מודול מנוע המערכת. (שימו לב: אין צורך לגרום למודול הקונסול של תרגיל 1 להמשיך לעבוד גם עם המנוע של תרגיל 2.)

איך מתחילים ? (המלצה...)

התחילו בהגיית המראה הכללי של המע'. ואז המשיכו לייצר את המע' ב java fx על פי הגדרתה הראשונית.

התחילו בהוספת כפתור של טעינת הקובץ. יצרו את המצב שבו אתם מצליחים לטעון את הקובץ בהצלחה. (כרגע בלי ה TASK אלא רגיל כמו שהיה בתרגיל 1). עכשיו כשהמידעים כבר בפנים, התחילו לעבור על תצוגת המידעים השונים.

התחילו בהגיית איך כל רכיב נראה (רכיב המציג מידע על לקוח ; רכיב המציג מידע על חנות ; רכיב המכיל מידע על מוצר וכו'), ואז צרו את ה CSS, controller, FXML של הרכיבים הללו בהתאם לדרישות השונות. עכשיו שלבו מופעים של הרכיבים האלה באפליקציה הראשית שלכם. ממליץ להתחיל במוצרים, אחרי כן בלקוחות וחנויות.

המשיכו בתהליך הגדרת יצירת הזמנה חדשה, כפי שהיה נהוג בתרגיל 1 (בלי תמיכה במבצעים, בלי תמיכה בהקצאה דינמית). המשיכו ביכולת להציג פרטי הזמנה (עכשיו כשכבר אפשר לייצר הזמנות במע').

לאחר שזה יעבוד היטב, יהיה לכם קל יותר לשלב, ראשית, תמיכה בהקצאה דינמית והשלכותיה על ה UI (לעבוד את הרכיב המציג חנות, המציג הזמנה וכו') ושנית תמיכה בכל הסוגייה של המבצעים השונים.

השאירו לסוף את ההתעסקות עם המפה הדינמית. יהיה יותר קל ותגיעו עם הרבה יותר נסיון לבצע אותה לקראת סוף התרגיל.

בונוסים

היכן שזה רלבנטי (בונוס 1 ו 2) על הבונוסים להגיע "מכובים" בתור התחלה כך שהבודק "יפתח" אותם רק לטובת הבדיקה שלהם

#	סוג	מהות	למה שווה לי ?	כמה שווה לי ?
1	פשוט מגניב !	אנימציות ממשו (לפחות) את האנימציות הבאות: <ul style="list-style-type: none"> בעת בחירת מוצרים במהלך ביצוע הזמנה – המוצר (בייצוג כזה או אחר) "מתעופף" אל עבר עגלת/רשימת הקניות האנימציה צריכה להתבצע לכל היותר במשך 2 שניות יש לאפשר גם לנטרל את האנימציות הנ"ל, כך שהיא לא תאיט את התקדמות המערכת 	כי זה מה-זה קל...	5 נקודות (ועד ל 100)
2	הגדלת ראש נאה	החלפת skin למע' בבונוס זה תממשו (לפחות) 2 ערכות צבעים נוספות על הערכה הדיפולטית, ותאפשרו למשתמש להחליף את ערכות הצבעים. שימו לב ערכת הצבעים כוללת (לכל הפחות): <ol style="list-style-type: none"> החלפת רקע של כל המסך החלפת המראה של הכפתורים על המסך החלפת הפונט והגודל של כל ה label על המסך 	כי זה בכל זאת תרגיל בממשק גרפי – ומה יותר גרפי מזה ???	5 נקודות (ועד ל 100)
3	הגדלת ראש קלילה	ממשו את הפעלת אלגוריתם הקצאת החנויות הדינמי באמצעות task מאחורי הקלעים אשר מעדכן את ה UI בהתקדמות השיבוץ על שלביו השונים של האלגוריתם שלכם. ההתקדמות היא פונקציה של הפריטים השונים בהזמנה. הטמיעו גם כאן השהיה מלאכותית כדי ליצור תחושה של עבודה רבה מאחורי הקלעים...	כבר עשיתם את זה על XML... מה הבעיה לעשות את זה על הטעינה עצמה ?	5 נקודות (ועד ל 100)
4	הקדמת תרופה למכה	אפשרו למשתמש להוסיף באופן דינמי חנות חדשה יש להוסיף את הפרטים הבאים: <ol style="list-style-type: none"> קבלת מספר סידורי לחנות החדשה ← יש לוודא כי הוא אינו קיים כבר במע' קבלת שם החנות קבלת מיקום החנות: 2 קורדינטות בטווח של 1 עד 50, כולל: [1,50] PPK הגדרת המוצרים שהחנות מוכרת. קבלת מחיר לכל יחידה של מוצר מתוך סך המוצרים המוגדרים במע'. החנות לא חייבת למכור את כל המוצרים האפשריים.	קצת סיזיפי – אבל משרד את החוויה !	7 נקודות (מעל ל 100)
5	הגדלת ראש חביבה	אפשרו למשתמש להוסיף באופן דינמי מוצר חדש יש להוסיף את המידעים הבאים: <ol style="list-style-type: none"> קבלת מספר סידורי לפריט החדש ← יש לוודא כי הוא אינו קיים כבר במע'. קבלת שם הפריט קבלת אופן רכישת הפריט (לפי כמות או לפי משקל) לאחר הגדרת מוצר חדש יש לבחור את החנויות שמוכרות אותו ולקבל את המחיר שבו הוא יימכר. (כדי שלא יוצר מצב שבו יש מוצר שלא נמכר ע"י אף אחת מהחנויות).	קצת סיזיפי – אבל משרד את החוויה !	8 נקודות (מעל ל 100)

		לאחר קבלת הפרטים הנ"ל המוצר מצטרף לרשימת המוצרים המוצגים במע' ואינו שונה בשום אופן וצורה מהמוצרים האחרים שנקראו (אולי) מקובץ ה XML.		
6	מאתגר	אפשרו למשתמש להגדיר מבצעים חדשים בחנות לצורך כך על המשתמש להגדיר את 2 שלבי המבצע: איזה מוצר קונים ובאיזו כמות אילו מוצר(ים) ניתן לקבל ובאיזה אופן	12 נקודות (מעל ל 100)	חווייה שתביא אתכם לכדי תחושת שלמות נפשית
7	מעניין	איתור המלצות למבצעים: אפשרו למע' לאתר ולהציע ללקוח להמיר את הפריטים שרכש במבצע שכבר מתקיים במע'. למשל, אם יש מבצע 1+1 על הקטשופ, והלקוח לא היה מודע לזה וכבר בחר 2 קטשופים (כחלק מסעיף d.7), אזי כחלק מסעיף f.7 יידעו את הלקוח כי ביכולתו להמיר את 2 הקטשופים שלו לכאלה שנרכשו במבצע 1+1 (כלומר לא לשלם מחיר על הקטשופ השני). במקרה זה את הקטשופ השני מציגים כמובן כפריט שמומש במסגרת מבצע.	5 נקודות (מעל ל 100)	My supermarket – here I come !

סבבה, סיימתי. מה ואיך להגיש ?

יש להגיש קובץ zip המכיל:

1. jar 2 (לפחות) המהווים את כל הקוד שלכם, בצירוף קובץ אצווה (batch) שהפעלתו תריץ את התוכנית
 2. קבצי קוד המקור של הפרויקט שלכם.
 3. קובץ ההגשה יכיל גם קובץ readme שיכיל הסבר על המערכת, בחירותיכם השונות במקומות שבהם היו לכם בחירה, כמו גם כל דבר נוסף העולה על דעתכם שחשוב שהבודקת תדע.
 4. יש לכלול בקובץ ה **readme** גם תיעוד והסבר כללי (וממצה) של המחלקות העיקריות החדשות ותפקידם. (חישבו מה יסייע לבדוק להיכנס ביתר קלות לקוד שלכם ולהבין מי נגד מי...)
 5. במידה ומימשתם את מי מהבונוסים לעיל, ציינו זאת בקובץ ה readme כדי שהבודק ידע לבדוק בהתאם.
- בונוס שימומש אבל לא יתועד – לא ייבדק !**

שאלות ותשובות

#	תאריך	שאלה	תשובה

פרטים יבשים

צפי תחילת עבודה: 25.9.20

צפי זמן לביצוע: 5 שבועות

משקל התרגיל: 40%

תאריך הגשה: 31.10.2020

ציון אפשרי מקסימלי: 140

קושי: ווא'עליה אלל'בחטי...

מטרות התרגיל העיקרית

1. מימוש המשחק כאפליקציית web.
2. מימוש חשבונות משתמשים וניהול כספים
3. מימוש פידבקים ודירוגים

דרישות

1. בתרגיל זה נפתח מע' web שתציע למשתמשים ממשק נוח לביצוע רכישות מוצרים מחנויות וירטואליות ברחבי הרשת. משתמשים שהם "בעלי חנויות" יוכלו לנסח את המידע על חנויותיהם, מבצעים והמוצרים שהם מוכרים בקובץ ה XML של התוכנה, להעלות אותו לאתר ובכך להפוך אותו לזמין עבור לקוחות מזדמנים פוטנציאליים. חישבו על כל בעל חנויות כמי שמציע את מרכולתו באזור מסויים (נגיד מכולות באזור הגליל הצפון צפון מזרחי) וכך לקוחות מזדמנים לאזור יוכלו לרכוש ממנו מוצרים בצורה זולה ונאמנה. יחד עם זאת, כדי לא לעודד מונופולים, נאפשר גם לבעלי חנויות אחרים "לפתוח" חנויות וירטואליות באזור מסוים ובכך "להתחרות" עם בעלי החנויות המקוריים באספקת מוצרים זולים.
2. בתרגיל זה כל משתמש מתחבר למע' מ browser שלו, והאינטראקציה שלו עם המשתתפים האחרים מתבצעת למעשה אך ורק ע"י השרת המתווך ביניהם. פורמלית, יש לתמוך בחווית משחק תקינה מעל chrome בלבד. גם אם פיתחתם מעל FF או IE (טפס...) – וודאו כי אתם עובדים בצורה מוצלחת מעל chrome. האפליקציה תיבדק רק מעל chrome. בקיצור chrome, קאפיש ?
3. ככלל, כל פרטי המידע במע' צריכים להתעדכן בצורה אוטומטית עבור כל משתמש (== כל browser) בצורה של Pull כפי שלמדנו בשיעור והודגם בדוגמא המסכמת.
4. אין צורך לתמוך ב back כאמצעי למעבר בין המסכים השונים. כמו כן הניחו כי לא נתעלל באפליקציה ולא נשנה את שורת הכתובת, לא נמחק cookies וכו'. הבדיקה תתבצע אך ורק ע"י שימוש בפונקציות שתאפשרו במסכי המשחק השונים (משל היינו הוריקם...)
5. אין צורך לבצע שמירה של נתונים מעבר למופע הנוכחי של השרת. במילים אחרות כשהשרת יורד - כל הנתונים נעלמים. לא שומרים את המשתמשים שנרשמו, את המוצרים/חנויות/הזמנות וכו'.
6. יש לדאוג ולוודא כי מסכי המערכת לסוגם מתנהגים יפה ב resize ולא מחסירים שום פרט. הם יכולים להיות מכוערים, אבל שישארו פונקציונליים (כשם שהיה נהוג בתרגיל 2).

1. מסך רישום למע' (sign up)

במסך זה יש לאפשר למשתמש להיכנס למע'.
לכל משתמש יש שם ייחודי לו ורק לו. את המספר הסידורי הייחודי יקבל הלקוח אוטומטית מהמע' לאחר הרישום.
לכל משתמש יש תפקיד במע': בעל חנות או לקוח (בדיוק תפקיד אחד. חובה שיהיה לו תפקיד אחד)
אחרי רישום מוצלח תעברו למסך מס' 2...

2. מסך ריכוז אזורי החנויות ותכולתם

בעמוד זו יוצגו כל המשתמשים הרשומים למע' ברגע נתון.
על כל משתמש יש להציג את המידעים הבאים:
✓ שם המשתמש
✓ תפקידו (לקוח או בעל חנות)

משתמש מסוג של "בעל חנות" (ורק הוא) יכול לבחור קובץ ממע' הקבצים (חובה לעבוד עם FileChooser Dialog, כפי שהוצג בכיתה) ולהעלות אותו לשרת.
קבצי ה XML יהיו מוגדרים על פי סכמה חדשה של תרגיל 3 (ראה [נספח XML](#) כדי להבין מה התוספות/הבדלים יחסית לסכמה של תרגיל 2)
קובץ ה XML לא יכיל הגדרת לקוחות (כפי שהיה בתרגיל 2) ויכיל תוספת אלמנט חדש שיגדיר את שם האזור, למשל: חנויות באזור השרון, חנויות של רמי לוי וכו'.
המשתמש שהעלה את הקובץ נחשב "לבעלים" של כל החנויות המצויות בו.
משתמש (בעל חנות) יכול להעלות יותר מקובץ אחד. ויכול כך להיחשב לבעלים של יותר מאזור אחד.

בהגיעו לשרת, הקובץ יעבור בדיקת תקינות (כל הבדיקות הרלבנטיות, כפי שהוגדרו בתרגיל 1 ו 2).
במידה ונמצא תקין הודעה מתאימה תוחזר למשתמש ומעכשיו כל המשתמשים מכל הסוגים במע' יכולים לראות את הנתונים הנ"ל אם ירצו בכך.
במידה ונמצא תקול, הודעה מתאימה תוחזר למשתמש שהעלה אותו בלבד והקובץ כמובן לא נכנס למע'.

שימו לב: שם אזור המכירה צריך להיות ייחודי במסגרת כל האפליקציה. כלומר יש לסכל (ולהסביר בהתאם) ניסיון להעלות קובץ המכיל שם אזור מכירה שכבר הועלה למע' מבעוד מועד ע"י אחד המשתמשים.

במסך זה כל סוגי המשתמשים רואים טבלה המציגה את כל אזורי המכירה השונים המנוהלים כרגע במע'.
עבור כל אזור/קובץ יש להציג את המידעים הבאים (רצוי בטבלה, אך לא חובה):

1. שם המשתמש (הבעלים) שהגדיר את אזור המכירה
2. שם אזור המכירה שניתן ע"י המשתמש
3. סך סוגי המוצרים המוצעים למכירה באזור זה
4. סך החנויות המוגדרות באזור זה
5. סך ההזמנות (כמות) שהוגדרו עד כה עבור אזור זה
6. ממוצע מחיר הזמנות (לא כולל המשלוח)

לחיצה על כפתור ייעודי עבור אזור מכירה מסויים מעבירה את המשתמש לעמוד מס' 3... (ראו בהמשך).

ניהול חשבון

בתרגיל זה נתמוך בפיצ'ר של ניהול חשבון כספי והעברת כספים ("בכאילו") עבור המשתמשים.

כל משתמש ינהל חשבון במע' שיאפשר לו להעביר/לקבל כספים בהתאם להזמנות שביצע/מוצרים שמכר. כל משתמש יראה בעמוד זה את יתרת הזכות שיש לו בחשבון. אין צורך לתמוך או לבדוק מצב של כניסה ליתרה שלילית.

ברגע שמתבצעת הזמנה, הכספים מועברים מהלקוח לחנות(יות) מיידית עם התאריך של מועד ביצוע ההזמנה.

עבור משתמשים מסוג 'לקוח', יש להציג כפתור ייעודי שלחיצה עליו תאפשר למשתמש הטענה של חשבונו הפרטי בסכום כסף. יש לבקש סכום זה מהמשתמש כמובן, כמו גם את התאריך של ביצוע הפעולה.

יש לאפשר למשתמש לראות את כלל התנועות בחשבון מיום הקמת המע'.

יש להציג 5 פרטי מידע עבור כל סוג פעולה (תנועה) בחשבון:

1. סוג: הטענה, קבלת תשלום, העברת תשלום
2. תאריך ביצוע הפעולה.
3. סכום הפעולה
4. יתרה לפני ביצוע הפעולה
5. יתרה אחרי ביצוע הפעולה

3. מסך חנויות אזורי

במסך זה 2 סוגי המשתמשים יוכלו לראות את כל החנויות והפריטים המופיעים באזור המדובר.

המשתמש יוכל לראות את רשימת כל הפריטים המוצעים למכירה באזור זה.

יש לספק על כל פריט את המידעים הבאים:

- ✓ מספר סידורי
- ✓ שם הפריט
- ✓ אופן הקניה (לפי כמות או לפי משקל)
- ✓ כמה חנויות מוכרות את הפריט
- ✓ מחיר ממוצע של הפריט (כפונקציה של כל החנויות המוכרות אותו)
- ✓ כמה פעמים סה"כ נמכר הפריט במע' (סה"כ מכירה, לא בכמה הזמנות הוא הופיע).

המשתמש יוכל לראות את רשימת כל החנויות הקיימות באזור זה.

עבור כל חנות יש לספק את המידעים הבאים:

1. מספר סידורי של החנות
 2. שם החנות
 3. שם הבעלים של החנות
 4. קורדינטת החנות
 5. רשימת הפריטים שהחנות מוכרת
- עבור כל פריט יש לכלול את המידעים הבאים:
- ✓ מספר סידורי
 - ✓ שם הפריט
 - ✓ אופן הרכישה (לפי כמות או לפי משקל)
 - ✓ מחיר הפריט ליחידה אחת בחנות
 - ✓ סה"כ פריטים שנמכרו עד עכשיו מחנות זו
6. מספר הזמנות שבוצעו עד כה מהחנות
 7. סך עלות הפריטים שנמכרו מחנות זו
 8. PPK
 9. סך תשלום שקיבלה החנות על משלוחים עד עכשיו

יש להציג כפתור ייעודי המאפשר למשתמש לחזור חזרה לעמוד מס' 2 (אין להסתמך על back של ה browser) בהתאם ל"תפקיד" של המשתמש (כפי שנקבע בעמוד הלוגין) ייתכנו מספר אפשרויות שונות לפעולה בעמוד זה.

אפשרויות עבור לקוח

1. ביצוע הזמנת מוצרים:

בדומה לסדר הפעולות כפי שהוא מוגדר בתרגיל 2, יש להציג ללקוח את תהליך ביצוע ההזמנה. בנוסף כחלק מתהליך הזמנה יש להזין את המיקום של הלקוח (קורדינטה, $Y \mid X$, כרגיל...). הלקוח רואה פריטים שהוא יכול לקנות שקיימים אך ורק בקובץ המדובר (ולא כאלה שקיימים בקבצים אחרים). ברגע שלקוח ביצע הזמנה, התראה על כך נשלחת ל(כל) החנות(יות) המעורבות בהזמנה. ההתראה תכלול פרטים על ההזמנה: מס' הזמנה; שם לקוח; סך סוגי הפריטים למכירה; סך עלות הפריטים; עלות המשלוח. לאחר אישור ההזמנה הלקוח יעביר סכום הכסף של ההזמנה מחשבון הלקוח לחשבון בעל החנויות.

2. דירוג בעלי החנויות:

בסיום ביצוע הזמנה יש להציע ללקוח אפשרות לדרג ולתת פידבק ל(כל אחת מ) החנות(יות) שממנה בוצעה ההזמנה. שימו לב כי הלקוח יכול לבחור לדרג את כל החנויות או חלקן, אך לא חייב לעשות כן ויכול לבחור לוותר על שלב זה. עבור כל דירוג חנות, הדירוג יכול מספר בין 1 ל 5 (1 נמוך; 5 גבוה) ולבחירתו גם פידבק מילולי על טיב השירות. בסיום מתן הפידבק, התראה מתאימה תשלח ל(כל) בעל(י) החנויות הרלבנטיות.

3. מידע לגבי הזמנות פרטיות:

הלקוח יכול לראות במסך זה מידע לגבי ההזמנות שהוא ביצע.

על כל הזמנה יש לכלול את המידעים הבאים:

- ✓ מספר סידורי
- ✓ תאריך ביצוע ההזמנה
- ✓ מיקום יעד ההזמנה (קורדינטה $Y \mid X$)
- ✓ סך החנויות מהן בוצעה הרכישה
- ✓ סך כמות הפריטים
- ✓ עלות הפריטים
- ✓ עלות המשלוח
- ✓ עלות כוללת של ההזמנה

לחיצה על שורת ההזמנה תציג את פרטי תכולת ההזמנה:

- ✓ מספר פריט
- ✓ שם הפריט
- ✓ אופן הרכישה (כמות או משקל)
- ✓ מאיזה חנות נלקח (קוד חנות + שם)
- ✓ כמות רכישה
- ✓ מחיר ליחידה
- ✓ סך מחיר
- ✓ האם חלק ממבצע

דגשים:

- התראות צריכות להתקבל אצל בעל החנויות בין אם הוא נמצא בעמוד 2 או בעמוד 3
- ההתראות צריכות להתקבל גם אם הוא מביט על קובץ חנויות אזורי אחר.

1. פירוט הזמנות בחנות:

כל בעל חנות יוכל לקבל פירוט של ההזמנות שבוצעו בחנות שבבעלותו בלבד בלחיצה על חנות שכזו (המוצגת במסך זה) יש להציג את הסטוריית ההזמנות שהתרחשו בה. עבור כל הזמנה יש להציג את המידעים הבאים:

- ✓ מספר הזמנה
- ✓ תאריך ביצוע הזמנה
- ✓ שם הלקוח המזמין ומיקומו (קורדינטה X ו Y)
- ✓ סך כמות הפריטים שנקנו בהזמנה זו
- ✓ עלות כוללת של הפריטים שנקנו בהזמנה זו
- ✓ עלות המשלוח בהזמנה זו

(במידה ומדובר בהזמנה דינמית שרק חלקה נקנה בחנות זו יש להציג את הסיכומים הנ"ל הרלבנטיים רק לחלק של החנות בהזמנה הכוללת)

לחיצה על שורת הזמנה תציג רק את הפריטים בהזמנה שנקנו מחנות זו בלבד. עבור כל פריט יש להציג את המידעים הבאים:

- ✓ מספר פריט
- ✓ שם הפריט
- ✓ אופן הרכישה (כמות או משקל)
- ✓ כמות רכישה
- ✓ מחיר ליחידה
- ✓ סך מחיר
- ✓ האם חלק ממבצע

2. הצגת פידבקים:

הציגו כפתור ייעודי המציג במרכז את כל הפידבקים שקיבל בעל חנות זה מאז עליית המע' (הפידבקים שהתקבלו גם כהתראה). יש להציג את הפידבקים בהקשר החנויות שבאזור/קובץ שמוצג בעמוד זה ברגע נתון (ולא את אלה של אזורים אחרים).

עבור כל פידבק הציגו:

- ✓ שם הלקוח שנתן את הפידבק
- ✓ תאריך ההזמנה שבמסגרתה ניתן הפידבק המדובר.
- ✓ דירוג מספרי
- ✓ תיאור מילולי (במידה וניתן)

3. פתיחת חנות חדשה:

בעל חנות יכול לפתוח חנות חדשה באזור שבו הוא כבר הבעלים או באזור הנמצא "בבעלות" בעל חנות אחרת במע'. בתהליך הגדרת חנות חדשה יש לאסוף מהמשתמש את הפרטים הבאים:

- ✓ שם החנות
- ✓ מיקום (2 קורדינטות בטווח של 1 עד 50, כולל: [1,50]). יש לוודא (ולהתריע כמובן) כי לא בוחרים קורדינטות עליהם יש כבר חנות אחרת.
- ✓ PPK
- ✓ פריטים שהחנות מוכרת:

יש לאפשר למשתמש לבחור פריט מסך הפריטים המוצעים למכירה באזור זה, ולהציע את המחיר ליחידה עבורו. החנות לא חייבת למכור את כל אחד מהפריטים המוצעים באזור. פקודה זו חוזרת על עצמה עד שבעל החנות בחר את כל הפריטים או בחר לעצור. יש להבטיח כי ייבחר לפחות פריט אחד למכירה בחנות.

בסיום הגדרת החנות היא מתווספת לרשימת החנויות באזור, אולם הבעלים שלה הוא המשתמש שכרגע יצר אותה. התראה מועברת בזאת לבעלים של אזור המכירה, המציינת שנפתחה חנות חדשה באזורו ע"י מתחרה.

ההתראה תכלול את הפרטים הבאים:

- ✓ שם בעל החנות החדש
- ✓ שם החנות
- ✓ מיקום החנות
- ✓ סך הפריטים שהחנות מוכרת מתוך סך הפריטים האפשריים המוצאים למכירה באזור (אפשר בפורמט של $x \setminus y$)

כמו ההתראות האחרות, התראה זו תגיע לבעל החנות בין אם הוא בעמוד 2 או 3 ; בין אם הוא מסתכל על האזור זה או אחר.

ריכוז סוגי התראות שבעל חנות יכול לקבל:

1. בוצעה הזמנה הכוללת פריטים ממי מהחנויות שבבעלותך
2. קבלת פידבק מלקוח בעקבות ביצוע הזמנה ממי מהחנויות שבבעלותך
3. חנות חדשה נפתחה באזורך

חלוקה למודולים

יש לייצר מודול נוסף שממנו ייבנה war המכיל את כל ה jar'ים הנדרשים (Jar של מנוע המע' ; gson.jar ; אולי אחרים ?). מודול זה מכיל גם את קוד ה UI (HTML, CSS, JS) וגם את קוד הסרבלטים המקשרים למודול המנוע.

איך מתחילים ?

אני ממליץ להתחיל במעבר יסודי על הדוגמא המסכמת של הקורס. היא תקנה לכם שליטה בסיסית ברזי אפליקצית ווב (בסיסית) ובתפקידים השונים של כל רכיב ושחקן בסיפור.

לאחר מכן התחילו בעבודה על פי סדר העמודים:

ראשית התחילו בעמוד הלוגין. הוא יכול להיות זהה בעיקרו לעמוד הלוגין כפי שהוצג בדוגמא המסכמת (הרגישו חופשי "להעתיק" משם את המידע הנדרש לכם. מתנה ממני אליכם. כזה אני. גדוש בטוב).
תו"כ כתיבת עמוד זה תתרגלו את שלב ה UI ל שרת וחזרה ל UI. זה גם ידרוש מכם להקים בצד השרת אובייקט כזה או אחר שמנהל את המשתמשים וכן הלאה.

לאחר מכן התחילו לעבוד על עמוד מס' 2. ביחרו פיצ'ר נוסף וממשו אותו בצורה מלאה מה-UI, עבור לסרבלט המתאים בשרת וחזרה ל UI, בו אתם מייצרים את ה HTML בהתאם לתשובה שחזרה מהשרת.

ממליץ מאוד (כמו ב JAVA FX) לייצר עמוד HTML סטטי שמכיל "סקיצה" ראשונית של המידעים שאתם רוצים להציג כך שתוכלו להבין איך הוא אמור להיראות, מי הרכיבים שבו, איך לעצב אותו מבחינת CSS וכן הלאה. זה גם ייתן לכם הבנה ראשונית של מהם כל קריאות ה HTTP שאתם עתידים לייצר מכל רכיב ורכיב, באיזו תדירות הם יוצאים ? איזה סרבלטים משרתים אותם (האם סרבלט אחד לכולם או כמה "סרבלטונים" קטנים ?) וכן הלאה. אחרי שיש לכם סקיצה – תוכלו ב JS להתחיל לייצר את התוכן ה HTML של כל אזור ואזור ולעבוד על רכיב ורכיב "לעומק" (UI <- SERVER <- UI).

כך מסיימים את עמוד 2 ואז עוברים באותו אופן לעמוד 3. אחרי שכל התשתית והשלד הראשי של האפליקציה בנוי – מוסיפים את כל הכפתורים השונים ומפתחים כל פיצ'ר עד סופו.

#	סוג	מהות	למה שווה לי ?	כמה שווה לי ?
1	הגדלת ראש מגניבה !	Chat: אפשרו למשתמשים לקיים chat ביניהם תו"כ התחברותם למע'. ה chat צריך להתנהל כולו אל מול השרת כמובן בצורה שבה כולם רואים את מה שכולם כותבים. (מאוווווווווו דומה לדוגמא הסופית) הצ'ט מתבצע בין המשתמשים הרשומים והמחברים למערכת בלבד.	היש מדהים מזה ???	10 נקודות (ועד ל 100)
2	הגדלת ראש זוטא	אפשרו למשתמש שהוא בעל חנות בלבד להוסיף מוצר חדש למכירה באזור שלו. בנוסף זה מתרחש בעמוד מס' 3. יש לספק את המידעים הבאים: ✓ קבלת שם הפריט ✓ קבלת אופן רכישת הפריט (לפי כמות או לפי משקל) הפריט מקבל מספר סידורי חדש אוטומטית מהמע'. מייד אחרי הגדרת הפריט יש לבחור חנות אחת או יותר שתאפשר את מכירתו (כדי שלא יוצר מצב שיש פריט שלא נמכר על ידי אף אחת מהחנויות).	פיסת עוגה ממש...	10 נקודות (ועד ל 100)
3	הגדלת ראש בלתי ניתנת להכלה במגניבותה !	יצרו אלמנט גרפי של מפה המציגה את האלמנטים השונים במע' על פי מיקומם. את המפה יש להציג בעמוד מס' 3, הן עבור הלקוח והן עבור בעל החנות יש להציג על המפה את החנויות וההזמנות השונות במע' בהתאם למיקומים שלהם על פי הקורדינטות הרלבנטיות. בהקלקה על כל חנות יש להציג את המידעים של החנות: • מספר • שם • PPK • מספר הזמנות שהייתה שותפה להם עד כה כמו כן יודגשו באמצעות חיווי כזה או אחר כל הלקוחות שכבר רכשו מן החנות. בהקלקה על כל לקוח יש להציג את המידעים הבאים: • מספר • שם • מספר ההזמנות שביצע עד כה במע' כמו כן יודגשו באמצעות חיווי כזה או אחר כל החנויות מהן בוצעה ההזמנה	MySupermarket: eat my dust !	15 נקודות (מעל ל 100)
4	הגדלת ראש סופר דופר !	ממשו את שכבת ה UI של תרגיל 3 כחלק מאפליקציית Java FX. אפשר ורצוי כמובן להסתמך על האפליקציה הקיימת שכבר מימשתם בתרגיל 2. בבנוסף זה יש לאפשר למשתמש להירשם למע', לאחר מכן לייצר עבורו עמוד (java fx) המקביל לעמוד 2 שבו יוכל לראות את פרטי האזורים השונים. כמו גם את פרטי החשבון שלו, כפי שמתואר בעמוד מס' 2 (אפשר לוותר על הצגת רשימת המשתמשים במע'). משם יוכל לבחור את אחד מאזורי המכירה ולאפשר להתנהל באזור זה כפי שכבר מומש זה לא מכבר בתרגיל 2 !! שימו לב: אין צורך לתמוך מממשק JavaFX ביכולת הוספת חנות חדשה לאזור. או ביכולת של לקוח לפדבק את בעלי החנויות. יש צורך לתמוך בקבלת התראות וצפייה בפידיבקים שכבר ניתנו ע"י משתמשים אחרים. עליכם לבחור ולהשתמש כאן ב HTTP Client שיפעל מול <u>אותו</u> השרת, כפי שקורה בתרגיל 3 (למעשה בנוסף זה ייבדק כך שחלק מהמשתמשים הם ב browser וחלקם ב java fx).	Oh My God !!!	20 נקודות (מעל ל 100)

סבבה, סיימתי. מה ואיך להגיש ?

יש להגיש קובץ zip המכיל:

1. WAR אחד בדיוק שיושם בספריית tomcat ויעבור deployment אוטומטי. על WAR זה לכלול את כל התלויות שלכם, אין להניח כי יסופקו לכם תלויות מבחוץ (למשל Gson וכו').
2. קבצי קוד המקור של הפרויקט שלכם (גם צד השרת וגם צד הלקוח).
3. קובץ ההגשה יכיל גם קובץ readme שיכיל הסבר על המערכת, בחירותיכם השונות במקומות שבהם היו לכם בחירה, כמו גם כל דבר נוסף העולה על דעתכם שחשוב שהבודקת תדע.
4. **בפרט חייב להכיל קובץ ה readme את ה url שאיתו מנווטים אל מסך ה login של המשחק שלכם.** יש לכלול בקובץ ה readme גם תיעוד והסבר כללי (וממצה) של המחלקות העיקריות החדשות ותפקידם. (חישבו מה יסייע לבודקת להיכנס ביתר קלות לקוד שלכם ולהבין מי נגד מי...)
5. במידה ומימשתם את מי מהבונסים לעיל, ציינו זאת בקובץ ה readme כדי שהבודקת תדע לבדוק בהתאם.
בונס שימושי אבל לא יתועד – לא ייבדק !
הגשה באיחור, שאינה באישור, תבטל כל מימוש בונס. אין להגיש באיחור בשביל להספיק לעשות בונסים. תכננו את הזמן בהתאם.

שאלות ותשובות

#	תאריך	שאלה	תשובה
1			

תיאור מבנה המערכת מפורט כחלק מקובץ ה XML

במהלך הבדיקה (של שלל התרגילים), תיבדק המערכת באמצעות מספר קבצים שונים, חלקם חוקיים וחלקם תקולים (תקולים אפליקטיביים, תקינים סכמטיים), במטרה לראות האם וכיצד המערכת מגיבה לשגיאות.

בחנו היטיב את קבצי הדוגמא שהועלו למע' ה Mama וודאו כי אתם מבינים את פרטיהם ומבניהם.

היכן שמצויין case insensitive הכוונה היא שאין חשיבות ל case של האותיות באנגלית. במקרה זה הערך milk זהה לערך Milk

היכן שמצויין שהמחרוזת יכולה להכיל רווחים – המדובר הוא רק על רווחים בתוך המחרוזת ולא על רווחים בתחילתה/סופה

סכמת קובץ XML – גרסה I

הקובץ מתאר מבנה של מוצרים וחנויות באזור מסויים:

```
<super-duper-market-descriptor xn
  <SDM-items>
  <SDM-stores>
</super-duper-market-descriptor>
```

#	סוג	שם	מהות
1	Element	super-duper-market-descriptor	מתאר את המבנה הכללי של אזור המכירה: חנויות, ומוצרים
2	Element	SDM-items	מתאר את כל הפריטים/מוצרים המוצעים למכירה במע'. לפחות פריט אחד יופיע בפנים.
3	Element	SDM-stores	מתאר את כל החנויות המוגדרות במע'. לפחות חנות אחת תופיע בפנים.

```
<SDM-items>
  <SDM-item id="1">
  <SDM-item id="5">
</SDM-items>
```

SDM-items

#	סוג	שם	מהות
1	Element	SDM-item	מתאר פריט אחד במע'.
2	Attribute	id	מספרו הייחודי של הפריט. מספר חיובי שלם.

SDM-item

```
<SDM-item id="1">
  <name>Ketshop</name>
  <purchase-category>Quantity</purchase-category>
</SDM-item>
```

#	סוג	שם	מהות
1	Element	name	שם המוצר. באנגלית בלבד. יכול להכיל רווחים. שמות המוצרים הם case insensitive (כלומר milk הוא אותו הדבר כמו Milk)
2	Element	Purchase-category	אופן רכישת המוצר. יכול להכיל רק Quantity או Weight

SDM-store

```
<SDM-store id="1">
  <name>super baba</name>
  <delivery-ppk>30</delivery-ppk>
  <location x="3" y="4"/>
  <SDM-prices>
    <SDM-sell item-id="1" price="20"/>
    <SDM-sell item-id="2" price="10"/>
    <SDM-sell item-id="5" price="50"/>
  </SDM-prices>
</SDM-store>
```

#	סוג	שם	מהות
1	Element	SDM-store	מתאר הגדרה של חנות אחת במע'
2	Attribute	id	מספר סידורי ייחודי לחנות. מספר שלם חיובי
3	Element	name	שם החנות. באנגלית. יכול להכיל רווחים. השמות הם case insensitive.
4	Element	delivery-ppk	עלות המשלוח עבור ק"מ מרחק. מספר שלם חיובי.
5	Element	location	מתאר את מיקום החנות במרחב. מכיל 2 attributes עבור קורדינטת x וקורדינטת y של המיקום. גם x וגם y הם מספרים שלמים חיוביים.
6	Element	SDM-prices	מתאר את טבלת המחירים של המוצרים השונים שחנות זו מוכרת

```
<SDM-sell item-id="2" price="20"/>
```

SDM-sell

#	סוג	שם	מהות
1	Element	SDM-sell	מתאר את פרטי מכירה של פריט אחד במסגרת חנות
2	Attribute	Item-id	מספר סידורי של הפריט לגביו מספקים את הפרטים
3	Attribute	price	מחיר ליחידה אחת של המוצר בחנות זו

סכמת קובץ XML – גרסה II

עדכונים מגרסה קודמת:

1. נוסף אלמנט רשות חדש בתחומי הגדרת כל חנות שמאפשר להגדיר "מבצע" שהחנות נוקטת בו.
2. נוסף אלמנט חובה חדש (ראשי) המגדיר את הלקוחות הרשומים במע'

SDM-discounts

```
<SDM-discounts>
  <SDM-discount>
    <SDM-discount>
  </SDM-discount>
</SDM-discounts>
```

#	סוג	שם	מהות
1	Element	SDM-discounts	מתאר את כל המידע לגבי המבצעים שנערכים בחנות

SDM-discount

```
<SDM-discount>
  <name>Balabait ishtagea !</name>
  <if-you-buy item-id="1" quantity="1"/>
  <then-you-get operator="ONE-OF">
    <SDM-offer item-id="4" quantity="1" for-additional="0"/>
    <SDM-offer item-id="5" quantity="2" for-additional="20"/>
  </then-you-get>
</SDM-discount>
```

#	סוג	שם	מהות
1	Element	name	מתאר את שם המבצע באנגלית. יכול להכיל רווחים, מספרים, תווים מיוחדים וכו'. case insensitive
2	Element	If-you-buy	מתאר את הטריגר להתקיימות המבצע. אילו מוצרים צריך לקנות (ובאיזו כמות) כדי להיות זכאי למבצע
3	Attribute	Item-id	מספר הפריט הייחודי שיש לרכוש כדי להיות זכאי למבצע. מספר חיובי שלם.
4	Attribute	Quantity	כמות שיש לקנות בה את הפריט המסוים כדי להיות זכאי למבצע. מספר חיובי עשרוני (יכול לייצג שבר כמו 0.5, בעיקר כדי לתאר מצב של מוצרים שקילים)
5	Element	then-you-get	מתאר את מהות המבצע – אילו מוצרים ניתן לקבל במימוש המבצע.
6	Attribute	operator	במידה ומימוש המבצע מזכה ביותר מפריט אחד, מתאר את אופן המימוש של סט המוצרים. יכול להכיל 3 ערכים בלבד: <ul style="list-style-type: none"> • IRRELEVANT – ערך ברירת המחדל במקרה ולא מופיע. מופיע כשיש בדיוק פריט אחד למימוש במסגרת המבצע ואז אין דילמה איך להתנהג איתו. • ONE-OF – מתאר שעל הלקוח לבחור בדיוק אחת מבין האפשרויות למימוש המבצע. • ALL-OR-NOTHING – מתאר שהלקוח מקבל את כל אפשרויות מימוש המבצע ולא יכול לבחור לקבל רק חלק מהם. זהו פרמטר אופציונלי שלא חייב להופיע. ערך ברירת מחדל: IRRELEVANT במידה ויש יותר מאפשרות מימוש אחת למבצע – חייב להופיע עם ערך ONE-OF או ALL-OR-NOTHING

SDM-OFFER

```
<SDM-offer item-id="4" quantity="1" for-additional="0"/>
```

#	סוג	שם	מהות
1	Element	SDM-offer	מתאר את אחת הזכאויות האפשריות למימוש המבצע
2	Attribute	Item-id	מספר סידורי של הפריט שאפשר לקבל במסגרת מימוש זה. מספר חיובי שלם.
3	Attribute	quantity	כמה יחידות של פריט זה אפשר לקבל במסגרת מימוש המבצע. מספר חיובי עשרוני (יכול לייצג שבר, בעיקר בשביל מוצרים שקילים)
4	Attribute	for-additional	איזה מחיר צריך לשלם עבור כל יחידה במסגרת מימוש המבצע. מספר חיובי שלם. יכול להיות גם 0. עבור מוצרים שקילים הכוונה כאן היא למחיר עבור ק"ג (כמו בהגדרת מחיר של מוצר שקיל בחנות)

SDM-Customers

```
<SDM-customers>
  <SDM-customer id="1">
    <name>Menash</name>
    <location x="7" y="3"/>
  </SDM-customer>
  <SDM-customer id="2">
  </SDM-customer>
</SDM-customers>
```

#	סוג	שם	מהות
1	Element	SDM-Customers	מתאר את הלקוחות הרשומים במע'.
2	Element	SDM-customer	מתאר פרטי לקוח בודד במע'.
3	Attribute	Id	מתאר מספר לקוח ייחודי. מספר חיובי שלם.
4	Element	Name	מתאר את שם הלקוח. באנגלית. יכול להכיל רווחים.
5	Element	Location	מתאר את מיקום הלקוח בקורדינטה על גבי המפה. זהה להגדרת מיקום החנות

סכמת קובץ XML – גרסה III

עדכונים מגרסה קודמת:

1. אין הגדרת לקוחות.
2. נוסף אלמנט חובה חדש (ראשי) המגדיר את שם האזור שבו מצויות החנויות

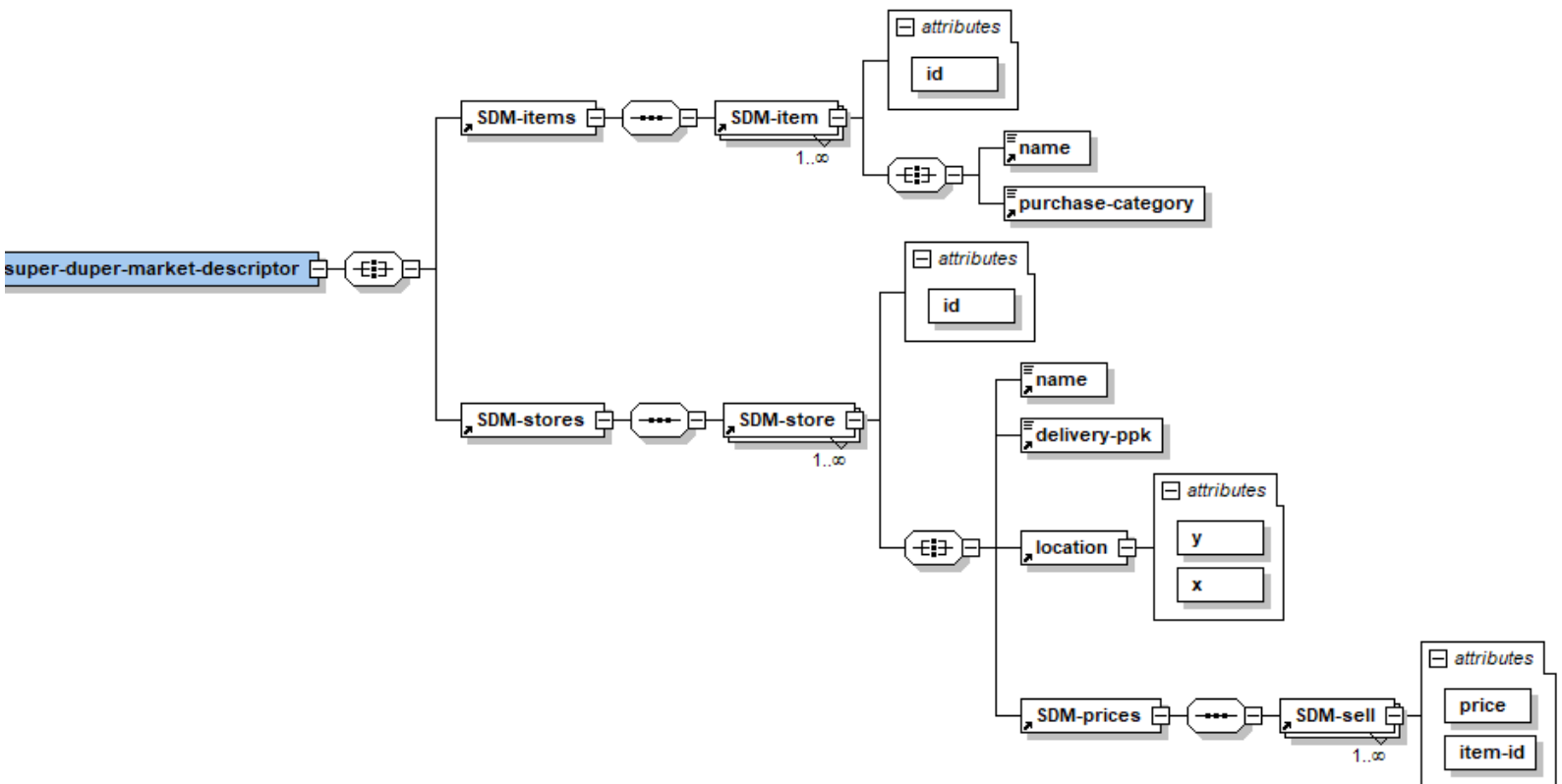
SDM-Zone

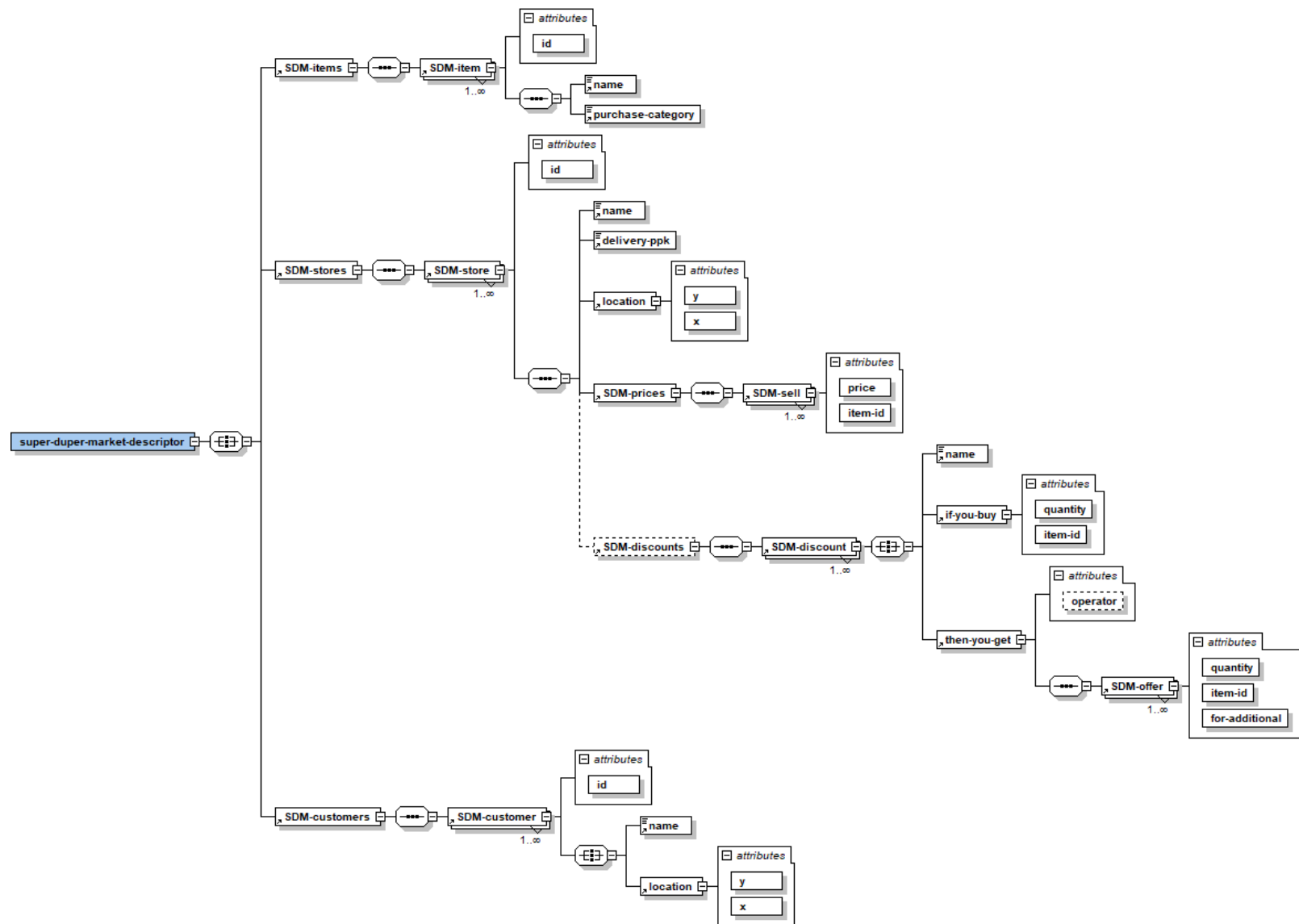
```
<SDM-Zone name="Galil Maarvi"/>
```

#	סוג	שם	מהות
1	Element	SDM-Zone	מתאר את האזור שבו נמצאות החנויות
3	Attribute	name	מתאר את שם האזור. שם באנגלית, יכול להכיל רווחים

קובץ ה XML מקושר לסכמה שמגדירה את חלקיו השונים ואת חוקיותם.
להלן תרשימים המתארים את מבנה הסכמות בגרסאותיהם השונות:

סכמה – גרסה 1





10/10/2020

