

חומרים שהוכנו על-ידי משתתפי קורס מורים מובילים תשע"א

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי

ללא קבלת אישור מראש מצוות הפיתוח

היבטים שונים של ייצוג מבני נתונים בעזרת מערך של תורים / מערך של מחסניות

תוך שימת דגש על גנריות מבני הנתונים

שאלה 2 – טלפון סולרי FreePhone

שאלה 3 – בניין "חנייה לגובה"

שאלה 4 – סופרמרקט "מוצר-לכל"

כתיבה ועריכה:

שרה וגנר

דורית ליקרמן

אהובה שפרלינג

שאלה 2

FixedPriorityStack היא "מחסנית-עדיפויות" המכילה איברים שלכל אחד מהם מוגדרת עדיפות. כל איבר מוכנס ל"מחסנית-עדיפויות" בהתאם לעדיפות שיש לו. הוצאת איבר מ"מחסנית-עדיפויות" מוציאה את האיבר בעל העדיפות הגבוהה ביותר. אם יש ב"מחסנית-עדיפויות" מספר איברים בעלי אותה עדיפות, יוצא זה שהוכנס אחרון ל"מחסנית-עדיפויות".

"מחסנית-עדיפויות" n היא "מחסנית-עדיפויות" בעלת n דרגות עדיפות שונות.
לדוגמא:

"מחסנית-עדיפויות" 5 – בעלת 5 עדיפויות כך שהעדיפות הגבוהה ביותר היא- 5 והנמוכה ביותר היא 1.

לפניך חלק מממשק הפעולות של המחלקה FixedPriorityStack:

| | |
|--|---|
| FixedPriorityStack<T>(int n) | פעולה המקבלת מספר n המייצג את מספר העדיפויות השונות ב"מחסנית-עדיפויות", ומחזירה FixedPriorityStack ריק. |
| void insertFixedStack (T item, int priority) | פעולה המוסיפה את item ל-FixedPriorityStack בהתאם ל-priority שלו |
| T removeFixedStack() | פעולה המוציאה מ-FixedPriorityStack את האיבר בעל העדיפות הגבוהה ביותר שהוכנס אחרון ל"מחסנית-עדיפויות" ומחזירה אותו. אם "מחסנית-עדיפויות" ריק, יוחזר null |
| boolean emptyFixedStack() | פעולה המחזירה "אמת" אם "מחסנית-עדיפויות" ריק, אחרת מחזירה "שקר" |

א. כתוב את כותרת המחלקה הגנרית FixedPriorityStack, רשום את התכונה/ות שלה והסבר בחירתך.

ב. ממש את פעולת הבנאי FixedPriorityStack<T>(int n) בהתאם להגדרה שכתבת בסעיף א'.

חברת FreePhone משיקה טלפון סלולרי חדש. בטלפון החדש לכל איש קשר מוגדרת עדיפות. (משפחה, חברים, עמיתים לעבודה, לא מזוהה).

יומן השיחות בטלפון חדש זה מציג את השיחות שלא נענו על פי ימים (מהיום הנוכחי אחורה) כך שהשיחה האחרונה שלא נענתה ביום הנוכחי, בעלת העדיפות הגבוהה ביותר מופיעה ראשונה על הצג, ואחריה כל יתרת השיחות של היום הנוכחי על פי סדר העדיפויות, ואחר"כ יתרת השיחות מהימים הקודמים, לפי סדר הימים מהיום הנוכחי אחורה, ובתוך כל יום על פי סדר העדיפויות (מהגבוהה לנמוכה).

לשם כך הוגדרו המחלקות הבאות: Caller – שיחה שלא נענתה, Date - תאריך.

ג. (1) כתוב את כותרת המחלקה Caller, את התכונות שלה ורשום תיעוד ליד כל תכונה.

(2) כתוב את כותרת המחלקה Date, את התכונות שלה ורשום תיעוד ליד כל תכונה.

בנוסף, הוגדרה המחלקה יומן-שיחות-שלא-נענו - NonAnsweredCalls.

ד. (1) כתוב את כותרת המחלקה, את התכונות שלה ורשום תיעוד ליד כל תכונה.

(2) ממש במחלקה זו פעולה המוסיפה שיחה שלא נענתה ליומן השיחות.

```
public void insertNAC (Caller c1)
```

יומן השיחות שלא נענו שומר רק את השיחות שלא נענו שבעה ימים אחרונים.
כל השיחות שלא נענו הישנות יותר, מועברות לארכיון השיחות הבנוי כמו יומן השיחות שלא נענו.

ה. ממש במחלקה NonAnsweredCalls פעולה שמעדכנת את יומן השיחות שלא נענו ומחזירה את השיחות שלא נענו שאותן יש להעביר לארכיון.

```
public NonAnsweredCalls oldCalls(Date currentDate)
```

הנח שכל פעולות ה- get/set ב- Caller ו- Date מוגדרות וניתן להשתמש בהן מבלי לממש אותן.

ו. מהי יעילות (סיבוכיות) הפעולה שכתבת בסעיף ה' ? הסבר את תשובתך.

שאלה 3

"חניה-לגובה" הוא מבנה חניה חדשני שבו המכוניות חונות אחת מעל השניה. מעלית ממוחשבת מעלה ומורידה מכונית אל החניה וממנה. במבנה "חניה-לגובה" המכוניות ממוקמות על פי שעת היציאה מהחניון. מכונית בעלת שעת יציאה מאוחרת יותר חונה מעל מכונית ששעת היציאה שלה מוקדמת יותר.

בשל מצוקת החנייה בעיר "שמחה" ניתן היתר למר "שמח" לבנות "חניון-לגובה" ממוחשב של 15 "חניות- לגובה", שיפעל 24 שעות ביממה.

לשם כך הוגדרה המחלקה Time.
לפניכם ממשק חלקי של המחלקה Timer:

| פעולה בונה | Time(int hour, int minutes) |
|---|------------------------------|
| מקבלת שעה ומחזירה "אמת" אם השעה t לפני השעה הנוכחית (של העצם המפעיל) ו"שקר" אם מאוחרת יותר מהשעה הנוכחית. | boolean early(Time t) |

בהגיעו ל"חניון-לגובה" מקיש הנהג את מספר הרכב, שעת הכניסה ושעת היציאה שלו מן ה"חניון-לגובה". המערכת הממוחשבת החכמה מדפיסה פתקית עם מספר ה"חניה-לגובה" אליה עליו להסיע את מכוניתו.

המערכת הממוחשבת בוחרת "חניה-לגובה" שבה מספר המכוניות שיש להוציא (ולהחזיר) כדי למקם את המכונית במקום המתאים על פי שעת יציאתה מה"חניון-לגובה" הוא מינימלי.

המערכת מאפשרת לבעל מכונית שמכוניתו כבר נמצאת ב"חניון-לגובה", לעדכן את שעת היציאה, פעולה היכולה לשנות את מיקומה של המכונית ב"חניון-לגובה", ולהדפיס לנהג פתקית חדשה עם מספר "חניה-לגובה" חדש.

א. רשום את כותרת המחלקה "חניה-לגובה" - HighParking ואת התכונה/ות שלה. מכונית מיוצגת על ידי מספר הרכב ושעת יציאה.

```
public class Car
{
    private String carNum;
    private Time goingOut;
}
```

- ב. רשום את כותרת המחלקה "חניון-לגובה" – HighParkingPool ואת התכונה/ות שלה.
- ג. ממש במחלקה "חניון-לגובה" את פעולת הבנאי.
- ד. ממש במחלקה "חניון-לגובה" את הפעולה

`insertCarToPool(String carNumber, Time goingIn, Time goingOut)`

הפעולה מקבלת מספר מכונית, שעת כניסה ושעת יציאה ומכניסה את המכונית ל"חניון-לגובה" לפי התיאור לעיל.

הנח שכל פעולות ה- get/set במחלקות "חניה-לגובה", "חניון-לגובה", Car ו- Hour נתונות.

- ה. מהי סיבוכיות האלגוריתם שכתבת בסעיף ד'.

שאלה 4

בסופרמרקט "מוצר לכל" 7 "עמדות-תשלום".
"עמדת-תשלום" מאופיינת על ידי מספר עמדה, סוג העמדה (רגילה / מהירה), סטטוס העמדה (פעילה / לא פעילה) וקונים הממתינים לתשלום.
קונים שסיימו את קנייתם בסופר, ניגשים לאחת מ"עמדות-התשלום" הפעילות, וממתינים לתורם.
לשם כך, הוגדרה מחלקה Customer – קונה, בעלת התכונות: שם הקונה – name ומספר הפריטים בעגלת הקניה – itemsNumber.
הנח שכל פעולות ה- set/get במחלקה Customer מוגדרות.

כמו כן הוגדרה המחלקה "עמדת-תשלום" – PaymentStand.

לפניך ממשק חלקי של המחלקה עמדת תשלום:

| | |
|--------------------------------|---|
| PaymentStand(int number) | הפעולה מקבלת מספר עמדה ובונה עמדת תשלום רגילה, לא פעילה, ללא קונים ממתינים. |
| void setType() | הפעולה משנה את סוג העמדה מרגיל למהיר ולהיפך. |
| void setStatus() | הפעולה משנה את הסטטוס מפעיל ללא פעיל ולהיפך. |
| void insertToStand(Customer c) | הפעולה מוסיפה את הקונה c לממתינים בעמדה. |
| Customer removeFromStand() | פעולה המוציאה את הממתין הראשון מבין הממתינים לתשלום בעמדה הנוכחית. |

בנוסף הוגדרו כל פעולות ה- set/get של המחלקה – PaymentStand.

א. רשום את כותרת המחלקה "עמדת תשלום" – PaymentStand, את תכונותיה ורשום תיעוד ליד כל תכונה.

במהלך יום העבודה, נפתחות ונסגרות "עמדות-תשלום" על פי עומס הקונים, ולפי הצורך נפתחות גם "עמדות-תשלום" מהיר. (המיועדות לקונים שבעגלת הקניות שלהם לכל היותר 10 פריטים).
כאשר העומס בסופרמרקט גדול, ונפתחת "עמדת-תשלום" מהיר עובר לעמדה זו הקונה הממתין ב"עמדת-תשלום" רגילה בעלת מספר הממתינים הגדול ביותר, שבעגלתו לא יותר מ- 10 פריטים. אם בין הממתינים בעמדה זו מספר קונים העונים על קריטריון זה, עובר ל"עמדת-התשלום" המהיר הממתין הראשון שהגיע לעמדה.

הנח, כי בעמדה שבה מספר הממתינים הגדול ביותר, יש לפחות קונה אחד העונה לקריטריון.

ב. (1) רשום את כותרת המחלקה "עמדות-תשלום" - PaymentStandPool,

את תכונותיה ורשום תיעוד ליד כל תכונה.

(2) ממש את פעולת הבנאי במחלקה "עמדות-תשלום".

ג. ממש במחלקה "עמדות תשלום" את הפעולה

`moveToQuickStand(int quickStand)`

הפעולה מקבלת מספר "עמדות-תשלום" מהירה שנפתחה.
הפעולה מדפיסה את שם הקונה שעובר ל"עמדות-תשלום" מהיר זו, ומעדכנת את
"עמדות-התשלום" בהתאם.
הנח שכל פעולות ה- `get/set` במחלקות השונות מוגדרות.

ד. מהי סיבוכיות האלגוריתם שכתבת בסעיף ג'.