

זמן המבחן 3 שעות + ½ שעה מנהלתית. יש לענות על כל 3 השאלות ע"פ ההוראות. בהצלחה.

לאחר הורדת המבחן ממערכת הבדיקות, פיתחו פרויקט Java **ובתוכו package בשם test**. העתיקו לשם את קובצי המקור.

שאלה 1 – שימוש במבני נתונים, Stream ועיבוד נתונים (30 נק')

נתונה לכם המחלקה Employee (אשר אינה לעריכה או להגשה) המייצגת עובד. לכל עובד יש שם, גיל ומשכורת. בקובץ Q1.java עליכם לענות על הסעיפים א, ב, ג, במתודות a, b, c בהתאמה.

עבור סעיפים א' ו ג' חל איסור להפוך את ה stream בחזרה לרשימת עובדים

- ממשו את המתודה a כך שבהינתן stream של עובדים היא תחזיר את המשכורת החציונית – כלומר משכורת (של עובד כלשהו שקיים ב stream) כך שבדיוק לחצי מהעובדים יש משכורת גדולה או שווה ממנה ולחצי מהעובדים משכורת קטנה ממש ממנה. (10 נק')
- ממשו את המתודה b כך שבהינתן רשימה של עובדים, היא תחזיר רשימת עובדים שמשכורתם גדולה ממש מהמשכורת הממוצעת. על הרשימה להיות ממוינת מהגיל הקטן לגדול. (10 נק')
- ממשו את המתודה c כך שבהינתן stream של עובדים היא תחזיר את העובד עם השכר הגבוה ביותר ששמו מתחיל ב 'a' (10 נק')

מוד האימון ומוד ההגשה זהים (הקלט אקראי). שאלה זו **תיבדק באופן אוטומטי בלבד**, ולכן עליכם לוודא שהקוד מתקמפל ורץ ללא שגיאות גם אם לא עניתם או הצלחתם לענות על אחד הסעיפים.

שאלה 2 – מחרוזות, מבני נתונים (40 נק')

טבלת עובדים היא מחרוזת גדולה המכילה את פרטי העובדים. לדוגמה:

name	age	salary
abc	18	10000.0
xyz	25	12000.0
ABC	32	25000.0

היא מתחילה בכותרות העמודות name, age, salary מופרדות ע"י טאב (\t) ולאחר מכן ירידת שורה (\n) בלבד). לאחר מכן בכל שורה מופיעים נתוניו של עובד כלשהו. גם כאן השדות מופרדים בטאב, ולאחר כל שורה יש \n בלבד.

ברצוננו לתרגם טבלאות עובדים לרשימה של אובייקטי Employee ולהיפך.

עליכם לממש במחלקה EmpTable את המתודות הסטטיות הבאות:

- המתודה getTable – בהינתן רשימת עובדים היא תחזיר מחרוזת של טבלת עובדים
- המתודה getEmps – בהינתן מחרוזת של טבלת עובדים היא תחזיר רשימת עובדים
- המתודה query – בהינתן מחרוזת של טבלת עובדים ופרדיקט, היא תחזיר רשימת עובדים שהפרדיקט מחזיר עליהם אמת.
- המתודה intersection – בהינתן שתי מחרוזות של טבלאות עובדים, היא תחזיר מחרוזת של טבלת עובדים שמייצגת את החיתוך בין שתי הטבלאות.

שאלה 3 – ביטויי למבדה (30 נק')

במחלקה Interpreter עליכם לממש את המתודות הסטטיות הבאות. כל אחת מהן מקבלת כמה פרמטרים שנרצה, ללא הגבלה, מסוג $\text{Predicate}<T>$, ותחזיר לנו $\text{Predicate}<T>$ חדש אשר משלב את הפרדיקטים ע"פ ההגדרות הבאות.

- המתודה `and` – תחזיר פרדיקט חדש אשר נוצר מפעולת `and` על כל הפרדיקטים שנתקבלו כפרמטר
- המתודה `or` – תחזיר פרדיקט חדש אשר נוצר מפעולת `or` על כל הפרדיקטים שנתקבלו כפרמטר
- המתודה `xor` – תחזיר פרדיקט חדש אשר נוצר מפעולת `xor` על כל הפרדיקטים שנתקבלו כפרמטר. כלומר תחזיר אמת אם ורק אם אחד הפרדיקטים החזיר אמת וכל השאר שקר.
- המתודה `nand` – תחזיר פרדיקט חדש אשר נוצר מפעולת `nand` על כל הפרדיקטים שנתקבלו כפרמטר. כלומר היא תחזיר את השלילה של פעולת `and` על כולם.
- המתודה `nor` – תחזיר פרדיקט חדש אשר נוצר מפעולת `nor` על כל הפרדיקטים שנתקבלו כפרמטר. כלומר היא תחזיר את השלילה של פעולת `or` על כולם.
- המתודה `xnor` – תחזיר פרדיקט חדש אשר נוצר מפעולת `xnor` על כל הפרדיקטים שנתקבלו כפרמטר. כלומר היא תחזיר את השלילה של פעולת `xor` על כולם.

שימו לב!

כל המתודות לעיל לא מחשבות תוצאה ישירה, אלא מחזירות פרדיקט חדש שיחשב את התוצאה בבוא העת כאשר יקבל פרמטר מסוג `T`.

לדוגמה:

```
Predicate<Double> pd=Interpreter.and(x->x>5.0 , x->x<10.5 , x->x!=7);
pd.test(6.0); // true
pd.test(7.0); // false

Predicate<Integer> pi=Interpreter.or(x->x<5 , x->x>10);
pi.test(6); // false
pi.test(11); // true

Predicate<String> ps =
    Interpreter.nor(s->s.startsWith("a"), s->s.length()==3, s->s.contains("B"));
ps.test("xyz"); // false
ps.test("bcdf"); // true
```

ניתן לראות מהדוגמאות לעיל

- המתודות של `Interpreter` יכולות לקבל כמה פרמטרים שנרצה מסוג פרדיקט, מופרדים בפסיק.
- קבלנו בכל פעם מופע של פרדיקט חדש שמורכב מהגדרות הפרדיקטים שניתנו כפרמטר
- אפשר להפעיל אותו (`test`) בכל פעם עם פרמטר שונה ולקבל תוצאה בוליאנית בהתאם.

מוד האימון ומוד ההגשה זהים (הקלט אקראי). שאלה זו **תיבדק באופן אוטומטי בלבד**, ולכן עליכם לוודא שהקוד מתקמפל ורץ ללא שגיאות.

הגשה

עליכם להיכנס למערכת הבדיקות למבחן במודול, או בכתובת: <https://cktest.cs.colman.ac.il/> ולהגיש ל PTM1 ומועד ב' את הקבצים Q1.java, EmpTable.java, Interpreter.java. בכל הגשה יש להגיש את כל הקבצים (ולהתייחס לפלט רק של השאלות שעניתם עליהן). ניתן להגיש במוד אימון ובמוד הגשה כמה פעמים שתמצאו עד לסוף המבחן. בסוף המבחן יש להגיש במוד הגשה ואז במוד הגשה סופית. אחריה תקבלו מס' אסמכתא בן 4 ספרות. לאחר הגשה במוד זה לא תוכלו להגיש יותר.

אם מתעוררות שאלות

תוכלו לצפות בסרטון המצורף לקובצי המבחן בו אני מסביר את טופס הבחינה. אם בכל זאת מתעוררת שאלה תוכלו לשלוח מייל לכתובת: oop.course.colman@gmail.com ניתן מענה לשאלות הבהרה בלבד – מה לא מובן בטופס הבחינה. בפרט, לא נענה על שאלות בסגנון של "יש לי באג", "עובד לי במוד אימון אך לא בהגשה" וכדומה. עליכם להתמודד עם מימוש הקוד בכוחות עצמכם.

יושרה אקדמית

אנו מצפים מכם לשמור על יושרה אקדמית ולא להעתיק, להיעזר או לעזור לנבחנים אחרים. מתבצעת בדיקת העתקות אוטומטית לאחר סיום המבחן, ובמקרה של חשד גם העוזר וגם הנעזר יעלו לוועדת משמעת. בפרט, אין "להשוות תשובות" ביום המבחן – גם לאחר שסיימתם. מהרגע ששלחתם תשובות למישהו אין לכם שליטה לאן זה יגיע, יש סטודנטים שעדיין נבחנים ולא תרצו למצוא את עצמכם כחשודים בהעתקה. בסופו של יום זהו אינטרס שלכם שכל סטודנט יקבל את הציון המאפיין אותו לאחר שהתאמץ עבורו.

בהצלחה!