<u>תכנות מונחה עצמים מטלה 0 (חלק שני):</u>
חלק 1: מידול בעיה, תכנון, דיאגרמת מחלקות.
חלק 2: מחלקות, ממשקים אבסטרקציה - ממשים פתרון

מטלה זו מוקדשת לחזרה על נושאים שנלמדו בקורסים קודמים של מבוא לחישוב, ומבנה נתונים.
בניגוד למטלות הבאות מטלה זו היא בעברית, ומוגשת דרך המודל (כל המטלות הבאות תהנה באנגלית והן יוגשו כפרויקט ב Github, אם זאת כל קובצי העזר של המטלה נמצאים באתר הקורס).

במטלה זו נפתח אלגוריתמיקה עבור מערכת מעליות של בניין רב קומות. כללית נחשוב על בניין (Building) כבעל קומת מינימום (יכולה להיות שלילית) וקומת מקסימום, כאשר כל קומות הביניים קיימות. בכל בניין יהיו מספר מעליות (אחת או יותר). לצורך הפשטות נניח שכל מעלית (Elevator) יכולה להגיעה לכל אחת מהקומות. מעבר לכל לכל מעלית יש מאפיינים של זמן עצירה, זמן תחילת תנועה, ומהירות (כמה קומות היא עוברת בשנייה אחת).

כללית נחשוב על המעליות בבניין כ"מעליות חכמות" משמע: המשתמש צריך פשוט להקיש מחוץ למעליות את קומת היעד (destination) ואז המערכת צריכה לשבץ (להקצות) לו מעלית מסויימת אשר מוגדר לה כבר לעצור בקומת היעד.
במטלה זו נתמקד באתגר האופטימיזציה הבא :
בהינתן קריאה למעלית מקומת המקור לקומת היעד - המערכת תרצה לשבץ את המעלית שתצמצם למינימום את זמן ההגעה (זמן ההגעה מוגדר להיות משך הזמן בשניות שבין הקריאה למעלית ובין ההגעה לקומת היעד). באופן יותר כללי נאמר שבהינתן אוסף של קריאות למעליות בזמן נרצה להגדיר אסטרטגיית שיבוץ מעליות לקריאות שתצמצם למינימום את סך משך זמן ההגעה עבור כלל הקריאות.

הנחיות והדרכה:

- התאימו את החלק הראשון של המטלה שלכם - להיות תיעוד של המטלה הנוכחית - שלבו אותו בתיעוד בדגש על המחלקות שממשות את האלגוריתם שלכם לשיבוץ מעליות.
- למען הפשטות נגדיר שבכל הבניינים שלנו, הקומות שערכן קטן או שווה לאפס מהוות קומות כניסה או יציאה מהבניין, וכל הקומות שערכן 1 ומעלה מהוות קומות (מגורים או משרדים) ללא יציאה מהבניין. נניח שבכל בניין קומת המינימום קטנה שווה ל 0 וקומת המקסימום גדולה שווה ל 1.
- באופן טבעי ניתן לחשוב שרוב הזימונים של מעליות הן מאחת מקומות הכניסה לאחת מקומות המגורים (או ההיפך). אבל יתכנו גם זימונים בין קומות כניסה או בין קומות מגורים.
- ניתן לחשוב שכל מעלית יכולה (אם כי לא חייבת) לעצור בכל קומה. ואין הגבלת כמות אנשים לכל מעלית.
- כל מעלית יכולה להיות במצב "עלייה", "ירידה" או "מנוחה". אם לא הוקצתה למעלית שום קריאה היא נשארת במקום במצב "מנוחה". כאשר המעלית מקבלת זימון (משובצת לה "קריאה") היא צריכה להגיעה לקומת הקריאה (המקור) של הזימון ולאחר מכן להגיע לקומת היעד. נגדיר זימון להיות "עולה" אם קומת המקור נמוכה מהיעד (והפוך לגבי זימון "יורד").
- בחלק זה של המטלה עליכם לממש אלג' online, משמע: ברגע שמתקבלת קריאה (זימון) יש לשבץ עבורו באופן מידי מעלית, ולא ניתן לשנות אותו לאחר מכן, ובהתאם לשלוח את המעלית לקומת המקור והיעד. (אין צורך לממש אלג' offline)
- ניתן לעצור את המעלית בקומת ביניים, לדוגמא אם המעלית עולה מקומה 1 לקומה 12 ניתן לעצור אותה בקומה 5 אם היא נמצאת בקומה קטנה או שווה ל 5, אבל לא אחרי זה.

מתחילים לתכנת "תכלס":

שלבי עבודה - חלק ב' מימוש האלגוריתם לכדי פרויקט תוכנה (קוד):

1. כללית רוב הקוד שאתם נדרשים לו ניתן לכם כבר בקישור הבא:
https://github.com/benmoshe/OOP_2021/tree/main/Assignments/Ex0
למעשה אתם נדרשים לממש רק את האלגוריתם online שמימשתם - וכמובן את מחלקות העזר והבדיקות הנדרשות.
2. הורידו את הפרויקט והתקינו אותו כפרויקט java, שימו לב חלק מהקוד ניתן לכם כקובץ jar, קובץ זה מהווה את הסמולטר שיאפשר לכם להריץ אלג' פשוטים שצירפנו לכם וכן לבדוק את האלג' שלכם.
3. הסתכלו על קובץ ה main, הוא כולל דוגמא מפורטת ליצירת קלט, יצירת אלג' (מעלית שבת), והרצה של האלג' על הקלט במשך זמן נתון (סימולציה). כללית סיפקנו לכם 10 תרחישים שונים - [0,9], הריצו אותם כל כל אחד משלושת האלגוריתמים הפשוטים שצירפנו לכם. שימו לב שלאחר כל הרצה - מודפסים נתוני הסימולציה: כולל מספר התרחיש, זמן המתנה מצרפי, ממוצע, וכמות הקריאות שלא הושלמו. שימו לב שגם ת"ז של בעלי הקוד מופיעים בדוח, ובסופו של דבר מופיעה "אסמכתא" של כלל הנתונים. ראו:

Ex0 - OOP Simulator
Time: start: 0.0, end: 1000.0, dt: 1.0
Building: ex0.simulator.Builging_A [-2,10]
0) ex0.simulator.Elevator_A, Elev_0, speed: 1.0, open/close: 2.0, start/stop: 3.0
1) ex0.simulator.Elevator_A, Elev_1, speed: 1.0, open/close: 2.0, start/stop: 3.0
Call log: Elevetor log, size: 100
********** Simulation Ended **********
Code Owners,123456789, Case,2, Total waiting time: 11785.792822120187, average waiting time per call: 117.85792822120187, unCompleted calls,10, certificate, -4231125461

4. לאחר שהצלחתם להריץ את הדוגמא, התחילו במימוש האלגוריתם. הקפידו לתעד היטב ולהוסיף בדיקות הן טכניות של כל אחת מהמחלקות שאתם כותבים, והן בדיקות מערכתיות של הרצה מלאה.
5. חלק מהדרישות של המטלה היא לדווח על ביצועי האלג' שפתחתם ביחס לכל אחד מ 10 התרחישים שהכנו לכם [0,9], לפיכך בכל פעם שאתם מגיעים לתוצאה מעניינת דווחו אליה ב Google Form הבא, הדיווחים שלכם יאפשרו לכם לראות את התוצאות שלכם ביחס לאחרים וכן לאפשר לכם לזהות באילו מקרים האלג' שלכם מגיע לתוצאות טובות ובאילו מקרים נדרש לשפר.
6. רכזו את כל התוצאות שלכם (המיטביות) בדוח קצר בנושא - והוסיפו אותו לתיעוד.
7. שלבו בחלק זה גם את החלק הראשון של המטלה - לפי הצורך הקפידו לעדכן את החלק הראשון (אם נדרש).
8. לאחר שהשלמתם את המטלה, הכינו קובץ דחוס אחד שכולל את הפרויקט שלכם הן את הקוד והן את הדוח (אלג' + תוצאות) - והעלו הכל למודל.

בהצלחה!