s

מס' גרסה:6

פיתוח תוכנה מבוסס java

או בעברית: שטיסלSheet-cell קיץ 2024 -

מרצה: **אביעד כהן** [aviadco@mta.ac.il](mailto:aviadco@mta.ac.il)

בודק: **איתי כהן**  itaych@mta.ac.il

התרגיל מנוסח בלשון זכר, אך מכוון לכלל המגדרים והתחושות בצורה שווה

תוכן העניינים

[דרישות הקורס 3](#_Toc175688872)

[כללי 3](#_Toc175688873)

[איך להגיש תרגילים באיחור, ולהישאר בחיים 4](#_Toc175688874)

[הנחיות כלליות לכתיבת התרגיל 5](#_Toc175688875)

[Sheet-Cell 7](#_Toc175688876)

[מוטיבציה 7](#_Toc175688877)

[שטיסל – תיאור המע' 8](#_Toc175688878)

[הנחיות כלליות למימוש התרגיל 9](#_Toc175688879)

[תרגיל 1 – מימוש **שטיסל** כאפליקציית Console (20%) - הגשה: 29.8.24 10](#_Toc175688880)

[פרטים יבשים 10](#_Toc175688881)

[דרישות 10](#_Toc175688882)

[חלוקה למודולים 13](#_Toc175688883)

[איך מתחילים ? (המלצה...) 14](#_Toc175688884)

[בונוסים 14](#_Toc175688885)

[סבבה, סיימתי. מה ואיך להגיש ? 14](#_Toc175688886)

[תרגיל 2 – מימוש **שטיסל** כאפליקציית JavaFX (40%) – הגשה: 28.9.24 15](#_Toc175688887)

[פרטים יבשים 15](#_Toc175688888)

[דרישות 15](#_Toc175688889)

[איך מתחילים ? 19](#_Toc175688890)

[בונוסים 20](#_Toc175688891)

[סבבה, סיימתי. מה ואיך להגיש ? 21](#_Toc175688892)

[תרגיל 3 – מימוש **שטיסל** כאפליקציית Client - Server (35%) – הגשה: 9.11.24 22](#_Toc175688893)

[פרטים יבשים 22](#_Toc175688894)

[דרישות 22](#_Toc175688895)

[חלוקה למודולים 25](#_Toc175688896)

[איך מתחילים ? 25](#_Toc175688897)

[בונוסים 25](#_Toc175688898)

[סבבה, סיימתי. מה ואיך להגיש ? 26](#_Toc175688899)

[נספח א' –פונצקיות עזר נתמכות 27](#_Toc175688900)

[נספח ב' – תיאור מבנה המערכת באמצעות קובץ XML 33](#_Toc175688901)

[סכמה תרגיל 2 34](#_Toc175688902)

[נספח ג' – תיאור גרפי של הסכמה 35](#_Toc175688903)

דרישות הקורס

## כללי

1. בקורס אין בחינה אך חובה להגיש תרגילים (סה"כ 4).
2. המלצתי היא להגיש את התרגילים ביחידים. אולם אם אין ברירה - את מרבית התרגילים (3) ניתן להגיש בזוגות, אך לא בשלישיות / רביעיות / חמישיות או יותר

(כן, גם אם מדובר בשלישיה / רביעיה / חמישיה הצועדת יחדיו לאורך שנים מאז גיל הגן והגישה עד עכשיו את כל הפרוייקטים ביחד).

את תרגיל ה - [reflection](#_כללי) חובה להגיש ביחידים.

1. בעבודה משותפת על תרגיל יש להקפיד על מעורבות אקטיבית של כלל המגישים בכל חלקי התרגיל.
2. במידה והוגדר בונוס לתרגיל מסוים, ציון הבונוס יתווסף לציון התרגיל בלבד (ולא לציון הסופי של הקורס כולו).
3. התרגילים יוגשו דרך מערכת Mama. מוגדר רכיב 'מטלה' נפרד לכל תרגיל.
4. 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.

1. ניתן להחליף את השותפ/ה בכל תרגיל, ללא צורך באישור או הודעה למרצה.

הניקוד על כל תרגיל נזקף לזכות הסטודנטים שבצעו אותו בלבד.

1. יש להעלות את התרגיל רק עבור אחד מבני הזוג ולהוסיף את שם בת/בן הזוג ומספר תעודת הזהות שלה/ו בקובץ ה readme המצורף (פרטים בהמשך).

## איך להגיש תרגילים באיחור, ולהישאר בחיים

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 הבודק יידע אתכם ויאפשר לכם לבדוק, לתקן ולהגיש מחדש את התרגיל כדי שאפשר יהיה לבודקו 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.

1. עומס בלימודים, בעבודה, בחיים, בגלל הילדים או ההורים, שכנים וחברים (או בכל תחום אחר) אינו נחשב כסיבה לגיטימית לבקשת הארכה.
2. במקרה של בקשה להארכה (מכל סיבה שהיא, לרבות מילואים ומחלה) יש לפנות למרצה מראש על מנת לקבל אישור.  
   הפנייה תתבצע במייל.
3. 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.
4. ניתן לערער על ציון של תרגיל לכל היותר שבוע מיום פרסום המשוב והציון במע' המאמא.

כדי לערער יש לשלוח מייל לבודק בצירוף כל הסיבות והטענות שלכם.

## הנחיות כלליות לכתיבת התרגיל

1. 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.

1. יש להגיש את התרגילים בתור קובץ zip/rar (לא 7Z !)

הקובץ יכיל:

1. כל הקבצים הרלבנטים להפעלת התרגיל (jar/war – 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 וכיוצב'.

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 הreadme להיות בפורמט word או pdf (**לא notepad !**).   
חי נפשי – אם מישהו מגיש readme כקובץ טקסט פשוט -ירד לו ניקוד מהתרגיל...

1. דווקא בגלל שאין זהו קורס שבו יכנסו לנבכי הקוד ויבדקו כל שורה ושורה, יש להקפיד ביתר שאת על קוד נקי, מסודר, קריא ויעיל. בפרט:
2. הימנעו משכפול קוד
3. פונקציות ארוכות מדי (בדר"כ יותר מגודל עמוד)
4. בחירת שמות גרועים למחלקות, לפונקציות ולמשתנים
5. הזחה (אינדנטציה) נכונה
6. imports מיותרים
7. יש להקפיד להשתמש ב-modifiers בצורה נבונה:
8. מחלקה שלא אמורים לבנות אובייקטים שלה אמורה להיות מוגדרת כ-abstract
9. קבועים יש לסמן כ- final
10. משתנים של המחלקה רצוי להגדיר כ-private אלא אם יש סיבה לגיטימית לבחירה אחרת.
11. 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.
12. התמודדות עם קלט שאינו תקין (במקומות הרלבנטים) היא חלק בלתי נפרד מחווית המפתח (לטוב ולרע...).

יש לוודא קלט תקין מהמשתמש בכל שלב ולהחזיר הודאות שגיאה קריאות, אינפורמטיביות במידה והקלט אינו תקין. (למשל: לא להגיד שהקובץ לא תקין – אלא מה לא תקין בקובץ בצורה מפורטת...)

1. 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

1. 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. מטרת התרגיל היא לתרגל את הנושאים המרכזיים הנלמדים בקורס. על כן, בכל מקום שלא מופיעה דרישה מדויקת – מוטל עליכם לבחור בדרך ההגיונית 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)
* התרגיל מתקיים כולו במסגרת ג'אווה גרסה 8. הקפידו להוריד, לעבוד, לקמפל ולהריץ עם הגרסה המתאימה בלבד.
* **ווידוא הגשת התרגיל טרם הגשתו:**

1. **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 mac/linux – 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 Level 0 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 בונוס.**
  + **ישנם 2 סוגי בונוסים:**
    1. **בונוס בתוך טווח התרגיל - יכול להביא אתכם לכל היותר לציון 100, ולחפות במקרה והורדו לכם נקודות בשל תקלות.**
    2. **בונוס מחוץ לטווח התרגיל – יכול להעלות את ציונכם אף מעבר ל 100 (וכן, יש כפל מבצעים לטובת הלקוח).**
  + **בכל מקרה יש לבצע את הבונוס אם ורק אם סיימתם את כל דרישות הבסיס ההכרחיות לתרגיל.**
  + **חלק מהבונוסים 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 ושאפו "להקדים תרופה למכה", במידת האפשר. (וגם זה כלל חשוב לחיים, בלי קשר לתרגיל ולקורס).**

* + **פירוט הבונוסים, משקלם ונקודותיהם מפורט בגוף התרגיל הספציפי.**
  + **אם כבר מממשים בונוס, יש לממש את כולו, עפ"י דרישתו כדי לזכות במלוא הניקוד שהוא מקנה. בכל מקרה ההחלטה על ניקוד הבונוס היא בידי הבודק/מרצה בלבד (המגמה היא להיות נדיבים ככל האפשר...)**
  + **כאמור, ולמען הסר כל ספק – ניקוד הבונוס 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 הבונוסים הגעתם לציון כזה(.**

Sheet-Cell

## מוטיבציה

גיליונות אלקטרונים מלווים את חיינו [מאז שנות ה 70](https://medium.grid.is/theres-a-programming-language-that-rules-the-world-but-doesn-t-have-a-name-8d973b3cae20).  
הגיליון האלקטרוני הראשון, visicalc, נחשב עד היום ל killer app (הראשון..) אשר פתח וסלל את הדרך להכנסת והחדרת המחשב האישי לכל משק בית בעולם, אי אז בתחילת המהפכה הצרכנית של עולם מדעי המחשב בשנות ה 80.

המימושים הנפוצים ביותר והידועים לכולם של גיליונות אלקטרונים הם כמובן excel ו google sheets, אולם יש עוד רבים אחרים וכולם מספקים פחות או יותר את אותו סט היכולות הבסיסיות, ושונים זה מזה בקצוות ובפיצ'רים פרטניים.

לא הרבה מודעים לכך, אולם אפשר להסתכל על גיליונות אלקטרונים כעל סוג של ... [קוד](https://medium.grid.is/spreadsheets-are-programs-d3a6b0bd214).   
אפליקציות רבות שאנחנו יכולים לכתוב אותם בקוד – ניתנות למימוש בגיליונות אלקטרונים:  
חישבו על אקסל שבניתם לניהול הוצאות הבית עם השותפים לדירה, או לריכוז כלל רשימות המוזמנים לחתונה וכו'.  
(fun fact: ניתן להוכיח שאפשר לקודד בגיליון אלקטרוני מכונת טורינג ובכך להראות שהוא מהווה מכונת מחשוב שלמה, בדיוק כמו מחשב אלקטרוני רגיל.)

היתרון הגדול בגיליון אלקטרוני הוא שכמעט כל אחד יכול לעבוד עליו (חישבו על הוריכם..). לא נדרשת מידה רבה של תחכום או ידע בשביל לנהל גיליון אלקטרוני מורכב ומסובך ובכך למעשה הוא הופך להיות "שפת התכנות" הנפוצה ביותר בעולם.

אם אכן נסכים כי אפשר להסתכל על גיליון אלקטרוני כעל "פלטפורמה", אשר מאפשרת לאנשים רבים ללא כל ידע טכנולוגי לעבוד איתה ומעליה, לבצע בה דברים אשר המקבילה להם היא כתיבת מוצר תוכנה מורכב שדורש ידע ונסיון – הרי שמדובר על מודל תוכנה מעניין המאפשר לממש באמצעותו אפליקציות רבות – וזו סוגייה מעולה לחקור ולממש.

מהות תרגיל זה תהיה להתמקד בליצור מנוע גיליון אלקטרוני משלנו שיכיל מימוש בסיסי של גיליון אלקטרוני עם פיצ'רים מוכרים מכלל הפלטפורמות השונות המוכרות לכולנו.

למע' נקרא sheet-cell – הלחם של google sheet ו excel יחד עם המילה cell שהוא הרכיב הבסיסי ביותר בגיליון אלקטרוני.  
אולם עבורנו, דוברי העברית, מפתה ביותר לבצע שיכול אותיות ובמקום לקרוא לזה שיט-סל – פשוט לקרוא לזה... שטיסל !

המע' תבנה בשלושה חלקים (לאורך כל הקורס):  
**חלק א'** – מימוש שטיסל כמע' בסיסית ביותר. תפעול המע' יבוצע באמצעות ממשק console  
**חלק ב'** – הרחבת והעשרת שטיסל עם עוד יכולות מתקדמות יותר, ובעיקר יצירת ממשק גרפי מבוסס Java FX  
**חלק ג'** – הפיכת שטיסל לשרת המאפשר גישה מרחוק למספר משתמשים ולעריכה סמולטנית של קבצים בו זמנית.

## שטיסל – תיאור המע'

עקרה של שטיסל הוא לקיים את המופע של הגיליון האלקטרוני.  
הגיליון הוא מופע של טבלה דו מימדית שבה כל תא מיוצג ע"י מספר השורה (מספר שלם רץ, מתחיל מ 1) וזהות העמודה (אות באנגלית ב Capital Case).

### Cell

מייצג תא בודד בגיליון.   
ערכו של התא יכול להכיל את מי מהאפשרויות הבאות:

1. ערך מספרי: שלמים או ממשיים, חיוביים או שליליים.   
   ערכים מספריים יוצגו עם מפריד אלפים.  
   יש להציג ערכים ממשיים עם דיוק של 2 ספרות אחרי הנקודה; יש להציג ערכים שלמים ללא ספרות אחרי הנקודה
2. ערך מחרוזת: רצף של אותיות, סימנים ומכל הבא ליד.   
   יש להתעלם מתווים לבנים (whitespaces: רווח, טאב וכו') המופיעים בתחילת או סוף המחרוזת.  
   )אפשרי כמובן שהמחרוזת תכיל רווחים בתוכה.(
3. ערך בוליאני: המחרוזת TRUE או FALSE (באותיות גדולות)
4. פונקציה המקבלת מספר קלטים ומחזירה ערך כלשהוא.   
   הקלטים יכולים להיות ערכים מספריים או מחרוזות, או להכיל הפנייה לתאים/טווחים (range'ים) אחרים בגיליון.   
   כך למעשה ניצור מצב שבו נוכל לממש לוגיקה מעניינת במסגרת שטיסל כזו שתואמת (גם אם במשורה) לשאר הגיליונות האלקטרונים בעולם.

עקבו אחר [נספח א'](#appendix_B) כדי לקבל תיאור והסבר על כלל הפונקציות אשר ייתמכו במסגרת שטיסל.

בשל העובדה שתא יכול להכיל הפעלה של פונקציה, הרי שייתכנו הבדלים בין ערכו המקורי של התא (כפי שהגדיר המשתמש – למשל הכוונה לפונקציה כלשהיא) לבין ערכו בפועל שהוא הערך הסופי שהתקבל בעקבות הפעלת/הערכת הפונקציה.  
על כן לכל תא יוגדרו שני סוגים של ערכים:

* ערך מקור (Original Value) – זהו הערך של התא שהוכנס ע"י המשתמש (למשל תיאור הפונקציה)
* ערך סופי/אפקטיבי (Effective Value) – זהו הערך של התא אשר יוצג למשתמש הקצה ואשר ישמש תאים אחרים המסתמכים עליו

ייתכן בהחלט כי ערך המקור והערך האפקטיבי של התא יהיו זהים לחלוטין, וזאת במקרים בהם מדובר במספר או מחרוזת פשוטה. המקום בו הם נבדלים הוא רק אם התא מכיל הכוונה לפונקציה במסגרת ערך המקור.

מאחר ותא יכול להכיל פונקציה הנסמכת בתורה על תאים אחרים – הרי שלמעשה שינוי בתא מסוים יכול להשפיע על ערכו (האפקטיבי) של תא אחר, במידה והאחרון נסמך על הראשון, ומכאן למעשה הצורך לנהל את הקשרים בין התאים השונים והיכולת לזהות שינויים המשפיעים על תאים אחרים

### גליון אלקטורני

הגיליון האלקטרוני הוא למעשה המבנה אשר מגדיר את אוסף התאים השונים והקשרים שבינהם.  
לכל גיליון שכזה יהיה שם ייחודי והוא יכיל עוד מידעים הוליסטים על כלל הגיליון כמו רוחב העמודות, גובה השורות, הגדרות טווחים וכו'  
מבנה הנתונים העקרי שיושב בתוך הגיליון הוא למעשה אוסף התאים והקשרים בינהם.  
מבנה זה יתאר את העובדה ששינוי בתא מסוים צריך לפעפע ולהשליך על מצבם של תאים אחרים, והם בתורם ישפיעו על תאים אחרים וכו'.  
במסגרת ביצוע עדכון של תא בגיליון יש לבצע חישוב מחדש של כל ערכי התאים הרלוונטים בגיליון שכן הם יכולים להיות מושפעים ממנו.   
תהליך זה ייקרא **מ**הלך-**ח**ישוב-**מ**חדש (או בקיצור: **מח"מ**) של הגיליון.

במסגרת המח"מ, ייתכנו מספר בעיות, לדוגמא \*:

* יצירת הפניה מעגלית
* הפניה (במסגרת פונקציה) לתא שאינו מהסוג שהפונקציה מצפה לקבל
* הפנייה לתא שלא קיים במסגרת גבולות הגיליון

**\*** ייתכנו עוד בעיות שונות שתיתקלו בהם במסגרת הפיתוח  
במידה ונתקלים בבעיה שכזו, יש לדחות את השינוי שבוצע בגיליון (השינוי שהוביל למח"מ) ולדווח למשתמש את סיבת התקלה בצורה מפורשת ומובנת.

### גרסאות

שטיסל יאפשר לבצע מעקב גרסאות עדכון פנימי על כלל התאים שבגיליון.  
מספר הגרסה הוא מספר שלם, מתחיל מ 1.  
כל עדכון ידני שמתבצע במע' מקפיץ את מספר הגרסה ב 1.  
עבור כל תא יישמר מספר הגרסה האחרונה שהוא התעדכן בה.  
אם תא מסוים מתעדכן בעקבות מח"מ של הגיליון – הוא מקבל את הגרסה הנוכחית העדכנית של הגיליון.  
המע' תשמור ותאפשר למשתמש לחזור אחורה בזמן ולבחור להציג את מצב הגיליון בכל גרסה וגרסה.

### פונקציות

הפונקציות הנתמכות ע"י שטיסל הן שנותנות לו את מירב הכוח והיכולת לתאר במסגרתו מידע מורכב.  
לכל פונקציה יוגדרו הקלטים שהיא מקבלת, וכיצד היא פועלת עליהם.  
כדי להקל על המימוש, כל פעולה פשוטה תמומש גם היא כפונקציה (למשל פעולות החשבון הפשוטות של +,- וכו')  
במסגרת [נספח א'](#appendix_B) מתוארים כללי הכתיבה וההפעלה של הפונקציות השונות

### יכולות נוספות

שטיסל תתמוך במגוון יכולות נוספות (שיפותחו במסגרת כלל התרגילים):

* שליטה בעיצוב גרפי של התא
* סינון מידעים שיוצגו מתוך כלל הגיליון
* מיון הגיליון על פי ערכי עמודות
* [WHAT-IF](https://grid.is/?utm_firstpage=%2F&utm_seq=5) – היכולת לשנות את ערכו של תא מספרי בצורה דינמית ולראות את ההשפעה המיידית על כלל הגיליון
* שיתוף משתמשים בגיליון – מספר משתמשים יכולים לעבוד ברגע נתון על אותו הגיליון
* ניהול הרשאות גישה למשתמשים
* ועוד..

## הנחיות כלליות למימוש התרגיל

1. **המטרה היא לבנות מנוע מערכת גנרי, כזה שידע לקבל את הפרטים לגבי אופי המע' 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.**

1. **כחלק מהמע' תצטרכו לחשוב ולבחור לבד את מבני הנתונים השונים שישרתו את הצרכים של דרישות המע'.   
   זהו לא קורס במבני נתונים או באלגוריתמים, ומבני הנתונים/אלגוריתמים שתבחרו לממש לא חייבים להיות היעילים ביותר או האופטימליים. מספיק שהם יעבדו בצורה נכונה (ללא טעויות) ובזמן סביר.**
2. **ממשקי המשתמש השונים יפעלו מול מנוע המע' אשר יפותח מתרגיל לתרגיל בהתאם לדרישות.**
3. **כל הממשקים הפונים אל המשתמש החיצוני ומציגים לו מידע ו/או מבקשים ממנו מידע מספרי, חייבים להיות מבוססי ספירה המתחילה מ 1. גם אם פנימית אתם מממשים את מי מהרכיבים כמערך או רשימה (אז בסיס הספירה מתחיל מ 0) – עליכם להקפיד ולודא כי כלפי חוץ "תדברו" אך ורק במונחים של בסיס 1.**
4. **המע' כולה תתואר בשפה האנגלית בלבד, עם ממשק משתמש המתנהל משמאל לימין (במקומות הרלוונטים)**
5. **המע' כולה תיכתב ותורץ בסביבת העבודה של ג'אווה 21**

תרגיל 1 – מימוש **שטיסל** כאפליקציית Console (20%) - הגשה: 27.8.24

## פרטים יבשים

צפי תחילת עבודה: **7.8.24** תאריך הגשה: **27.8.24**

צפי זמן לביצוע: **3 שבועות** ציון אפשרי מקסימלי: 105

משקל התרגיל: **20%** קושי: **סביר**

**מטרת התרגיל העיקרית**

1. הקמת מנוע המע' הבסיסי
2. יצירת ממשק console לתפעול המערכת

## דרישות

1. בתרגיל זה תקימו את תשתית העבודה הראשונית והבסיסית של שטיסל.   
   את התשתית "תתפעלו" באמצעות ממשק console פשוט המציג תפריט פקודות שדרכו מפעילים את המע'.  
   בתרגיל תממשו מספר סוגים של פונקציות (כפי שמוגדר [בנספח א'](#appendix_B)), תגדירו את היישויות העקריות במע' (גליון ותא) ותאפשרו תפעול בסיסי של הגליון.  
   כל אלה יישבו בתוך "מנוע" המערכת, אשר ידע להגיב לכל פנייה המגיעה משכבת ממשק המשתמש, לעבד את הקלט ולהחזיר פלט רלבנטי.
2. יש לוודא תקינות קלט כחלק מכל אינטרקציה עם המשתמש, ובכל מקום שבו זה רלוונטי:

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 מישהו שמגיע מהתחום ו"מכיר" איך דברים עובדים לבד.

1. **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** לנקות את המסך בין פקודה לפקודה.

1. עליכם לכתוב ממשק משתמש בתצורת console.

ממשק המשתמש יכיל סט סופי של פקודות שדרכן ניתן יהיה להפעיל את המערכת.

אחרי הצגת תפריט הפקודות יש לחכות לקלט מהמשתמש באשר לפעולה אותה הוא רוצה לבצע. לאחר ביצוע הפעולה (שאולי 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 להן פקודות אחרות. במידה וזה קורה יש להציג הודעת שגיאה רלוונטית למשתמש ולאפשר את המשך פעילות המע'.
* בכל המקומות שבהם מציגים "רשימות של דברים" וצריך לאפשר למשתמש לבחור פריט(ים) מרשימה – יש לאפשר בחירה זו ע"י הצמדת מספר לכל אחת מהאפשרויות ולאפשר לו לבחור על פי המספר המזהה של האפשרות מהרשימה (או באמצעות כמה מספרים במקומות הרלוונטיים). המספרים יתחילו מ 1 (לא מ 0)  
  **אין להניח** שהמשתמש הולך להקליד לכם מלל חופשי של תיאור האפשרות !

להלן רשימת הפקודות שיש לתמוך בהן:

1. קריאת קובץ פרטי המע'

פקודה זו טוענת את פרטיו הראשוניים של גליון מסוים 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).   
הנתיב יכיל רק אותיות באנגלית (לא ג'יבריש של אותיות בעברית וכו')

הקובץ יהיה קובץ XML שפרטיו וחוקיו המפורטים מובאים [בנספח ב'](#appendix_C) לתרגיל זה.   
אתם מצופים לעבור על פרטים אלה ולהתייחס אליהם כחלק אינטגרלי מהגדרת התרגיל.

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...)

בפרט יש לוודא את הפרטים הבאים:

1. הקובץ קיים, והוא מסוג XML (די לבדוק לשם כך כי הוא נגמר בסיומת .xml)
2. גודל הגליון בשורות הוא מספר שלם בין 1 ל 50 ובעמודות הוא מספר שלם בין 1 ל 20
3. במסגרת הגדרת התאים, אין תא שמיקומו מוגדר להיות מחוץ לגבולות הגליון
4. התאים המגדירים שימוש בפונקציות מכווינים לתאים המכילים מידע שמתאים לארגומנטים של הפונקציה

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 שהקובץ נמצא תקין ונטען במלואו למשתמש.  
טעינה תקינה של הקובץ גם מגדירה את מצבו בגרסה 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 נסיון טעינה של קובץ תקול לא דורס את פרטי הקובץ (התקין) האחרון שהיה במע' (ככל שהיה כזה)

**דגשים:**

* + - פקודה זו מוצגת ומאופשרת תמיד. אפשר לבחור בה בכל רגע נתון במע'.

1. הצגת הגליון  
   פקודה זו תציג למשתמש את המצב הנוכחי של הגליון.  
   יש להציג את הפרטים הבאים:

* מספר הגרסה הנוכחי שהגליון מצוי בו
* שם הגליון
* עליכם להדפיס את התאים השונים על פי סדר השורות וסדר העמודות, משמאל לימין ומלמעלה למטה (כמו כל גליון אלקטרוני לועזי רגיל שאתם מכירים)

בשורה העליונה יודפסו אותיות אנגליות גדולות לסמן את שמות העמודות; בעמודה השמאלית ביותר יודפסו מספרי השורות. כל מספר יודפס באמצעות שתי ספרות (למשל 04, 34 וכו')

כמות השורות וכמות העמודות יהיו תואמים לגודל הגליון כפי שזה מופיע בפרטי הקובץ.

* בין כל עמודה לעמודה יופיע תו ה | (pipe) אשר יסמן הפרדה בין העמודות
* אין לסמן קווי הפרדה בין השורות.
* העובי של כל עמודה יהיה כמות התווים שהוגדרה בקובץ הגליון
* רוחב השורה יהיה על פי הכמות שהוגדר בקובץ הגליון
* בכל תא יוצג הערך האפקטיבי שלו.
* לא כל התאים בעלי ערך – ועל כן תאים ריקים – לא יכילו שום תו (אפקטיבית - רווחים)

1. הצגת ערכו של תא בודד

פקודה זו תציג את ערכו ומצבו של תא בודד  
יש לבקש מהמשתמש את זהות התא המדובר.   
המשתמש יכניס את אות העמודה (כאות גדולה) ואת מספר השורה כשהם צמודים. דוגמא: 4A  
לאחר מכן יש להציג למשתמש את הפרטים הבאים:

* זהות התא (הנתון שהוא הכניס עתה)
* הערך המקורי (original value) של התא
* הערך האפקטיבי (effective value) של התא
* הגרסה האחרונה שבה הוא שונה
* רשימת זהות התאים שהוא תלוי בהם (קרבה ראשונה). במידה והרשימה ריקה - יש לציין זאת
* רשימת זהות התאים שהוא משפיע עליהם (קרבה ראשונה). במידה והרשימה ריקה - יש לציין זאת

1. עדכון ערכו של תא בודד  
   פקודה זו תאפשר למשתמש לבחור תא מסוים ולעדכן את ערכו.  
   יש לבקש מהמשתמש את זהות התא המדובר.  
   המשתמש יכניס את אות העמודה (כאות גדולה) ואת מספר השורה כשהם צמודים. דוגמא: 4A  
   לאחר מכן יש להציג למשתמש את הפרטים הבאים:

* זהות התא (הנתון שהוא הכניס עתה)
* הערך המקורי (original value) של התא
* הערך האפקטיבי (effective value) של התא

יש לבקש מהמשתמש את ערכו החדש של התא, יהיה אשר יהיה.   
עבור איפוס ומחיקת התא יש לאפשר לקלוט גם מחרוזת ריקה.  
יש להחיל את הערך החדש בגליון ולבצע את המח"מ.  
  
כאמור, ייתכן כי תו"כ המח"מ יתגלה כי הערך החדש יוצר בעיה בעדכון התאים הנסמכים עליו (ישירות או בעקיפין).  
במידה ומתגלה תקלה שכזו, ו/או הערכים אינם חוקיים (למשל פונקציה שאינה מוכרת או לא בתוך {}; הפניה לתא שלא קיים מחוץ לגבולות הגליון וכו') - יש לזהות זאת, להודיע למשתמש בצורה מסודרת את סיבת ומהות התקלה, ולאפשר לו להכניס את הערך שנית.  
  
  
אחרי מח"מ מוצלח יש לעדכן את הגרסה של הגליון ולהקפיץ אותה ב 1, כמו גם לעדכן בגרסה זו את גרסת העדכון של כל התאים שהושפעו מהמח"מ.  
  
בסיום פקודה זו יש להציג את מצבו הסופי של הגליון כפי שהוא מוגדר בפקודה 2.

1. הצגת גרסאות

פקודה זו תאפשר את הצגת הגרסאות השונות של המע'.  
ברגע שהמשתמש בוחר בה הוא יקבל טבלה המציגה את מספר הגרסה, החל מ 1 (בראש) וליד כל מספר גרסה את כמות התאים שהשתנו בה. גרסה 1 היא סך כל התאים שיש בהם מידע שנטענו מהקובץ.  
אפשרו למשתמש להכניס מספר גרסה רצויה "להצצה" אליה.  
במידה והוא הכניס מספר גרסה לא תקין, יש כמובן לזהות זאת ולהודיע לו במסודר ולאפשר לו לבחור מחדש.  
בהכנסת מספר גרסה תקין יש להציג את מצבו של הגליון כפי שהיה בגרסה זו.  
הדפסת הגליון תעשה באמצעות ועל פי ההוראות בפקודה מס' 2.  
  
**שימו לב**: המערכת רק מציגה את מצב הגליון בהתאם לגרסה הנבחרת. היא לא משנה את מצב הגליון הנוכחי בגרסה הנוכחית !  
אם המשתמש יבחר לבצע את פקודה 4 (המשנה ערכו של תא/גליון) – זה יקרה כמובן בהקשר הגרסה העדכנית ביותר

1. יציאה מהמערכת

פקודה זו מסיימת את פעולת התוכנית.

## 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 זהו 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'.

* + 1. 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...)

התחילו מהבנת מבנה המע'. אני ממליץ להגות על נייר (אמיתי, כמו פעם..) מבנה כללי של המחלקות העיקריות, הקשרים והיחסים בינהן.

צרו פרויקט חדש ב Intellij אשר ישמש כפרויקט האב לכלל התרגילים.   
בתוך הפרויקט צרו מודול נפרד עבור מנוע המע'. המנוע יכיל את כלל החלקים הנדרשים לתפעול המע' (הגדרה של תא, גליון, פונקציות וכו').  
המנוע יחשוף סט של יכולות (הפקודות השונות בתפריט), יחזיק מופע של המע' ויתווך את הפקודות השונות מולה, הלוך וחזור. התחילו בבניית תשתית המנוע (מומלץ לעטוף את המנוע בממשק ולהתחיל לתרגל (ולהתרגל) לחוויית העבודה הנ"ל.)  
  
התחילו בבניית האובייקטים העקרים (תא, גליון, פונקציות, מנגנון ההרצה). תוכלו להתחיל להרים את המע' גם לפני קריאת הנתונים מקובץ ולוודא כי היא מתחילה לעבוד. ממשו ראשונה דווקא את פקודה 2 שתאפשר לכם לראות כי בידכם המידע הנדרש ואתם מציגים אותו כראוי.  
בהתאם להתקדמות, המשיכו במימוש פקודה 1 (על כלל היביטי הטעינה), ו/או עיברו לממש את פקודות עדכון התאים ותהליך המח"מ בעקבותיהם.  
אני ממליץ רק בסוף התהליך לעבוד על כל סוגיית ניהול הגרסאות ופקודה 5.

לסיום, צרו מודול נוסף שהוא מודול שכבת ה UI (console). זה המודול שבו תשב בסופו של דבר מטודת ה main הראשית שתתפעל את כלל המע'. זה המקום היחיד שבו מוצג פלט (System.out.pritnln) ונאסף קלט (scanner) מהמשתמש. זה המקום המכיל את לולאת התפריט הראשית המניעה את כלל המע'. שכבת ה UI תכיל הפנייה (reference) למופע המנוע (שבתוכו מכיל הפנייה למופע המע') וכך תוכל להעביר ולתרגם לו את הפקודות הנאספות מהמשתמש ולהציג חזרה את הפלטים החוזרים מהמנוע.

## בונוסים

|  |  |  |  |
| --- | --- | --- | --- |
| # | מהות | למה שווה לי ? | כמה שווה לי ? |
| 1 | **שמירה וטעינה של המע'**  בבונוס זה המצב הנוכחי של המע' (כולל כלל הגרסאות של הגליון הנוכחי) נשמר לקובץ חיצוני (באיזה פורמט וטכניקה שתבחרו).   יש להוסיף פקודה שמאפשרת לשמור את מצב המע' וגם פקודה המאפשרת לטעון מצב קיים (מקובץ שנשמר זה לא מכבר), וזאת בניגוד לטעינה רגילה מקובץ ה XML של התרגיל.  יש לאפשר למשתמש לבחור את הנתיב המלא כולל שם הקובץ (בלי הסיומת) שהוא היה מעוניין לשמור את המע' אליו (ולטעון אותה ממנו) | כי עם תכנון נכון זה אמור להיות משהו כמו 4-5 שורות... | 5 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').

1. קבצי קוד המקור של הפרויקט שלכם.
2. קובץ ההגשה יכיל גם קובץ **readme** שיכיל הסבר על המערכת, בחירותיכם השונות במקומות שבהם היו לכם בחירה, או כל דבר נוסף העולה על דעתכם שחשוב שהבודק ידע.
3. יש לכלול בקובץ ה **readme** גם תיעוד והסבר כללי (וממצה) של המחלקות העיקריות ותפקידם.
4. יש לכלול בקובץ ה **readme** גם פירוט של המגישים שם, ת.ז. ואי מייל זמין ורלבנטי (!!) – במידה ויהיה צורך ליצור קשר.
5. במידה ומימשתם את מי מהבונוס(ים) לעיל, ציינו את שם הבונוס שמימשתם בראשית קובץ ה readme כדי שהבודק ידע לבדוק בהתאם.

בונוס שימומש אבל לא יתועד – לא ייבדק !

הגשה באיחור, שאינה באישור, תבטל כל מימוש בונוס. אין להגיש באיחור בשביל להספיק לעשות בונוסים.   
תכננו את הזמן בהתאם.

תרגיל 2 – מימוש **שטיסל** כאפליקציית JavaFX ( 40%) – הגשה: 28.9.24

## פרטים יבשים

צפי תחילת עבודה: **29.8.24** תאריך הגשה: **28.9.24**

צפי זמן לביצוע: 3 **שבועות +** ציון אפשרי מקסימלי: **128**

משקל התרגיל: **40%** קושי: **מאתגר**

**מטרות התרגיל העיקרית**

1. מימוש תצוגת ותפעול המע' כאפליצקיה גרפית JavaFX
2. הוספת תמיכה בטווח (range) ותמיכה בפונקציות מורכבות
3. הוספת יכולת מיון וסינון
4. הוספת יכולת ניתוח דינמי

## דרישות

1. בתרגיל זה תממשו שכבת משתמש גרפית דרכה המשתמש יוכל לתפעל את המע' ולצרוך את המידע.

השכבה הגרפית תציג את שטיסל בצורה "נורמלית", כמו כל שאר הגליונות האלקטרונים המוכרים לכם, וכ-כזו תתמוך במספר היבטים עיצוביים של הגליון: יישור, צבע רקע, פונט וכו'.   
למגדילי הראש שמור עונג מיוחד: הוספת יכולת יצירת גרפים ויזואלים מהמידע שבגליון  
לבד מזו יתווספו יכולות תפעול של מיון וסינון של המידע על פי קרטריונים שונים.  
נרחיב ונוסיף פונקציות חדשות מורכבות יותר, בין היתר כאלה הנסמכות על עבודה על טווח של תאים. (עיקבו אחר [נספח א'](#appendix_B) לקבלת פרטים על הפונקציות המדוברות)

יכולת נוספת תהיה ביצוע שינויים דינמיים על גבי הגליון (what-if-scenario)

לטובת הכנת האפליקציה סופק לכם קובץ סקיצה גרפי המציג בקווים כלליים ביותר את מבנה המע' מבחינת מיקומי הרכיבים הגרפים. יש להיצמד אל קווי המתאר הכללים המתוארים בקובץ זה.

1. המשתמש יבחר קובץ גליון ראשוני להעלותו.   
   הקבצים שיועלו יהיו רק קבצים מהפורמט של תרגיל 2. עיקבו אחר [נספח ב'](#appendix_C) כדי לראות את ההבדלים בין הסכמות השונות.

כשם שהיה בתרגיל 1, יש לאפשר לטעון כמה קבצים אחד אחרי השני. כל קובץ שנטען בהצלחה מחליף את קודמו. ברגע נתון יש (לכל היותר) רק קובץ אחד הטעון במע'.  
טעינת קובץ ה xml תתבצע באמצעות file chooser dialog בלבד ! (אין להניח שהקובץ יחכה לכם בספרייה ייעודית, או שהבודק יקליד את תוכנו לתוך תיבת טקסט וכו'.).  
הקובץ יכול להימצא בכל ספרייה חוקית על המחשב (כולל ספריה עם רווחים). **נא לוודא כי הדבר אינו מכשיל אתכם**

טעינת הקובץ תתבצע באמצעות Task של JFX. יש להציג progress bar או מכוון התקדמות כרצונכם לתהליך הטעינה.  
תהליך הטעינה עצמו, טכנית, הוא קצר. על כן הוסיפו השהיה קלה (ברמה של שנייה או שתיים) כדי לסמלץ את ההתקדמות.  
  
עליכם לבצע בדיקות תקינות קלט של הקובץ כפי שבצעתם בתרגיל 1.  
בנוסף עליכם לוודא את הדברים הבאים:  
1. אין הגדרות range'ים בעלות שם זהה

2. אין הגדרת range הכוללת הכוונה לתא שאינו במסגרת גבולות הגליון  
3. אין הכוונה ל range שלא קיים מן הפונקציות שונות  
  
אם הקובץ אינו תקין אין לטעון אותו למע' ויש להחזיר הודעה מפורטת למשתמש מדוע אינו תקין.  
אם הקובץ תקין הוא נטען למע' בהצלחה וניתן לצפות בפרטיו ולהפעילו בהתאם.

1. הצגה ויזואלית של הגליון:  
   חלקו המרכזי של המסך יציג את הגליון האלקטרוני במבנה המוכר לכולנו:  
   כותרות העמודות הנתמכות למעלה; מספרי השורות בצד שמאל.  
   בכל תא יוצג ערכו האפקטיבי.   
   ערך בוליאני יוצג כמילה TRUE או FALSE באותיות גדולות בלבד.  
   כברירת מחדל, הערכים בכל תא ימורכזו אופקית ואנכית.

שימו לב כי ייתכן עכשיו כי גודל הגליון יחרוג משטח המסך האפשרי ועל כן יש כמובן להציג scroller'ים (אופקי ואנכי) כדי לאפשר לגלול את סך שטח הגליון.  
  
הנה מספר יכולות שעליכם לתמוך בהם בהיביט הויזואלי:

1. אפשרו להגדיר את רוחב העמודות/שורות דרך הממשק הגרפי.  
   במידה והמידע להצגה גדול מרוחב העמודה שנקבע – ביכולתכם לקבוע את אופן ההתנהלות:   
   עדיפות היא לביצוע wrap, אבל אפשר גם לחתוך (clip).  
   בחירת האפשרות היא ברמת העמודה והיא תחול על כלל המידעים המוצגים בה.
2. אפשרו לבחור את היישור (alignment) של כל המידעים המוצגים במסגרת עמודה.  
   היישור יכול להיות לשמאל, לימין או למרכז.  
   בחירת האפשרות היא ברמת העמודה והיא תחול על כלל המידעים המוצגים בה.
3. אפשרו למשתמש לבחור את העיצוב של תא מסוים:  
   יש לאפשר לבחור את צבע הרקע ואת צבע הטקסט.  
   בחירת הצבע תעשה מתוך פלטת צבעים עם color picker מסודר (אין לממש זאת בעצמכם !)  
   אפשרו כפתור ייעודי שמבטל את כל העיצוב שבתא ומחזיר אותו לקדמותו.  
   העיצוב מקושר לתא וצריך להישמר ולזוז איתו במסגרת המיון/סינון שיתבצע בטבלה.
4. תמיכה ב range ופונקציות מורכבות  
   יש לתמוך בהגדרת range המתאר מקבץ של תאים.  
   אפשר יהיה להשתמש בהגדרת ה range במסגרת הפונקציות המורכבות המוגדרות במע'.  
   לכל range יוענק שם ייחודי שיאפשר להצביע אליו ולהתייחס אליו.  
   הגדרת ה range תכיל אך ורק תאים צמודים אחד לשני בעמודה, בשורה או בטווח דו מימדי, והיא תתבצע בצורה הבאה:  
   <top left cell id>..<right bottom cell id>  
   דוגמא:

* A1..A4 – מגדיר טווח תאים בעמודה A
* A3..D3 – מגדיר טווח תאים בשורה 3
* A3..D4 – מגדיר טווח תאים דו מימדי בין עמודה A ל D ובין שורות 3 ל 4

את כל ה range'ים יש לנהל באזור מוגדר במע' (על פי הסקיצה).  
הפעולות שיש לתמוך בהן הן:

* הוספת range חדש. במסגרתו יש לבקש את השם הייחודי (הודעת שגיאה מתאימה אם בוחרים שם שכבר קיים) ואת טווח ה range.  
  שימו לב: ניתן להגדיר range'ים חופפים ל range'ים אחרים
* מחיקת range קיים:  
  יש לאפשר למחוק הגדרת range קיים מהמע'.   
  במידה ויש שימוש ב range זה במסגרת מי מהפונקציות השונות יש להתריע בפני המשתמש ולמנוע את מחיקתו.
* צפייה ב range קיים:  
  בחירת range מסוים תדגיש את התאים הנכללים במסגרתו (בדרך מסוימת הנתונה לבחירתכם).

1. מידע ועדכון תאים:  
   אפשרו למשתמש לקבל את מלוא המידע עבור כל תא נבחר.  
   האפליקציה תכיל "שורת עבודה" שבמסגרתה יוצגו המידעים השונים של תא נבחר.  
   בחירת התא תעשה ע"י הקלקה עליו (כמו בכל אפליקציות גליון אלקטרוני מוכרת)  
   בבחירת תא מסוים בגליון יש להציג את כלל המידעים שהצגתם במסגרת פקודה 3 בתרגיל 1, בצורה הבאה:

* בשורת העבודה יש להציג את הפרטים הבאים:
  + זהות התא (דוגמא: 5A). יש להציג באות גדולה בלבד.
  + ערכו המקורי של התא
  + גרסה האחרונה שבה הוא עודכן
* יש להדגיש ברקע כחול בהיר את כל התאים שתא זה נסמך עליהם (באם ישנם כאלה)
* יש להדגיש ברקע ירוק בהיר את כל התאים שתא זה משפיע עליהם

עדכון המידע בתוך התא יתאפשר בקלות ע"י עדכונו בשורת המידע.   
מייד לאחר ביצוע העדכון (באמצעות כפתור מיוחד) יתבצע המח"מ ותוצאתו תשוקף במידע הגליון.  
במידה והעדכון הוא כזה שמוליד מצב לא תקין מבחינת המח"מ יש להציג למשתמש הודעת שגיאה מתאימה וכמובן לא לקבל את התיקון, לא להעלות גרסה וכו'.

1. מיון השורות:  
   הציעו למשתמש אפשרות למיין את המידע שבטבלה.  
   המיון יתבצע כמובן על פי הערך האפקטיבי של כל תא.  
   המיון לא מייצר "גרסה" חדשה של הטבלה כי אם רק מעניק לה view אחר. (אפשר להציג את המיון בחלון עצמאי -popup)  
   במהלך ביצוע המיון יש להשאיר את מספרי השורות בסדר עוקב (כלומר המיון לא משפיע על מספרי השורות)  
   כדי להקל, ניתן יהיה למיין על פי ערכים מספריים בלבד. המיון יתבצע על פי הסדר הטבעי של האברים, מהנמוך אל הגבוה. (אין צורך לתמוך במיון הפוך).  
     
   כדי למיין יש לבקש מהמשתמש מספר מידעים:

* מהו אזור הטבלה שנדרשת למיון. יש להגדיר את האזור בדומה להגדרת טווחים, בצורה הבאה:  
  <top-left-cell>..<bottom-right-cell>   
  דוגמא: A3..V9  
  על האזור להכיל את השורות המכילות בין היתר את המידע המספרי שרוצים למיין על פיו. יכולות להיכלל בו עוד עמודות שמכילות מחרוזות או מידע בולאיני – אבל המיון יתבצע אך ורק על פי העמודות המספריות.  
  אם יש לטבלה שורה של כותרות לעמודות – אין לבחור אותה במסגרת שטח המיון.
* על פי אילו עמודות יש לבצע את המיון:  
  יש לבחור את אותיות העמודות לביצוע המיון על פיהן.   
  שימו לב: יש חשיבות לסדר העמודות: העמודה הראשונה תהווה את המיון הראשוני. במידה ויש ערכים זהים בעמודה זו, הן ימוינו על פי העמודה הבאה בתור וכן הלאה.  
  אפשר לבחור למיון את כל העמודות בטווח האזור שנבחר.

מרגע שבוצעה הבחירה, המיון יתבצע על פי העמודה/ות שנבחרה, כך ששורות המידע (בטווח שנבחר) ימוינו בהתאם.  
(המיון ישפיע כמובן רק על התאים בטווח שנבחר)  
במידה ויש ערכים זהים בשורות אחרי המיון על פי כל העמודות שנבחרו לכך – יש להשאיר את המידע כפי שהוא היה (מיון יציב).

אפשרו למשתמש כפתור שמבטל/סוגר את המיון ומחזיר את הגליון למצבו המקורי.

1. סינון המידעים (filtering):  
   הציעו למשתמש אפשרות לסנן את המידע שבטבלה.  
   הסינון יתבצע ע"י בחירת עמודה אחת בודדת ובחירה להציג רק את הערכים הרלוונטים שנמצאים בה.  
   בבחירת הערך הרלוונטי יוצגו רק השורות שהערך הזה נמצא בהן.  
   במהלך ביצוע הסינון יש להשאיר את מספרי השורות בסדר עוקב (כלומר הסינון לא משפיע על מספרי השורות)  
   אפשר להציג את הסינון בחלון עצמאי (popup)  
     
   כדי לבצע את הסינון יש לבקש מהמשתמש את הערכים הבאים:

* מהו אזור הטבלה שנדרשת לסינון. יש להגדיר את האזור בצורה הבאה  
  <top-left-cell>..<bottom-right-cell>   
  דוגמא: A3..V9  
  על האזור להכיל רק את המידע שרוצים לסנן. אם יש לטבלה שורה של כותרות לעמודות והן לא הולכות להיות חלק מהסינון – אין לבחור אותה במסגרת שטח הסינון.
* על פי איזו עמודה יש לבצע את הסינון:  
  יש לבחור את אות העמודה לביצוע הסינון.
* אילו ערכים להציג:  
  לאחר שנבחרה העמודה הנדרשת, על המע' לאתר את כל הערכים הייחודיים בעמודה זו ולהציג אותם למשתמש.  
  המשתמש יוכל לבחור ערך אחד או יותר שאותו הוא רוצה לראות.

בגמר בחירת המידעים, יוצגו רק השורות באזור המוגדר המכילות את הערכים שנבחרו.

אפשרו למשתמש כפתור המבטל/סוגר את הסינון ומחזיר את הגליון למצבו המקורי.

1. ניהול גרסאות:  
   הציגו את המידע של הגרסה הנוכחית של הגליון במקום ברור.  
   אפשרו למשתמש לבחור להציג את הגליון בכל גרסת עבר. בחירת הגרסה תציג את הגליון בהתאם למצבו בכל גרסה.  
   במסגרת הצגת הגליון בגרסת עבר אפשר להתעלם מכל ההיביטים העיצוביים שהיו בגרסה זו ולהציג את הגליון בצורתו הגולמית.
2. 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.**

1. תלונות מלקוחות:  
   אחרי הגרסה הראשונה של המע' התקבלו תלונות רבות מלקוחות על כך שהמודל התמודדות עם קלטים שגויים הגביל מאוד את אופן השימוש במע'.  
   ההנהלה החליט לממש מודל מקל יותר על קליטת ועיבוד מידע מהמשתמש שיש לתמוך בו לאורך כל הגליון.  
   הנה עיקריו:

* קלט של פונקציה ייחשב לא תקין רק אם מדובר על מבנה לא נכון: שם פונקציה לא מוכר; כמות ארגומנטים לא נכונה וכו').
* קלט של ארגומנטים מסוג לא נכון לפונקציה, בין אם ישירות (לדוגמא: concat(4,5)) ובין אם בעקיפין (לדוגמא: ב a1 יש מספר 4 ואז מבצעים concat(ref(a1),aaa)) – ייחשב קלט תקין (שלא עוצר מח"מ) אולם אז הפונקציה תקבל ערך פסול (על פי סוגה)
* בתא ריק אין ערך. ביצוע ref לתא ריק הוא חסר משמעות, אולם טכנית הוא קלט חוקי ותקין. ערכו של התא שעושה ref לתא ריק הוא... התא הריק.
* כל הפונקציות שעובדות על range'ים יהיו אופטימיסטיות נצחיות, כלומר יתעלמו מערכים פסולים או תאים ריקים במסגרתם.
* בכל מקום שבו נדרש להכניס זהות של תא (הגדרת range; ארגומנט ל ref) הוא יכול להינתן גם באות קטנה.  
  בכל מקום שבו נדרש להציג זהות של תא (הגדרת range; ארגומנט ל ref) הוא יוצג תמיד באות גדולה
* המחרוזת true ו false (case insensitive) מוקצות לשימוש כמחרוזות. הן ייצגו את הערכים הבוליאנים בלבד.  
  בכל מקום שבו נדרש להכניס קלט בוליאני ע"י המשתמש הוא יכול להינתן case insensitive. בכל מקום שבו מציגים תוכן בוליאני למשתמש הוא יוצג תמיד כ TRUE או FALSE באותיות גדולות  
  (למתחכמים שבינכם: אין צורך לתמוך, לחשוב או להתייחס לקומבינה של concat(tr,ue))

אין שום צורך לתמוך ביכולות ובקבצי העבר של תרגיל 1.   
אפשר להרגיש חופשי לשנות את המנוע ואת המחלקות שבו כך שיתאימו מעכשיו רק לדרישות ולצרכים של תרגיל 2.

**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).**

## 

## איך מתחילים ?

התחילו בהקמת השלד הראשי של המע' ובהתאם לקווי המתאר של האפליקציה.   
  
המשיכו ביכולת לבצע טעינה של הקובץ שתתחבר למימוש הקיים בצד המנוע.  
התרכזו ביצירת הרכיב הדינמי שמציג את הגליון עצמו. אין ספק שיש כאן אתגר מסוים אולם ברגע שתצלחו אותו – סלולה לכם הדרך אל האושר !  
לאחר שביכולתכם להציג את הטבלה בצורה מסודרת, התמקדו בשורת ההפעלה ובהצגת המידעים השונים בהינתן שתא מסוים נבחר.  
בגמר שלב זה יש לכם גליון אלקטרוני פעיל ! סחטיין !

זה הזמן להתחיל לעבור ולהתרכז במימוש הפקודות והפיצ'רים השונים של האפליקציה. אני ממליץ לבחור פיצ'ר מסויי (למשל מיון, סינון, דינמי) ולממש אותו "עד הסוף": החל מהצגתו ב UI, דרך תפעלו בצד המנוע, השלכותיו חזרה על ה UI (באם רלוונטי) וכו'.

## בונוסים

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

|  |  |  |  |
| --- | --- | --- | --- |
| # | מהות | למה שווה לי ? | כמה שווה לי ? |
| 1 | אפשרות החלפת skin למערכת. בבונוס זה תממשו (לפחות) 2 ערכות צבעים נוספות על הערכה הדיפולטית, ותאפשרו למשתמש להחליף את ערכות הצבעים. שימו לב ערכת הצבעים כוללת (לכל הפחות):   1. החלפת רקע של כל המסך 2. החלפת המראה של הכפתורים על המסך 3. 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)** |
| 2 | ממשו 2-3 אנימציות המלוות את תהליך ההפעלה  האנימציה צריכה להתבצע לכל היותר במשך 2 שניות  יש לאפשר גם לנטרל את האנימציות הנ"ל, כך שהיא לא תאיט את התקדמות המערכת. | כל כך קל... | 5 נקודות  (ועד ל 100) |
| 3 | סינון של מספר עמודות אפשרו למשתמש לבצע סינון של מידעים באמצעות יותר מעמודה אחת. יש לאפשר למשתמש לבחור כמה עמודות ובכל אחת מהן את הערכים שהוא רוצה לראות. הסינון יתבצע באמצעות ביצוע "וגם" בין הערכים הנבחרים. בגמר הבחירה התוצאה בטבלה תשקף את המידע שנבחר. | כבר יש לכם יכולת לבחור סינון על עמודה אחת.. מה זה כבר לאפשר כמה עמודות ?? | 5 נקודות **(מעל ל 100)** |
| 4 | ניתוח דינמי: ממשו כבר עכשיו את פיצ'ר הניתוח הדינמי (תרגיל 3 סעיף 10). תוכלו לחסוך לכם עבודה להמשך ובנוסף לקבל עליו בונוס במסגרת תרגיל זה ! | הקדמת תרופה למכה... | 5 נקודות **(מעל ל 100)** |
| 5 | גרפים: אפשרו למשתמש לייצר גרפים מהמידע המספרי שיש לו בגליון ! אפשר להשתמש ביכולות ה built in של java fx לצייר גרפים (ויש הרבה כאלה..) או להשתמש בכל ספריה חיצונית העולה על רוחכם...  יש לתמוך בגרפים של עמודות ושל קווים. הגרפים יהיו פשוטים של X (ציר אופקי) מול Y (ציר אנכי).  במסגרת הגדרת הגרף יש לאפשר למשתמש לבחור את טווח התאים מעמודה שתשמש כציר ה X ואת טווח התאים מעמודה שתשמש את ציר ה Y | אחד הדברים המגניבים והמתבקשים בגליון אלקטרוני ! | 8 נקודות **(מעל ל ל100)** |
| 6 | בניית נוסחה אינטרקטיבית הציעו ממשק גרפי מלא, שבמסגרתו המשתמש יוכל לבנות ביטוי מורכב של פונקציה.  במסגרת הממשק המשתמש יוכל לבחור את שם הפונקציה (ולא לכתוב אותה), להציע את הארגומנטים השונים (שיכולים להיות פונקציות וביטויים מורכבים בעצמם) ולקבל הצצה מהירה לערך החישוב של כל ביטוי לאורך הדרך | למשוגעים לדבר... | 10 נקודות **(מעל ל 100)** |

## סבבה, סיימתי. מה ואיך להגיש ?

יש להגיש קובץ zip המכיל:

1. jar (אחד או יותר) שהוא 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 לבודק להיכנס ביתר קלות לקוד שלכם ולהבין מי נגד מי...)
5. במידה ומימשתם את מי מהבונוסים לעיל, ציינו את שם הבונוס בראשית קובץ ה readme כדי שהבודק ידע לבדוק בהתאם.

בונוס שימומש אבל לא יתועד – לא ייבדק !

תרגיל 3 – מימוש **שטיסל** כאפליקציית Client - Server (35%) – הגשה: 9.11.24

## פרטים יבשים

צפי תחילת עבודה: **29.9.24** תאריך הגשה: **9.11.24**

צפי זמן לביצוע: **4 שבועות +** ציון אפשרי מקסימלי: **107**

משקל התרגיל: 35**%** קושי: **ווא'עליה אלל'בחטי**...

**מטרות התרגיל העיקרית**

1. מימוש המערכת כאפליקציית client-server.
2. הוספת מנגנון הקצאות ומשתמשים

## דרישות

1. בתרגיל זה נממש את היכולת לתפעל את שטיסל כ-שרת המאחסן מספר גליונות ממשתמשים שונים.  
   המשתמשים יוכלו לאפשר גישה לגליונותיהם למשתמשים אחרים, אם לתצוגתם בלבד ואם לעריכתם.  
   במידה ולכמה משתמשים יש יכולות עריכה – הם יוכלו לערוך את הגליון בו זמנית (בדומה ל google sheets) ולראות את השינויים שכל אחד מהם מייצר במע'.
2. השרת (tomcat) יכיל את מנוע המערכת ויחשוף את היכולות בדמות endpoints שונים.  
   אף לא אחד מהקליינטים השונים יכול לדבר ישירות אחד עם השני.   
   כל "השיחה" והעברת המידעים בינהם מתבצעת אך ורק דרך/באמצעות השרת.

**Shticell  
Server  
(Tomcat)**

**HTTP**

Client

Client

Client

**…**

1. לכל משתמש במע' יש שם ייחודי.  
   ההתחברות למע' מתחילה בעמוד הלוגין בו יידרש המשתמש להכניס את שמו.  
   באם שם המשתמש כבר קיים אזי יש להעביר הודעת שגיאה מתאימה למשתמש ולאפשר לו להתחבר מחדש.  
   (**אין** "להגדיל ראש" ולעבוד עם סיסמאות או כל תהליך של sign up, וזאת כדי להקל על הבודק)  
     
   לאחר התחברות מוצלחת יועבר המשתמש למסך ניהול הגליונות
2. העלאת קבצים:  
   כל משתמש רשאי להעלות קבצי גליונות למע' (קובץ ה XML בגרסה של תרגיל 2).  
   הפעם, בניגוד לתרגיל 2 ו 3, המע' צוברת את הקבצים המועלים אליה (ולא מחליפה אותם אחד בשני.)  
     
   הקובץ נבחר ממחשב המשתמש ומועלה אל השרת.   
   העלאת הקובץ יכולה וצריכה להתבצע בהתאם לדוגמא שנלמדה בכיתה.   
   **אין שום צורך** להשתמש בספריות צד שלישי (apache commons וכיוצב') בשביל להעלות את הקובץ

**אין שום צורך** לשמור את תוכן הקובץ בשרת, ויש להימנע מכך בכל מחיר !   
לשרת (של הבודק) אין הרשאות לכך ואתם תקרסו !

כשתוכן הקובץ מגיע לשרת, יבוצעו בו בדיקות הקלט כפי שבוצעו במהלך תרגיל 1 ו 2.   
נוסף לבדיקות הנ"ל, יש להקפיד כי לכל גליון יש שם ייחודי רק לו.   
  
באם הקובץ תקול – הודעה מתאימה הכוללת את סיבת התקלה צריכה להופיע למשתמש והקובץ **אינו** מוכר כתקין בצד השרת.   
אם הקובץ תקין – הודעה מתאימה צריכה להגיע למשתמש  
שימו לב: אין צורך לתמוך או להמשיך להציג את העלאת הקובץ באמצעות task של JFX תוך הצגת progress bar או הטמעת השהיה מלאכותית. גם ככה עכשיו זו תהיה פעולה אסינכרונית כי היא מתרחשת מול השרת ותיקח את הזמן שלה...

1. ניהול גליונות והרשאות:  
   במסך זה ניתן לראות רשימה מסודרת של כלל הגליונות המתנהלים במע'.  
   עבור כל גליון יש להציג את המידעים הבאים:
   1. שם המשתמש שהעלה את הגליון
   2. שם הגליון
   3. גודל הגליון: nXm (כפי שמוגדר בקובץ ה XML)
   4. סוג ההרשאה שיש למשתמש הנוכחי עבור הגליון (הסבר בסעיף הבא)

משתמש יכול לבחור לצפות בגליון שהוא העלה. בחירת גליון תדלג אותו למסך השלישי שבו צופים בגליון בודד.

1. על מנת לקבל גישה לגליון של משתמש אחר, המשתמש חייב לקבל לכך הרשאה מפורשת.  
     
   מנגנון ההרשאות מאפשר מספר סוגי הרשאות:

* OWNER – הרשאה זו קיימת רק לבעל הגליון שהעלה אותו. לא ניתן להעניק אותה למשתמשים אחרים.  
  רק ה Owner יכול לשנות ולעדכן הרשאות של משתמשים במע'.
* READER - הרשאה לצפייה בלבד:  
  הרשאה זו מאפשרת למשתמש רק לראות את הפרטים בגליון אולם הוא לא יכול לערוך אותו, לשנות את התאים שלו, את העיצוב וכו'.  
  משתמש עם הרשאה זו כן רשאי לבצע כל פעולה שלא משנה את הגרסה של הגליון, כולל מיון, סינון, כמו גם לראות את הגליון באחת מגרסאותיו הישנות.
* WRITER - הרשאה לעדכון:   
  הרשאה זו מאפשרת למשתמש לבצע את כל העדכונים האפשריים בגליון, בדיוק כשם שהיה במסגרת תרגיל 2.
* NONE – אין כרגע שום הרשאה למשתמש. זוהי ברירת המחדל עבור כל גליון שהועלה ע"י משתמש שהוא לא אני

על מנת לקבל הרשאה, על המשתמש לבחור את הגליון ולהגיש בקשה לקבלת הרשאה.   
במסגרת הבקשה יש לציין את הגליון הנדרש כמו גם את סוג ההרשאה המבוקשת.  
  
עבור הגליון הנבחר, ניתן לראות מידע על כלל המשתמשים וההרשאות שהתקבלו.  
יש להציג טבלה המציגה את כלל המשתמשים שיש להם הרשאה לגליון זה.  
יש להציג את המידעים הבאים בטבלה:

* שם המשתמש
* סוג ההרשאה
* האם אושרה ע"י ה Owner

כאשר משתמש אחר מגיש בקשה לקבלת הרשאה – הפרטים של הבקשה יופיעו גם כן בטבלה והיא תצוין ככזו שלא אושרה ע"י ה owner.  
ברצותו, יוכל המשתמש לאשר את הבקשה ואז המשתמש אשר ביקש את ההרשאה ייודע על כך, ויוכל לבחור לתצוגה את הגליון המדובר.  
שימו לב: המשתמש שהוא ה Owner של הגליון תמיד יופיע בטבלת המשתמשים עם סוג ההרשאה של owner.  
  
כך או אחרת, בבחירת גליון שיש הרשאה לראותו, המשתמש מועבר למסך השלישי שבו אפשר לתפעל גליון.

1. מסך תפעול גליון:  
   זהו למעשה מסך הזהה במתכונתו למסך המרכזי של המע' מתרגיל 2, להוציא את החלק העליון של העלאת הקבצים.  
   במסך זה המשתמש רואה את הגליון ובהתאם להרשאה שלו יכול להתעסק איתו כשם שהיה בתרגיל 2.
2. עדכונים סמולטנים:  
   מאחר ועכשיו כמה משתמשים יכולים לצפות ו/או לערוך גליון נתון, הרי שהמע' תתמוך בהצגה ועדכונים סמולטנים של הגליון.  
     
   במידה ולמשתמש יש רק הרשאות צפייה, ובמידה וקיימת גרסה עדכנית יותר של הגליון בבטן השרת (בעקבות עדכונים של משתמשים אחרים, ואחרי מח"מ מוצלח) – המשתמש יקבל חיווי כי קיימת גרסה עדכנית יותר של הקובץ מאשר זו שהוא רואה.  
   החיווי יופיע בתיבת הבחירה של הגרסה הנצפית.   
   לבחירתכם כיצד לממש זאת: אולי שינוי צבע הכפתור ? הבלטת המסגרת ?   
   (בכל מקרה אין להקפיץ את החיווי באמצעות Popup טורדני שיפריע למשתמש במסגרת צפייתו בגליון.)  
   ברצות המשתמש (ורק ברצותו, אין לאלץ אותו לכך) – יבחר להתקדם לגרסה העדכנית יותר של הגליון.  
     
   במידה ולמשתמש יש הרשאות עריכה, הרי שהוא יכול לערוך בצורה חופשית את כלל התאים במע'.   
   במקרה זה פעולת העריכה תצלח אך ורק אם המשתמש צופה ומעדכן על בסיס הגרסה האחרונה והעדכנית ביותר של הגליון.  
   במידה והוא אכן במצב זה – הרי שהשרת יבצע את העדכון שלו, ויקדם את הגרסה (והגרסה העדכנית תופץ לכלל המשתמשים שצופים/עורכים גליון זה).  
   במידה והמשתמש ניסה לבצע עדכון על גרסה לא עדכנית של השרת – עליו לקבל הודעת מפורטת המסבירה זאת ולמעשה הוא יכול לבחור לחזור לגרסה העדכנית של ביותר של הגליון ולנסות לבצע את העדכון שלו בשנית.  
     
   במסגרת כל עדכון שמתבצע בהצלחה במע' יש להוסיף למידע של כל תא את שם המשתמש שביצע את העדכון (גם במקרים בהם תא מסויים התעדכן בעקיפין בעקבות שינוי של משתמש זה או אחר).  
   המידע על זהות המשתמש ששינה את התא מתווסף לשורת הפעולה במסך השלישי כאשר בוחרים תא מסויים
3. ניתוח דינמי:  
   יכולת זו תאפשר למשתמש לבצע ניתוח דינמי של תוצאות הגליון במידה וערך מספרי מסוים משתנה (הנה [דוגמא](https://grid.is/?utm_firstpage=%2F&utm_seq=5)).  
   יכולת זו מותאמת ומוגבלת לעבודה על/עם ערכים מספריים בלבד.  
   במסגרתה, המשתמש יוכל לבחור את אחד התאים השונים (אשר אינם מכילים קריאה לפונקציה), להגדיר טווח מספרי (מינימום ומקסימום) ואת גודל הצעד, וכך לשנות את ערכו של התא בין שני הטווחים הנ"ל, כגודל הצעד.   
   יש להציג את היכולת הזו למשתמש באמצעות slider מסודר.  
     
   על כל שינוי שיבוצע בערכו של התא יבוצע המח"מ ברקע ויוצג הגליון עם העדכון הנדרש.  
   תהליך זה יאפשר ביתר קלות ניתוח דינמי של הגליון בהתאם להשתנות ערך מסוים.  
   המידע שיוצג יהיה מידע זמני ובפועל לא ישנה את ערכו של התא בגליון ולא יוסיף גרסה חדשה לגליון
4. חלק מפרטי המידע במע' צריכים להתעדכן בצורה אוטומטית עבור כל משתמש בצורה של Pull כפי שלמדנו בשיעור והודגם בדוגמא המסכמת (אם בשיטה של all או של delta fetching או כל דבר באמצע – כל מקרה לגופו).   
   אפשר לבצע את ה Pull בטווח זמן של עד 2 שניות, אולם זמן סביר ל pull הוא סדר גודל של חצי/שנייה.
5. אין צורך לבצע שמירה של נתונים מעבר למופע הנוכחי של השרת. במילים אחרות כשהשרת יורד - כל הנתונים נעלמים.

לא שומרים את המשתמשים שנרשמו, הגליונות וההרשאות שניתנו וכו'.

1. יש לדאוג ולוודא כי מסכי המע' לסוגם מתנהגים יפה בresize ולא מחסירים שום פרט
2. חלק ניכר מהרכיבים כבר קיימים לכם במסגרת תרגיל 2. יש לשאוף למחזר ולהשתמש בהם כמה שרק אפשר.

## חלוקה למודולים

יש לייצר מודול נוסף שממנו ייבנה war המכיל את כל ה jar'ים הנדרשים (Jar של מנוע המע' ; gson.jar ; אולי אחרים ?).

בנוסף, יש לייצר מודול חדש עבור האפליקציה של תרגיל 3, אשר יכיל את כלל המסכים והרכיבים הנדרשים לה.  
מודול זה יכול וצריך להתבסס על הרכיבים הקיימים לכם כבר במסגרת תרגיל 2.

## איך מתחילים ?

אני ממליץ להתחיל במעבר יסודי על הדוגמא המסכמת של הקורס.   
היא תקנה לכם שליטה בסיסית בדרך העבודה עם http client ובעבודה מול השרת ומהלך החיים והתפקידים השונים של כל רכיב ושחקן בסיפור.

התחילו בפיתוח אפליקציית המשתמש ללא מודל ההרשאות.   
המשתמש יוכל להעלות קבצים ולראות את כלל הגליונות הנצברים במע'. כל משתמש שמתחבר למע' יוכל לראות רשימה זו גם כן ולקבל עליה עדכונים שוטפים וכו'.

המשיכו בפיתוח מודל ההרשאות בצד המנוע שבשרת. לאחר מכן פתחו את הרכיב הנדרש במסך המשתמש כדי לתאר את כלל הפעולות והניהול של ההרשאות.  
עכשיו תוכלו לפתח את המסך השלישי (הזהה בעיקרו לתרגיל 2), רק שהפעם הוסיפו עליו את כל היכולות (או המגבלות) הנגזרות מהרשאת המשתמש (מינעו ממנו לבצע עדכונים וכו' במידה ויש לו הרשאה לקריאה בלבד).

עכשיו המשיכו לממש את היכולת לבצע עריכה סמולטנית על כלל מרכיבי העדכון שלה עבור המשתמשים השונים.  
לבסוף התעסקו עם הפיצ'ר של הניתוח הדינמי.

לאורך פיתוח האפליקציה יש לשאוף להשתמש כמה שאפשר ברכיבים הקיימים בתרגיל 2, גם כדי להקל משמעותית על עומס הפיתוח ובעיקר כדי להתנסות (ולהוכיח לעצמכם !) איך אפשר לפתח רכיב פעם אחת ולהשתמש בו בהקשרים נרחבים אחרים.

ממליץ מאוד (!), לאורך כל העבודה, להקפיד לעבוד ולבחון את עצמכם דרך postman כאמצעי וידוא כי הסרבלטים עובדים כראוי ללא תלות בקליינט כזה או אחר. כך תוכלו לוודא כי צד השרת עובד טוב (up to a degree) טרם כתיבת החלק הרלבנטי בקליינט.

## בונוסים

|  |  |  |  |
| --- | --- | --- | --- |
| # | מהות | למה שווה לי ? | כמה שווה לי ? |
| 1 | Chat:  אפשרו למשתמשים לקיים chat בינהם תו"כ התחברותם למע'.  ה chat צריך להתנהל כולו אל מול השרת בצורה שבה כולם רואים את מה שכולם כותבים. (מאוווווווווווד דומה לדוגמא הסופית)  הצ'ט מתבצע בין המשתמשים הרשומים והמחוברים למערכת בלבד. | היש מדהים מזה ??? | 5 נקודות  (ועד ל 100) |
| 2 | ניתוח דינמי מרובה משתנים אפשרו למשתמש לבצע את הניתוח הדינמי באמצעות כמה משתנים ולא רק באמצעות אחד | במימוש נכון – אמור להיות פיסת עוגה ! | 7 נקודות **(מעל ל 100)** |

## סבבה, סיימתי. מה ואיך להגיש ?

יש להגיש קובץ zip המכיל:

1. WAR אחד בדיוק אשר יושם בספריית tomcat\webapp ויעבור deployment אוטומטי.   
   על WAR זה לכלול את כל התלויות שלכם. אין להניח כי יסופקו לכם תלויות מבחוץ (למשל Gson וכו').
2. ספרייה עבור הקליינט החדש, בה יישבו כל ה jar'ים הנדרשים כדי לתפעל את האפליקציה הרלבנטית.   
   יש לספק לאפליקציה קובץ batch שמפעיל אותה (כמו שהיה בתרגיל 1 ו 2). קובץ זה ישב גם כן בספריה הרלוונטית.  
   שימו לב כי האפליקציה שלכם צריכה להכיר אוטומטית את השרת ואת ה context path הרלוונטי לה   
   אפשר כמובן להניח את localhost:8080 כדומיין של השרת; אפשר להניח כי לא נשנה את שם קובץ ה WAR שהגשתם.
3. קבצי קוד המקור של הפרויקט שלכם (גם צד השרת וגם אפליקציות java fx של ה client).
4. 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.
5. יש לכלול בקובץ ה **readme** גם תיעוד והסבר כללי (וממצה) של המחלקות העיקריות החדשות ותפקידם. (חישבו מה יסייע לבודק להיכנס ביתר קלות לקוד שלכם ולהבין מי נגד מי...)
6. במידה ומימשתם את מי מהבונוסים לעיל, ציינו את שם הבונוס **בראשית** קובץ ה readme כדי שהבודק ידע לבדוק בהתאם.

בונוס שימומש אבל לא יתועד – לא ייבדק !

הגשה באיחור, שאינה באישור, תבטל כל מימוש בונוס. אין להגיש באיחור בשביל להספיק לעשות בונוסים.   
תכננו את הזמן בהתאם.

נספח א' –פונצקיות עזר נתמכות

שטיסל תתמוך בפונקציות עזר שניתן יהיה לשלבן במסגרת התאים השונים.  
לכל פונקציה יש שם ייחודי, כמו גם הגדרה של אילו ארגומנטים היא מקבלת (שם, סוג, סדר) ואיזה ערך היא מחזירה.  
  
המערכת תאפשר להגדיר ביטוי (expression) שיאפשר הרכבה של פונקציה המקבלת כארגומנט פונקציה אחרת וכו'.  
  
הפעלה של פונקציה תוגדר באמצעות המבנה הבא:

**{**<function name>,<arg 1>,<arg 2>,…,<arg n>**}**

כאשר:

1. כל הגדרת פונקציה תחומה בתוך מסולסליים **{** … **}**
2. function name: הוא שמה הייחודי של הפונקציה. נתון באותיות גדולות בלבד
3. כל ארגומנט מופרד בתו פסיק (,) מהארגומנט הבא בתור. אפשר להניח שהארגומנטים צמודים אחד לשני ואין רווחים בשביל הנראות בין הפסיקים (כלומר אם יש רווח – הוא חלק מהארגומנט, ויש להתייחס אליו ככזה)
4. arg i הוא הארגומנט ה i שהפונקציה מקבלת.  
   הוא יכול להיות מספר/מחרוזת/ערך בוליאני מפורש, או הפניה לפונקציה אחרת (שגם היא תחומה ב **{ }** וכו')
5. במידה ופונקציה לא יכולה להתחשב, יוקצו ערכים ייעודיים לסמן תוצאה פסולה:
   1. NaN – עבור ערך מספרי. Not A Number
   2. !UNDEFINED! – עבור ערך מחרוזת
   3. UNKNOWN – עבור ערך בוליאני
6. במידה ופונקציה אינה תקינה מבחינת המבנה שלה (שם לא מוכר; כמות ארגומנטים לא נכונה) – אין לקבל את הערך שלה והיא עוצרת את המח"מ.
7. כל מצב שבו הארגומנטים אינם על פי הגדרת הפונקציה ייחשב קלט תקין, אך כזה שמוביל לקבלת ערכים פסולים
8. כל פעולה שמתבצעת על/באמצעות תא עם ערך פסול – דינה להיות פסולה גם כן (הפוסל במומו פוסל..)

דוגמאות:

* {PLUS, 4, 5} – חיבור של 4 עם 5
* {MINUS, {PLUS,4,5},{POW,2,3}} – חיבור של 4 עם 5 ולחסר ממנו את תוצאת החזקה של 2 ב 3

הנה רשימת פונקציות העזר שהמערכת תומכת בהן, בחלוקה לפי תחומים ועל פי דרישות התרגיל:

|  |  |  |
| --- | --- | --- |
| פונקציות לוגיות | | |
| # | שם הפונקציה | נדרש החל מתרגיל |
| 1 | EQUAL | 2 |
| 2 | NOT | 2 |
| 3 | OR | 2 |
| 4 | AND | 2 |
| 5 | BIGGER | 2 |
| 6 | LESS | 2 |
| 7 | IF | 2 |

|  |  |  |
| --- | --- | --- |
| פונקציות מתמטיות | | |
| # | שם הפונקציה | נדרש החל מתרגיל |
| 1 | PLUS | 1 |
| 2 | MINUS | 1 |
| 3 | TIMES | 1 |
| 4 | DIVIDE | 1 |
| 5 | MOD | 1 |
| 6 | POW | 1 |
| 7 | ABS | 1 |
| 8 | SUM | 2 |
| 9 | AVERAGE | 2 |
| 10 | PERCENT | 2 |

|  |  |  |
| --- | --- | --- |
| פונקציות מחרוזת | | |
| # | שם הפונקציה | נדרש החל מתרגיל |
| 1 | CONCAT | 1 |
| 2 | SUB | 1 |

|  |  |  |
| --- | --- | --- |
| פונקציות מערכתיות | | |
| # | שם הפונקציה | נדרש החל מתרגיל |
| 1 | REF | 1 |

1. PLUS(arg1, arg2)

**תיאור:**

הפונקציה מבצעת חיבור בין שני ארגומנטים

**ארגומנטים:**

arg1 – מספר / פונקציה המחזירה מספר

Arg2 – מספר / פונקציה המחזירה מספר

**ערך מוחזר:**

תוצאת החיבור.

1. MINUS(arg1, arg2)

**תיאור:**

הפונקציה מבצעת חיסור בין שני ארגומנטים. arg1 – arg2  
  
**ארגומנטים:**

arg1 – מספר / פונקציה המחזירה מספר

Arg2 – מספר / פונקציה המחזירה מספר

**ערך מוחזר:**

תוצאת החיסור.

1. TIMES(arg1, arg2)

**תיאור:**

הפונקציה מבצעת כפל בין שני ארגומנטים.  
 **ארגומנטים:**  
arg1 – מספר / פונקציה המחזירה מספר

Arg2 – מספר / פונקציה המחזירה מספר  
  
**ערך מוחזר:**

תוצאת המכפלה

1. DIVIDE(arg1, arg2)

**תיאור:**  
הפונקציה מבצעת חילוק בין שני ארגומנטים. arg1 / arg2   
במידה ו arg2 = 0 הפונקציה לא יכולה להיות מחושבת וערכה הוא NaN.  
 **ארגומנטים:**  
arg1 – מספר / פונקציה המחזירה מספר. זהו המונה

Arg2 – מספר / פונקציה המחזירה מספר. זהו המכנה  
  
**ערך מוחזר:**

תוצאת החלוקה

1. MOD(arg1, arg2)

**תיאור:**  
הפונקציה מבצעת פעולת מודולו בין שני ארגומנטים: arg1 % arg2  
  
**ארגומנטים:**  
arg1 – מספר / פונקציה המחזירה מספר

Arg2 – מספר / פונקציה המחזירה מספר

**ערך מוחזר:**  
תוצאת הפעולה – מספר

1. POW(arg1, arg2)

**תיאור:**  
הפונקציה מבצעת העלאה בחזקה בין שני ארגומנטים: arg1arg2  
**ארגומנטים:**  
arg1 – מספר / פונקציה המחזירה מספר

Arg2 – מספר / פונקציה המחזירה מספר  
  
**ערך מוחזר:**

תוצאת החזרה – מספר

1. ABS(arg1)

**תיאור:**  
הפונקציה מבצעת ערך מוחלט של ארגומנט: |arg1|  
  
**ארגומנטים:**  
arg1 – מספר / פונקציה המחזירה מספר  
  
**ערך מוחזר:**

הערך המוחלט של הארגומנט – מספר

1. CONCAT(str1, str2)

**תיאור:**  
הפונקציה מבצעת שרשור של שני ארגומנטים.  
  
**ארגומנטים:**  
str1 – מחרוזת / פונקציה המחזירה מחרוזת

str2 – מחרוזת / פונקציה המחזירה מחרוזת  
  
**ערך מוחזר:**

המחרוזת המשורשרת

1. SUB(source, start-index, end-index)

**תיאור:**הפונקציה מבצעת חיתוך של תת מחרוזת מתוך מחרוזת מקור.  
במידה ואחד מהאינדקסים חורג מגבולות המחרוזת (קטן מ 0; גדול מאורכה וכו') – יש להחזיר !UNDEFINED!

**ארגומנטים:**  
source – מחרוזת / פונקציה המחזירה מחרוזת: זוהי מחרוזת המקור שעליה צריך לעבוד

start-index – מספר / פונקציה המחזירה מספר. החל מאינדקס זה (מבוסס 0) יש להתחיל לחתוך

end-index – מספר / פונקציה המחזירה מספר. עד לאינדקס זה (כולל) יש לסיים לחתוך

**ערך מוחזר:**תת המחרוזת

1. EQUAL(arg1, arg2)

**תיאור:**  
הפונקציה מבצעת השוואה בין שני ארגומנטים.  
אם הארגומנטים זהים בסוגם (שניהם מספרים/מחרוזות/בוליאנים) – ההשוואה תתבצע בצורה הטבעית  
אם הארגומנטים שונים בסוגם – יש להחזיר FALSE  
  
**ארגומנטים:**  
arg1 – כל סוג של מידע

arg2 – כל סוג של מידע  
  
**ערך מוחזר:**

ערך בוליאני: TRUE FALSE |

1. NOT(exp1)  
     
   **תיאור:**פונקציה זה תבצע הופכי בוליאני של הביטוי שהיא מקבלת  
     
   ארגומנטים:  
   exp1 – ערך בוליאני / פונקציה המחזירה ערך בוליאני  
     
   **ערך מוחזר:**  
   ערך בוליאני: TRUE FALSE |
2. BIGGER(arg1, arg2)

**תיאור:**  
הפונקציה תבדוק אם arg1 גדול או שווה ל arg2: arg1 >= arg2  
  
**ארגומנטים:**  
arg1 – מספר / פונקציה המחזירה מספר

arg2 – מספר / פונקציה המחזירה מספר  
  
**ערך מוחזר:**

ערך בוליאני: TRUE FALSE |

1. LESS(arg1, arg2)

**תיאור:**  
הפונקציה תבדוק אם arg1 קטן או שווה ל arg2: arg1 <= arg2

**ארגומנטים:**  
arg1 – מספר / פונקציה המחזירה מספר

arg2 – מספר / פונקציה המחזירה מספר

**ערך מוחזר:**  
ערך בוליאני: TRUE FALSE |

1. OR(exp1, exp2)

**תיאור:**  
הפונקציה מבצעת פעולת OR בין שני ארגומנטים בוליאנים: arg1 || arg2  
  
**ארגומנטים:**  
exp1 – ערך בוליאני

exp2 – ערך בוליאני  
  
**ערך מוחזר:**

ערך בוליאני: TRUE FALSE |

1. AND(exp1, exp2)

**תיאור:**  
הפונקציה מבצעת פעולת AND בין שני ארגומנטים בוליאנים: arg1 && arg2  
  
**ארגומנטים:**  
arg1 – ערך בוליאני

arg2 – ערך בוליאני

**ערך מוחזר:**  
ערך בוליאני: TRUE FALSE |

1. REF(cell-id)

**תיאור:**הפונקציה מאפשרת לקבל את ערכו של תא אחר  
  
**ארגומנטים:**  
cell-id – מזהה תא במע'. דוגמא: 5A. יוצג באות גדולה ומספר שלם צמוד אחריה; יכול להינתן גם באות קטנה.

**ערך מוחזר:**  
ערכו האפקטיבי של התא הנוכחי.

1. IF(condition, then, else)

**תיאור:**  
הפונקציה מבצעת הערכה של התנאי. אם הוא חיובי – אזי היא תבצע הערכה של ה then ותחזיר אותו.  
אם הוא שלילי היא תבצע הערכה של ה else ותחזיר אותו.  
במידה וה then או ה else לא מפיקים את אותו סוג התוצר – הגדרת הפונקציה אינה חוקית.  
  
**ארגומנטים:**condition – ערך בוליאני / פונקציה שמחזירה ערך בוליאני

Then – כל מידע במע'  
else – כל מידע במע'

**ערך מוחזר:**תוצאת הערכת ביטוי ה then או else

1. SUM(range)

**תיאור:**

הפונקציה תבצע סכימה של כלל התאים המספריים הנמצאים ב range המדובר.  
אם ה range לא קיים – הגדרת הפונקציה אינה חוקית  
אם ה range לא כולל אף תא מספרי – הסכום הוא 0 (זו אינה נחשבת תקלה !)

**ארגומנטים:**range – שם של range שהוגדר מבעוד מועד

**ערך מוחזר:**ערך מספרי. תוצאת הסכימה של כלל הערכים המספריים בטווח

1. AVERAGE(range)

**תיאור:**

הפונקציה תחשב ממוצע של כלל התאים המספריים הנמצאים ב range המדובר.  
המנה של הממוצע תהיה ככמות התאים המספריים הנמצאים ב range המדובר  
אם ה range לא קיים – הגדרת הפונקציה אינה חוקית  
אם ה range לא כולל אף תא מספרי – הגדרת הפונקציה אינה חוקית

**ארגומנטים:**range – שם של range שהוגדר מבעוד מועד

**ערך מוחזר:**ערך מספרי. תוצאת הממוצע של כלל הערכים המספריים בטווח

1. PERCENT(part, whole)  
     
   **תיאור:**  
   הפונקציה תחשב את חלק האחוז מתוך השלם: part\*whole/100  
   (דוגמא: percent(10, 50) = 5)  
     
   **ארגומנטים:**  
   part – ערך מספרי / פונקציה המחזירה ערך מספרי  
   whole – ערך מספרי / פונקציה המחזירה ערך מספרי  
     
   **ערך מוחזר:**  
   ערך מספרי. ערכו של האחוז מתוך השלם.

נספח ב' – תיאור מבנה המערכת באמצעות קובץ XML

תיאור המע' ומרכיביה בתרגילים השונים נתון באמצעות קובץ XML.

במהלך הבדיקה (של שלל התרגילים), תיבדק המערכת באמצעות מספר קבצים שונים, חלקם חוקיים וחלקם תקולים, 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 במפורש case **in**sensitive הכוונה היא שאין חשיבות ל case של האותיות באנגלית.   
במקרה זה הערך milk זהה לערך MiLk  
בכל מקרה אחר הנחת היסוד היא כי יש חשיבות ל case של האותיות (כלומר case sensitive). במקרים אלה הערך milk שונה מהערך MiLk.

היכן שמצויין שהמחרוזת יכולה להכיל רווחים – המדובר הוא רק על רווחים בתוך המחרוזת.   
אם מופיעים רווחים בתחילתה/סופה יש להתעלם מהם (רמז: המטודה trim() על המחלקה String)

כברירת מחדל כל ה element'ים הם mandatory, אלא אם מצויין במפורש כי הם אופציונלים.

A white background with black and red text

Description automatically generatedמבנה המע' מאוגד תחת האלמנט **STL-Sheet**:

|  |  |  |  |
| --- | --- | --- | --- |
| # | סוג | שם | מהות |
| 1 | Element | STL-Sheet | אלמנט זה מכיל את כלל פרטי הגליון ומתאר את מאפייניו השונים |
| 2 | Attribute | name | מתאר את שם הגליון. מחרוזת היכולה להכיל רווחים |
| 3 | Element | STL-Layout | מתאר מידע על מבנה הגליון |
| 4 | Element | STL-Cells | מתאר מידע על תוכן התאים שבגליון |

A math equations on a white background

Description automatically generated האלמנט **STL-Layout** מתאר את סך משתני הסביבה של הסימולציה

|  |  |  |  |
| --- | --- | --- | --- |
| # | סוג | שם | מהות |
| 1 | Attribute | rows | מגדיר את כמות השורות בגליון. מספר שלם חיובי, גדול מ 0 |
| 2 | Attribute | columns | מגדיר את כמות העמודות בגליון. מספר שלם חיובי, גדול מ 0 |
| 3 | Element | STL-Size | מגדיר את רוחב/גובה העמודות/שורות |
| 4 | Attribute | column-width-units | מגדיר את רוחב כל עמודה ביחידות. ב console יחידה היא תו בווד (תו הרווח) ובממשק גרפי היחידה היא pixel. מספר שלם חיובי, גדול מ 0 |
| 5 | Attribute | rows-height-units | מגדיר את גובה כל שורה ביחידות. ב console יחידה היא תו בווד (תו הרווח) ובממשק גרפי היחידה היא pixel. מספר שלם חיובי, גדול מ 0 |

האלמנט **STL-Cells** כולל תיאור של כלל התאים בגליון.   
הוא כולל 0 או יותר אלמנטים מסוג של **STL-Cell**:

A black and red text

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| # | סוג | שם | מהות |
| 1 | Attribute | row | מתאר את מספר השורה של התא המדובר. מספר שלם גדול מ 0. |
| 2 | Attribute | column | מתאר את מספר העמודה של התא המדובר. מספר שלם גדול מ 0. |
| 3 | Element | STL-Original-Value | מתאר את הערך המקורי של התא המדובר. מחרוזת תווים היכולה גם להכיל רווחים |

## סכמה תרגיל 2

סכמה זו מוסיפה A screenshot of a computer code

Description automatically generated**תמיכה בטווחים:**האלמנט STL-Ranges מגדיר את כלל הטווחים המוגדרים בגליון.  
הוא מכיל 0 או יותר אלמנטים מסוג של STL-Range:

|  |  |  |  |
| --- | --- | --- | --- |
| # | סוג | שם | מהות |
| 1 | Attribute | name | זהו השם של הטווח. מחרוזת תווים ללא רווחים. |
| 2 | Element | STL-Boundaries | מייצג את גבולות הטווח |
| 3 | Attribute | from | מייצג את זהות התא בגבול השמאלי עליון שממנו מתחיל הטווח. זוהי מחרוזת זהות תא: אות גבוהה ומספר |
| 4 | Attribute | to | מייצג את זהות התא בגבול הימני תחתון שבו מסתיים הטווח. זוהי מחרוזת זהות תא: אות גבוהה ומספר |

נספח ג' – תיאור גרפי של הסכמה

סכמה 1V:

A screenshot of a computer

Description automatically generated

סכמה 2V:

A screenshot of a computer

Description automatically generated