

# המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

ת.ז. הסטודנט: _____ מספר חדר: _____ מספר נבחן: _____ מספר אסמכתא: _____	ברקוד נבחן
--	------------

## מבחן בקורס: תכנות פיתוח תוכנה מתקדם 1

תאריך הבחינה: 06.02.19

שנת הלימודים: תשע"ט, סמסטר: א', מועד: א'.

משך הבחינה: 4 שעות

שם המרצה/ים:	שם המתרגל/ים:
ד"ר אליהו חלסצ'י	

מבנה הבחינה: הבחינה מורכבת מחלק אחד.

מספר השאלות בבחינה: שתי שאלות תכנותיות

משקל כל שאלה: בצמוד לכל שאלה

הוראות לנבחן:

- מותר השימוש בכל חומר עזר לרבות חומר מודפס, עמדת המחשב במעבדה, דיסק און קי.  
○ אסור השימוש בכל אמצעי תקשורת לרבות מחשב אישי, מחשב לוח או טלפון.
- יש לענות במחשב
- לא נדרש להחזיר את השאלון
- לא מצורף נספח לבחינה
- מחברת טיוטה: אין
- מחברת נפרדת לכל שאלה: לא
- למבחן בדיקה אוטומטית ולכן יש להגיש ע"פ הוראות ההגשה במדויק!!!
- שאלה שלא תתקמפל או שתהיה לה שגיאת ריצה ציונה אוטומטית 0. אין בדיקה ידנית למבחן. יש מספיק זמן למבחן כדי לנפות את כל השגיאות.
- אזהרה: למבחן מתקיימת בדיקת העתקות אוטומטית ובעבר סטודנטים שהעתיקו הועלו לוועדת משמעת והורחקו מהמכללה לשנה. אל תעתיקו במבחן.

**בהצלחה!!**

# המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

## שאלה 1 (50 נקודות) מבני נתונים, streaming

נתונה המחלקה WordCounter שתפקידה לספור את כמות ההופעות של כל מילה בקובץ טקסט כלשהו.

- א. השלימו את הבנאי כך שבהינתן שם קובץ, ו HashSet של מילים שיש להתעלם מהן, הבנאי יחשב כמה פעמים כל מילה מופיעה בקובץ. כמובן, הספירה לא תכלול את המילים שמופיעות ב HashSet.
- ב. בהינתן מילה המתודה getCount תחזיר ב  $O(1)$  כמה פעמים המילה הופיעה בקובץ.
- ג. המתודה getOrderedList תחזיר רשימה של כל המילים שהופיעו בקובץ. על הרשימה להיות ממוינת על פי מספר ההופעות שלהן מהגדול לקטן.

**הערה:** מילים יכולות להיות מופרדות ע"י כל white space או רצפים שלהם, למשל כמה רווחים, טאבים ירידות שורה וכדומה. לדוגמא בטקסט הבא:

Hello world and one-one-one

a Happy hello world to you

המילה world מופיעה פעמיים ושאר המילים פעם אחת. נשים לב שגם a היא מילה, ושיש הבדל בין אותיות גדולות לקטנות, ושכל רצף אותיות (ללא white spaces) מהווה רק מילה אחת למשל one-one-one.

**אופי הבדיקה:** תחילה, שימרו במקום כלשהו במחשב את הקובץ הנלווה file1.txt.

המתודה test1 של MainTrain

- יצרה HashSet של מספר מילים נפוצות שיש להתעלם מהן.
  - יצרה מופע של WordCounter.
  - **כמובן עליכם להתאים את שם הקובץ למיקום שבו שמרתם אותו.**
  - בדקה שהמילה the מופיעה 0 פעמים (כי בקשנו להתעלם ממנה)
  - בדקה את מספר ההופעות של מילים אחרות
  - בדקה את 3 המילים הנפוצות ביותר ברשימה הממוינת שחזרה
- במוד הגשה יבדקו טקסטים שונים ומילים שונות אך אופי הבדיקה דומה.

## שאלה 2 (50 נק') תבניות עיצוב

נתונה המחלקה A שמחזיקה state מסוג int. מותר לכם לערוך מחלקה זו על פי הצורך.

עליכם להשלים את המימוש של MyPriorityQueue כך שיהווה תור עדיפויות של A-ים (מהקטן לגדול) כך שאם מופע כלשהו משנה את ה state שלו, אז אוטומטית תור העדיפויות יעדכן את המיקום שלו בתור.

**אופי הבדיקה:**

המתודה test2 ב MainTrain

- מכניסה לתור מופעים של A בלי סדר מסוים
- לאחר הכנסתם לתור, המופעים של a1,a2,a3 משנים את ה state שלהם
  - תור העדיפויות צריך בשלב זה לעדכן את המיקום שלהם אוטומטית
- שליפה של כל האיברים בזה אחר זה מהתור בודקת שהם אכן ממוינים מהקטן לגדול.

**הגשה:** עליכם להגיש ל <https://ck.cs.colman.ac.il/test.jsp> לקורס PTM1\_semester\_A תיבה moed\_a את הקבצים WordCounter.java, MyPriorityQueue.java ו A.java ואותם בלבד. הקפידו על הוראות ההגשה כפי שכתוב בקובץ המצורף. **בהצלחה!**