

## תרגיל בית 3

### הנחיות לפתרון תרגילי הבית

- על הקוד המוגש להיות מתועד היטב ועליו לכלול:
  - מפרט, כפי שהודגם בתרגול.
  - תיעוד של כל מחלקה ומתודה ושל קטעי קוד רלוונטיים.
  - במידת הצורך, יש להוסיף תיעוד חיצוני במסמך "ההגשה היבשה"
  - במהלך תרגיל בית זה אתם נדרשים לשרטט תרשימי UML, אפשר ואפילו רצוי להיעזר בתוכנות לשרטוט תרשימים כמו Visio, LucidChart, Dia, draw.io.

### הנחיות להגשת תרגילי בית

- תרגילי הבית הם חובה.
- ההגשה בזוגות בלבד.
- עם סיום פתירת התרגיל, יש ליצור קובץ zip להגשה המכיל את:
  - כל קבצי הקוד והתיעוד לפי מסמך ההוראות להגשת תרגילי הבית במודל.
    - השורה הראשונה בתוך כל קובץ קוד java צריכה להיות  
package homework3;
  - פתרון לשאלות ה"יבשות" בקובץ PDF. על הקובץ להכיל את מספרי תעודות הזהות של שני הסטודנטים המגישים בלבד, ללא שמות.
- הגשת התרגיל היא אלקטרונית בלבד, דרך אתר הקורס ע"י אחד מבני הזוג בלבד. הקובץ המוגש יקרא **hw3\_<id1>\_<id2>.zip** כאשר <id1> ו-<id2> הם מספרי הזהות של הסטודנטים המגישים. לדוגמא hw3\_12345678\_9876541.zip (כמובן יש להשתמש במספרי הזהות שלכם).

יש להגיש את תרגיל בית 3 לפי ההנחיות הבאות בסדר הבא:

1. שני בני הזוג צריכים להצטרף לאותה קבוצה ב"בחירת קבוצות לתרגיל בית 3" במודל.
2. אחד מבני הזוג צריך להגיש את קובץ ההגשה ב"מטלת תרגיל בית 3".

לא יהיו הארכות למועד הגשת התרגיל למעט בקשות מוצדקות כמו מילואים, אשפוז, וכו'.  
אנא תכננו את הזמן שלכם בהתאם.

- על התוכנית לעבור הידור (קומפילציה). על תכנית שלא עוברת הידור יורדו 30 נקודות.

מועד ההגשה:  
יורד 17.6.2025

התרגיל בנוי משני חלקים:  
בחלק הראשון תבצעו modeling ותכן למערכת תכנה תוך שימוש בתרשימי UML.  
בחלק השני תבצעו refactoring לקוד קיים.

### חלק ראשון (50 נקודות):

- המטרות של תרגיל בית זה הן להתנסות בתחומים הבאים:
- ניתוח דרישות וניתוח use cases על מנת ליצור מסמך דרישות.
  - תכן תוכנה.
  - שימוש בתרשימים בשפת UML.

### הצגת הבעיה

יוסי סלט הוא מנהל מסעדה המבקש שתיצרו עבורו תוכנה שתעזור לו בניהול המסעדה. את דרישותיו מהתוכנה סיכם יוסי סלט במסמך הבא:

אני מעוניין במערכת לניהול מסעדה. המסעדה שלי, "הצלע השנייה", היא מסעדה מצליחה מאוד, המשרתת מאות לקוחות בכל יום. יש במסעדה צוות מלצרים, מטבח גדול, דוכן עוגות ובר. בשעות השיא, העומס גדול מאוד והבלגן חוגג, ולכן אני צריך מערכת שתעזור לי להשאיר את העניינים בשליטה.

מה שאני רוצה שיקרה בעת השימוש במערכת הוא. כאשר מגיעים לקוחות, אחד המלצרים יושיב אותם ליד שולחן. לאחר מכן, המלצר ייקח את הזמנתם. המנות שהוזמנו יבחרו ע"י המלצר מתוך תפריט ממוחשב. תפריט זה צריך להיות קל לשימוש משום שהמלצר יהיה תחת לחץ של זמן. בסיום בחירת המנות, העמדה המתאימה במסעדה תקבל הודעה על המנות להכנה – המטבח אחראי על האוכל, הבר אחראי על המשקאות ודוכן העוגות אחראי על הקינוחים. לאחר שהעובד בעמדה סיים להכין את המנה, הוא יסמן אותה כמוכנה. כאשר כל המנות המרכיבות את ההזמנה מוכנות, המלצר יקבל על כך הודעה ויגיש את המנות לשולחן. בכל זמן לפני שהוגשה ההזמנה לשולחן, הלקוח יכול לבטל או לשנות את הזמנתו. בסיום הארוחה, המערכת תדפיס את החשבון. הלקוח יכול לשלם במזומן או בכרטיס אשראי. תשלום בכרטיס אשראי דורש אימות מול חברת כרטיסי האשראי. התשלום עשוי להיות עבור השולחן כלו או עבור כל לקוח בנפרד. לסיום, תודפס קבלה.

המסעדה מגישה תפריט מגוון ואני מעוניין שהלקוח יקבל בדיוק מה שהזמין. כלומר, אם למנה יש צורות שונות (למשל, סלט עם רטבים שונים) או אם למנה יש תוספות שונות (למשל, תוספות לפיצה), צריכה להיות אפשרות לבחור זאת במפורש. יש לשים לב שלגרסאות שונות של מנה עשויים להיות מחירים שונים.

אני מעוניין שהמערכת תאפשר להציג סטטיסטיקות של ההזמנות. אני במיוחד רוצה לדעת אילו מנות הן הפופולאריות ביותר, כדי שניתן יהיה לקנות חומרי גלם בהתאם, ומהן השעות העמוסות ביותר, כדי שאוכל להעסיק יותר מלצרים בשעות אלה.

בברכה

יוסי סלט,

מנהל מסעדת "הצלע השנייה"

## שאלה 1 (15 נקודות)

עליכם לבצע ניתוח דרישות של המסמך הנ"ל ולהפיק ממנו מסמך דרישות כתוב באנגלית העומד בקריטריונים למסמך דרישות טוב שנלמדו בכיתה.

במקרה של אי בהירות או סתירות בדרישותיו של יוסי סלט, יש להניח כי שוחחתם אתו והבהרתם את הדברים. הניחו הנחות הגיוניות במקרים אלה והשתמשו בהן במסמך הדרישות.

עליכם לרשום באופן מסודר במסמך נפרד את ההנחות שהנחתם כתוצאה מהשיחה ששוחחתם כביכול עם יוסי סלט ואת השפעתן על מסמך הדרישות שיצרתם.

להגשה "יבשה": מסמך הדרישות שיצרתם ובנוסף, מסמך המכיל את ההנחות שהנחתם בעת יצירתם מסמך הדרישות ואת השפעתן על מסמך הדרישות.

## שאלה 2 (35 נקודות)

בשאלה זו תבצעו ניתוח תרחישי שימוש (use cases) חלקי למערכת.

א.

כתבו רשימה של כל השחקנים ושל כל תרחישי השימוש המשתתפים במערכת. עבור כל שחקן ועבור כל תרחיש שימוש רשמו תיאור קצר של לא יותר משתי שורות.

ב.

כתבו תיאור מפורט של שני תרחישי השימוש הבאים: הזמנת ארוחה, הצגת דו"ח של העומס במסעדה בשעות שונות. על התיאור להיות מלא ולהיות כתוב באנגלית במבנה דומה לזה שהוצג בתרגול.

ג.

ציירו תרשים תרחישי שימוש (use case diagram) עבור המערכת.

ד.

ציירו תרשים רצף (sequence diagram) עבור הזמנת ארוחה.

להגשה "יבשה": רשימת השחקנים ותרחישי השימוש עם תיאור קצר, שני תרחישי השימוש הנ"ל, תרשים תרחישי שימוש עבור המערכת.

## חלק שני (50 נקודות):

המטרות של תרגיל בית זה הן להתנסות בתחומים הבאים:

- איתור code smells.
- Code Refactoring.

### הצגת הבעיה

בתרגיל זה תקבלו משחק שנקרא Yatzy וקוד התחלתי שמממש אותו ביחד עם בדיקות של המשחק שכתובות בעזרת ספריית JUnit.

עליכם להבין את המשחק, לעבור על הגרסה הראשונה של הקוד ולזהות code smells. ואז עליכם כתוב גרסה חדשה של הקוד תוך כדי שאתם שומרים על הלוגיקה של המשחק. לצורך כך אתם צריכים לכתוב בדיקות נוספות כדי לוודא שהשינויים שביצעתם לא הכניסו באגים.

הכללים של המשחק נמצאים בקישור הזה:

[https://sammancoaching.org/kata\\_descriptions/yatzy.html](https://sammancoaching.org/kata_descriptions/yatzy.html)

יש לוודא שאתם מבינים את חוקי המשחק טוב לפני שאתם מתחילים את התרגיל.

בכתיבת הקוד, התייחסו לכללים מהסוג הבא (לא בהכרח כולם מופיעים):

- Magic numbers
- Good names
- Indentation
- Comments
- DRY (Don't Repeat Yourself)
- Global data
- Long functions
- Long parameter list
- Static checking
- Fail Fast
- Single responsibility principle

להגשה "רטובה": הקוד שנעשה לו שכתוב + קובץ הבדיקות.

להגשה "יבשה" מסמך שמסביר כל refactor שעשיתם על מה הוא מסתמך.

עבודה נעימה!

