

# המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

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

## מבחן בקורס: תכנות פיתוח תוכנה מתקדם 1

תאריך הבחינה: 05.03.19

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

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) תבניות עיצוב, מבנה נתונים

נתונה המחלקה Point המייצגת נקודה תלת-ממדית. מחלקה זו לא ניתנת לעריכה ואינה הגשה. כמו כן, נתונים הממשק DistCalculator המגדיר מתודה לחישוב מרחק בין שתי נקודות, וכן המחלקה EuclideanDistance שממשה את הממשק ומחזירה מרחק אוקלידי. גם טיפוסים אלו לא ניתנים לעריכה ואינם להגשה. עליכם לממש את המחלקה EucDistProxy כך שתהווה Proxy של EuclideanDistance ע"פ תבנית העיצוב Proxy Pattern. אופן הפעולה הנדרש של ה Proxy הוא שמירה של חישובים שכבר בוצעו בעבר כך שישלפו מהזיכרון ב O(1). אחרת, הוא יזום את פעולתו של ה EuclideanDistance וישמור את התוצאה להבא.

#### **אופי הבדיקה:**

הביטו ב MainTrain1.

- נחשב באמצעות EucDistProxy את המרחק בין הנקודה (0,0,0) ל (3,4,5) ונבדוק את התוצאה ושאכן EuclideanDistance הופעל רק פעם אחת. הרי זו הפעם הראשונה שבה מבוצע החישוב הזה.
- נחזור על הבדיקה כדי לוודא ש EuclideanDistance לא הופעל שוב, אלא התוצאה נשלפה מהזיכרון.
- כעת נבדוק שגם המרחק ההפוך בין הנקודה (3,4,5) ל (0,0,0) נשלף מהזיכרון ו EuclideanDistance לא הופעל שוב.

אופי הבדיקה של מוד ההגשה דומה. בבדיקה ננסה מספר גדול יותר של חישובים ונוודא ש EuclideanDistance מופעל רק כאשר נבקש חישוב חדש.

### שאלה 2 (50 נק') תקשורת

סעיף א' (25 נק')

בקובץ DistServer.java ממשו את המתודה start (העליונה) כך שבהינתן port ובהינתן פונקציה f מ String ל String (עליכם להשלים את הטיפוס החסר) אז המתודה

- תפתח שרת המאזין על port
- תקרא שורה אחר שורה מהלקוח
  - על כל שורה שהתקבלה מהלקוח תופעל הפונקציה f וערך החזרה שלה יוחזר ללקוח
  - כל זאת עד להופעת השורה "bye" שעבורה לא תחזיר דבר ללקוח.
- תסגור את כל מה שצריך ותסיים את פעולתה

#### הערה: הקפידו על ירידות שורה ושימוש ב flush

למען הסר ספק, השרת מטפל אך ורק בלקוח אחד ומסיים את עבודתו. אין צורך לפתוח Thread

דוגמה להפעלה:

```
DistServer.start(port, s->new StringBuffer(s).reverse().toString());
```

תפעיל את שרת שמחזיר את המחרוזת ההפוכה לזו ששלח לו הלקוח. באותה המידה יכולנו להזריק כל פונקציה שבהינתן מחרוזת תבצע איזשהו חישוב אחר ותחזיר מחרוזת כתוצאה. כך ביצענו הפרדה בין השרת לבין הפונקציה שהשרת מפעיל.

## המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

סעיף ב' (25 נק')

בהמשך המחלקה עליכם לממש את המתודה `start(port)` שתפקידה להפעיל את הפונקציה `start` לעיל (העליונה) יחד עם הפונקציה הבאה:

- בהינתן שורה המורכבת מ 6 ערכי `double` המופרדים ע"י רווח,
- כל 3 ערכים מהווים את  $x, y, z$  של נקודה תלת-ממדית (מופע של `Point`)
  - כך שבכל שורה יש בעצם שתי נקודות תלת-ממדיות
- הפונקציה תחשב ותחזיר (כמחרוזת) את המרחק בין שתי הנקודות שנוצרו
  - את המרחק יש לעגל ל 3 ספרות אחרי הנקודה

### אופי הבדיקה:

לסעיף א' נגדיר נפעיל את השרת עם פונקציה כלשהי, נתחבר כלקוח, ונצפה לקבל את התוצאה הנדרשת.

לסעיף ב' נפעיל את השרת ונשלח סט של 6 ערכים בכל פעם ונצפה לקבל את המרחק המתאים

אופי הבדיקה של מוד ההגשה דומה.

**הגשה:** עליכם להגיש ל <https://ck.cs.colman.ac.il/test.jsp> לקורס `PTM1_semester_A` תיבה `moed_b` את הקבצים `EucDistProxy.java`, `DistServer.java` ואותם בלבד. הקפידו על הוראות ההגשה כפי שכתוב בקובץ המצורף.

**בהצלחה!**