

האוניברסיטה הפתוחה

20554

תכנות מתקדם

בשפת Java

חוברת הקורס – אביב 2019ב

כתבה : תמר בניה

פברואר 2019 – סמסטר אביב- תשע"ט

פנימי – לא להפצה.

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

תוכן העניינים

א	אל הסטודנט
ב	1. לוח זמנים ופעילויות
ד	2. תיאור המטלות
ד	2.1 מידע כללי
ד	2.2 מבנה המטלות ואופן הגשתן
ה	2.3 משקל המטלות ואפיוניהן
ן	3. התנאים לקבלת נקודות זכות
1	ממ"ן 11
5	ממ"ן 12
9	ממ"ן 13
11	ממ"ן 14
13	ממ"ן 15
15	ממ"ן 16

אל הסטודנט,

אנו מקדמים את פניכם בברכה עם הצטרפותכם אל הלומדים בקורס "תכנות מתקדם בשפת Java".

בחוברת זו תמצאו את התנאים לקבלת נקודות זכות בקורס, לוח זמנים ופעילויות ומטלות.

לקורס קיים אתר באינטרנט בו תמצאו חומרי למידה נוספים, בנוסף, האתר מהווה עבורכם ערוץ תקשורת עם צוות ההוראה ועם סטודנטים אחרים בקורס. פרטים על למידה מתוקשבת ואתר הקורס, תמצאו באתר שה"ם בכתובת:

<http://telem.openu.ac.il>

מידע על שירותי ספרייה ומקורות מידע שהאוניברסיטה מעמידה לרשותכם, תמצאו באתר הספרייה באינטרנט www.openu.ac.il/Library.

שעות הייעוץ הטלפוני שלי הן בימי ד' בשעות 11:00 - 13:00 בטלפון 09-7781265 או בדואר אלקטרוני tamar@openu.ac.il. פגישות יש לתאם מראש. לצורך בירורים בנושאים אדמיניסטרטיביים יש לפנות בכתב או טלפונית למחלקות האוניברסיטה הפתוחה.

לתשומת לב הסטודנטים הלומדים בחו"ל:

למרות הריחוק הפיסי הגדול, נשתדל לשמור אתכם על קשרים הדוקים ולעמוד לרשותכם ככל האפשר.

הפרטים החיוניים על הקורס נכללים בחוברת הקורס וכן באתר הקורס. מומלץ מאוד להשתמש באתר הקורס ובכל אמצעי העזר שבו וכמובן לפנות אלינו במידת הצורך.

בברכת לימוד מהנה,

תמר בניה

מרכזת ההוראה בקורס

1. לוח זמנים ופעילויות (20554 / ב2019)

שבוע לימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח ממ"ן (למנחה)
1	1.3.2019-24.2.2019	פרק 1-3	מפגש 1	
2	8.3.2019-3.3.2019	פרק 4-6		
3	15.3.2019-10.3.2019	פרק 7-8	מפגש 2	ממ"ן 11 15.3.2019
4	22.3.2019-17.3.2019 (ה-ו פורים)	פרק 9-10		
5	29.3.2019-24.3.2019	פרק 10-11	מפגש 3	ממ"ן 12 29.3.2019
6	5.4.2019-31.3.2019	פרק 12		
7	12.4.2019-7.4.2019	פרק 12-13	מפגש 4	
8	19.4.2019-14.4.2019 (ו ערב פסח)	פרק 14-15		ממ"ן 13 18.4.2019
9	26.4.2019-21.4.2019 (א-ו פסח)	פרק 16	מפגש 5	

* התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".

לוח זמנים ופעילויות - המשך

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
10	3.5.2019-28.4.2019 (היום הזכרון לשואה)	פרק 20 - 21		
11	10.5.2019-5.5.2019 (ד' יום הזיכרון) (ה' יום העצמאות)	פרק 21 - 22	מפגש 6	ממ"ן 14 10.5.2019
12	17.5.2019-12.5.2019	פרק 23		
13	24.5.2019-19.5.2019 (ה' ל"ג בעומר)	פרק 23	מפגש 7	ממ"ן 15 24.5.2019
14	31.5.2019-26.5.2019	פרק 28		
15	7.6.2019-2.6.2019	פרק 28	מפגש 8	ממ"ן 16 7.6.2019
16	14.6.2019-9.6.2019 (א' שבועות)			

מועדי בחינות הגמר יפורסמו בנפרד

* התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".

2. תיאור המטלות

קראו היטב עמודים אלו לפני שתתחילו לענות על השאלות

2.1 מידע כללי

על מנת לתרגל את החומר הנלמד ולבדוק את מידת הבנתכם, עליכם לפתור את המטלות המצורפות. פתרון המטלות הוא חלק בלתי נפרד מלימוד הקורס - הבנה מעמיקה של חומר הלימוד דורשת תרגול רב. המטלות ייבדקו על-ידי המנחה ויוחזרו אליכם בצירוף הערות המתייחסות לתשובות.

לתשומת לבכם!

כדי לעודדכם להגיש לבדיקה מספר רב של מטלות הנהגנו את ההקלה שלהלן:

אם הגשתם מטלות מעל למשקל המינימלי הנדרש בקורס, **המטלות** בציון הנמוך ביותר, שציוניהן נמוכים מציון הבחינה (**עד שתי מטלות**), לא יילקחו בחשבון בעת שקלול הציון הסופי.

זאת בתנאי שמטלות אלה **אינן חלק מדרישות החובה בקורס** ושהמשקל הצבור של המטלות האחרות שהוגשו, מגיע למינימום הנדרש.

זכרו! ציון סופי מחושב רק לסטודנטים שעברו את בחינת הגמר בציון 60 ומעלה והגישו מטלות כנדרש באותו קורס.

2.2 מבנה המטלות ואופן הגשתן

כל מטלה מורכבת מכמה שאלות. בראש כל שאלה מצוין משקלה היחסי בקביעת ציון המטלה. את הפתרונות למטלה עליכם להריץ במחשב. יש להגיש את המטלות דרך מערכת המטלות האלקטרונית באופן הבא:

מטלה תהייה מורכבת מקובץ אחד המכונה בפורמט ZIP. הקובץ יכיל ספריות כך שכל פתרון לשאלה יהיה בספרייה נפרדת. מלבד הספריות האלה אין ליצור ספריות נוספות. כל ספרייה תכיל: קבצי מקור (עם סיומת java), קבצי הרצה (עם סיומת class), קובץ הרצה (run.bat) כפי שיתואר בהמשך.

לחיצה כפולה על קובץ ההרצה תפעיל את התוכנית, יש לוודא את תקינות קובץ ההרצה לפני ההגשה. אין לצרף קבצים מיותרים (קבצים המתווספים על ידי סביבות עבודה).

קובץ ההרצה (run.bat) יכיל את השורות :

```
java ClassFileName  
pause
```

אין צורך לכלול בקובץ ההרצה את פקודת ה**היזור** `javac SourceFileName`

אם השאלה בממ"ן אינה ברורה לכם, אל תהססו להתקשר אל אחד המנחים (בשעות הייעוץ הטלפוני) לצורך קבלת הסבר.

2.3 משקל המטלות ואפיוניהן

מספר מטלה	הפרק שאליו מתייחסת המטלה	משקל המטלה
11	עד פרק 7 כולל	4
12	עד פרק 11 כולל	4
13	עד פרק 15 כולל	4
14	עד פרק 22 כולל	4
15	עד פרק 23 כולל	4
16	עד פרק 28 כולל	4

לתשומת לבכם:

מדיניות קורס זה היא לאשר הזנת ציון אפס במטלות שלא הוגשו כנדרש בקורס. סטודנטים אשר לא הגישו את מכסת המטלות המינימאלית לעמידה בדרישות הקורס ולקבלת זכאות להיבחן, ומבקשים שמטלות חסרות יוזנו בציון אפס, יפנו למוקד הפניות והמידע בטלפון 09-7782222 או יעדכנו בעצמם באתר שאילתא <http://www.openu.ac.il/sheilta>

קורסים ← ציוני מטלות ובחינות ← הזנת ציון 0 למטלות רשות שלא הוגשו. יש לקחת בחשבון כי מטלות אשר יוזן להן ציון אפס ישוקללו בחישוב הציון הסופי ובכך יורידו ציון זה ולא ניתן יהיה להמירן במטלות חלופיות במועד מאוחר יותר. על כן קיימת אפשרות שסטודנט אשר יעבור את הבחינה בהצלחה ייכשל בקורס (כשהממוצע המשוקלל של המטלות והבחינה יהיה נמוך מ-60).

כלל זה איננו חל על מטלות חובה או על מטלות שנקבע עבורן ציון מינימום.

3. התנאים לקבלת נקודות זכות

כדי לקבל נקודות זכות בקורס זה עליכם לעמוד בדרישות הבאות:

- א. הגשת 4 מטלות לפחות.
- ב. ציון של לפחות 60 נקודות בבחינת הגמר.
- ג. ציון סופי בקורס של 60 נקודות לפחות.

שימו לב:

פרקים נבחרים מספר הקורס:

Deitel & Deitel, *Java: How to Program*, 10th ed. (Prentice Hall, 2015)

משמשים גם כחומר לימוד בסדנה בתכנות מתקדם בשפת Java 20503. אם הנכם מתכוונים בעתיד ללמוד את הסדנה אנא שימרו על הספר כך שיוכל לשמש אתכם גם בסדנה.

מטלת מנחה (ממ"ן) 11

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 7 (כולל)

משקל המטלה: 4 נקודות

מספר השאלות: 2

מועד אחרון להגשה: 15.3.2019

סמסטר: 2019ב

קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
 - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

שאלה 1 (50 נקודות)

כתבו יישום המממש את משחק ניחושי המילים הבא:

- המחשב בוחר מלה אקראית מתוך מאגר מילים ומציג סדרת קווים כאורך המילה. כל קו מייצג אות אחת.
 - המשתמש מנסה לנחש את המלה על ידי כך שבכל תור הוא בוחר אות אחת מתוך הא"ב. אם האות מופיעה במילה, היא תוצג במקומה המתאים בסדרת הקווים. אם האות מופיעה ביותר ממקום אחד, היא תוצג בכל המקומות בהם היא מופיעה.
 - המשתמש ממשיך לנחש אות אחר אות עד שהוא מנחש את המלה כולה.
 - לבסוף יש להציג למשתמש את מספר הניחושים ולאפשר לו להתחיל משחק חדש.
- ממשק המשתמש ימומש באמצעות תיבת דו שיח שתכלול את האלמנטים הבאים:
- המלה אותה מנחשים (מיוצגת בהתחלה בתור סדרת קווים),
 - מחרוזת המייצגת את אותיות הא"ב שעדיין לא נבחרו,
 - שדה קלט המאפשר למשתמש לבחור אות אחת. במקרה שהמשתמש מקיש קלט לא חוקי, התכנית תתעלם מהקלט.

הדרכה:

- אפשר לאתחל את מאגר המילים בתוך התכנית
- הגדירו מחלקות מתאימות המייצגות את הישויות השונות: מאגר המילים, המילה הנבחרת, ומחלקת המשחק.
- העזרו ב-API של המחלקה String כדי למצוא מתודות מתאימות לטיפול במחרוזות.

שאלה 2 (50 נקודות)

הגדרה: ניתן להגדיר P/Q כמספר רציונלי אם P הוא מספר שלם ו- Q הוא מספר חיובי.

א. כתבו מחלקה בשם Rational עבור מספרים רציונלים. המחלקה תכלול את הפעולות הבאות:

– בנאי המקבל שני פרמטרים (מונה ומכנה) ומייצר מספר רציונלי. אם ערכי הפרמטרים אינם תקינים יש לייצר את המספר 0.

– מתודה בוליאנית greaterThan המקבלת מספר רציונלי ובודקת אם המספר (עליו מופעלת המתודה) גדול ממש מהמספר המתקבל כפרמטר.

$$\text{הגדרת "גדול מ-": } \frac{a}{b} > \frac{c}{d} \text{ אם } ad > bc$$

– מתודה בוליאנית equals המקבלת מספר רציונלי ובודקת אם המספר (עליו מופעלת המתודה) שווה למספר המתקבל כפרמטר.

$$\text{הגדרת שוויון: } \frac{a}{b} = \frac{c}{d} \text{ אם } ad = bc$$

– מתודה plus המקבלת מספר רציונלי ומחזירה מספר רציונלי המהווה את סכום המספר והפרמטר.

$$\text{הגדרת סכום: } \frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$$

– מתודה minus המקבלת מספר רציונלי ומחזירה מספר רציונלי המהווה את ההפרש בין המספר והפרמטר (המספר פחות הפרמטר).

$$\text{הגדרת חיסור: } \frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$$

– מתודה multiply המקבלת מספר רציונלי ומחזירה מספר רציונלי המהווה את המכפלה של המספר והפרמטר.

$$\text{הגדרת מכפלה: } \frac{a}{b} \times \frac{c}{d} = \frac{ac}{bd}$$

– מתודה getNumerator המחזירה את המונה.

– מתודה getDenominator המחזירה את המכנה.

– מתודה toString המחזירה ייצוג מחרוזתי של המספר הרציונלי (למשל: 3/4).

– מתודה reduce המחזירה מספר רציונלי המהווה צמצום של המספר. (אין לשנות את המספר המקורי). עליכם לבצע את הצמצום בעזרת מציאת המחלק המשותף הגדול ביותר (GCD). השתמשו באלגוריתם של אוקלידס המוגדר כך:

$$\text{כאשר } y \neq 0 : \gcd(x, y) = \gcd(y, x \bmod y)$$

$$\text{כאשר } y = 0 : \gcd(x, 0) = x$$

ב. כתבו תכנית ראשית המשתמשת במחלקה שכתבתם בסעיף הקודם. התכנית תקלוט מהמשתמש ערכים עבור שני מספרים רציונלים, ותבצע עליהם את פעולות החישוב וההשוואה שהוגדרו בסעיף הקודם.

התכנית תצמצם את התוצאות ותציג את התרגילים בצורה המקובלת. לדוגמה :

$$2/3 * 1/6 = 1/9$$

הערה: ניתן להניח שהמשתמש מזין מספרים שלמים. במקרה שמוזן מכנה 0 יש להציג הודעה מתאימה למשתמש ולבקש שיזין ערך אחר.

מטלת מנחה (ממ"ן) 12

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 11 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 29.3.2019

סמסטר: 2019ב

קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
 - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

שאלה 1 (50 נקודות)

- מספרים שלמים הם בעלי גודל בלתי מוגבל (באופן תיאורטי), כלומר רצף בלתי מוגבל של ספרות. א. כתבו מחלקה בשם `BigInt` המטפלת בשלמים "בלתי מוגבלים" הכוללת את הפעולות הבאות:
- בנאי המקבל מחרוזת ומייצר ממנו מספר בלתי מוגבל. את המספר יש לממש באמצעות `ArrayList` שאיבריו מייצגים את ספרות המספר. שימו לב שגם למספר בלתי מוגבל יש סימן (+ או -). במקרה שהמחרוזת לא מייצגת מספר חוקי יש לעורר מצב חריג מסוג `IllegalArgumentException` המוגדר במארז `java.lang`.
 - מתודת `plus` המקבלת `BigInt` כפרמטר ומחזירה `BigInt` חדש המהווה את סכום המספר והפרמטר. שימו לב לכך שהסכום מתקבל על-ידי סיכום הספרות החל מספרת האחדות, ושחיבור מספר שלילי לחיובי הוא למעשה חיסור.
 - מתודת `minus` המקבלת `BigInt` כפרמטר ומחזירה `BigInt` חדש המהווה את ההפרש בין המספר והפרמטר. שימו לב לכך שההפרש מתקבל על-ידי החסרת הספרות החל מספרת האחדות, ושחיסור מספר גדול ממספר קטן יתבצע על-ידי חיסור המספר הקטן מהגדול ושינוי סימן התוצאה.
 - מתודת `multiply` המקבלת `BigInt` כפרמטר ומחזירה `BigInt` חדש המהווה את מכפלת המספר והפרמטר. שימו לב לכך שהמכפלה מתקבלת על-ידי הכפלת כל ספרה של המספר (עליו הופעלה הפעולה) במספר שהתקבל כפרמטר, וחיבור תוצאות המכפלות השונות (תוך הכפלת כל תוצאת כפל כזו בכפולת 10 המתאימה).
 - מתודת `divide` המקבלת `BigInt` כפרמטר ומחזירה `BigInt` חדש המהווה את המנה המתקבלת מחלוקת המספר בפרמטר (חלוקה בשלמים). שימו לב לכך שאת המנה אפשר לחשב באמצעות פעולת החיבור. במקרה של חלוקה באפס יש לעורר מצב חריג מסוג `ArithmeticException` המוגדר במארז `java.lang`.

- מתודה toString המחזירה מחרוזת המייצגת את המספר הבלתי מוגבל.
- ממשו במחלקה את המתודה equals הנורשת מ-Object. המתודה תבדוק ותחזיר ערך בוליאני המציין אם הפרמטר והמספר עליו הופעלה המתודה שווים.
- ממשו במחלקה את הממשק Comparable כך שהמתודה compareTo (המקבלת BigInt כפרמטר) תחזיר מספר שלילי, אפס, או חיובי; במקרה שהמספר קטן, שווה, או גדול מהפרמטר בהתאמה.
- ב. כתבו במחלקה נפרדת, תכנית ראשית המשתמשת במחלקה BigInt שכתבתם בסעיף הקודם. התכנית תקלוט מהמשתמש שתי מחרוזות המייצגות שני מספרים בלתי מוגבלים, ותבצע עליהם את הפעולות השונות שהוגדרו בסעיף הקודם. התכנית תציג את התרגילים בצורה המקובלת ותטפל במצבים החריגים לפי באופן הבא:
- במקרה של ניסיון ליצור מספר שאינו תקין יש לקלוט מהמשתמש מספר חדש עד שיספק מחרוזת המייצגת מספר חוקי.
- במקרה של חלוקה באפס יש להציג הודעה מתאימה.

שאלה 2 (50 נקודות)

- ברצוננו לכתוב יישום גרפי המדמה עולם של רובוטים שיכולים לנוע בתוך מטריצה.
- א. כתבו מחלקה בשם Robot המייצגת רובוט. לרובוט יהיו התכונות הבאות:
- זיהוי הרובוט – מספר שלם המזהה את הרובוט
- מיקום הרובוט – אובייקט מסוג Position המייצג את קואורדינטות ה-x וה-y של מיקום הרובוט בעולם (ניתן להניח שהקואורדינטות הן מספרים שלמים כלשהם). עליכם להגדיר את המחלקה Position.
- כיוון הרובוט – הרובוט יכול לפנות לאחד מארבעה כיוונים – למעלה, למטה, ימינה ושמאלה. הגדירו ארבעה קבועים ציבוריים סטטיים (LEFT, RIGHT, DOWN, UP) המייצגים כיוונים אלו.
- כתבו במחלקה בנאי המקבל שלושה פרמטרים – זיהוי, מיקום וכיוון – ומאתחל את משתני המופע בהתאם.
- הוסיפו למחלקה את המתודות הבאות:
- public void move() – מתודה המזיזה את הרובוט צעד אחד קדימה בכיוון אליו מסתכל הרובוט.
- public void turnLeft() – מתודה המסובבת את הרובוט ב-90 מעלות שמאלה.
- public void turnRight() – מתודה המסובבת את הרובוט ב-90 מעלות ימינה.
- הוסיפו מתודות גישה לקבלת פרטי הרובוט.
- ב. כתבו מחלקה בשם RobotsWorld המייצגת את עולם הרובוטים כמטריצה ריבועית של תאים. בכל תא יכול להיות רובוט אחד לכל היותר ותנועת הרובוטים מתבצעת מתא לתא בהתאם לכיוון הרובוט. העולם ייוצג באמצעות מטריצה של רובוטים (מסוג Robot שהוגדר לעיל) ומשתנה המייצג את מימדי העולם.

כתבו במחלקה בנאי המקבל כפרמטר את מימדי העולם (מספר חיובי קטן מ-100), ומאתחל אותו כעולם ריק מרובוטים.

הוסיפו למחלקה את המתודות הבאות:

public void addRobot(Position p) – המתודה תקבל כפרמטר אובייקט מסוג Position. המתודה תיצור רובוט במיקום המצויין הפונה לכיוון אקראי כלשהו ותוסיף אותו לעולם. אם המיקום מחוץ לגבולות העולם או במיקום זה כבר קיים רובוט אחר, הרובוט לא יתווסף לעולם, והמתודה תעורר מצב חריג מסוג IllegalPosition. עליכם להגדיר מחלקה זו כתת מחלקה של Exception.

public Robot removeRobot(Position p) – המתודה תקבל כפרמטר אובייקט מסוג Position ותסיר מהעולם רובוט מהמיקום המצויין ותחזיר אותו. אם המיקום מחוץ לגבולות העולם או במיקום זה לא קיים רובוט, המתודה תחזיר null.

public Robot getRobot(Position p) – המתודה תקבל כפרמטר אובייקט מסוג Position ותחזיר את הרובוט שנמצא במיקום המצויין. אם המיקום מחוץ לגבולות העולם או במיקום זה לא קיים רובוט, המתודה תחזיר null.

public void moveRobot(Position p) – המתודה תקבל כפרמטר אובייקט מסוג Position ותזיז את הרובוט הנמצא במיקום המצויין. אם המיקום מחוץ לגבולות העולם או במיקום זה לא קיים רובוט, או שהרובוט לא יכול לזוז כנדרש, המתודה תעורר מצב חריג מסוג IllegalPosition.

ג. כתבו תכנית שתבצע סימולציה לעולם הרובוטים באופן הבא:

צרו עולם רובוטים שמימדיו מתקבלים מתיבת דו-שיח. הציגו את העולם בתוך חלון הכולל פנל שעליו מצוירת מטריצה של תאים בגודל של 20x20 פיקסלים, כאשר בכל אחד מהתאים אפשר למקם לכל היותר רובוט אחד.

מקמו בעולם 5 רובוטים במיקום וכיוון אקראי והציגו אותם בתוך מטריצת העולם. עליכם להציג כל רובוט כעיגול בצבע ייחודי ולסמן את כיוונו באמצעות אחד התווים <, >, ^ או v שיוצג על פני העיגול.

הזיזו את אחד הרובוטים 30 צעדים. אם הרובוט נתקל ברובוט אחר או שהוא נמצא בקצה העולם, עליכם לסובב אותו ימינה ולנסות שוב להזיזו. אם הרובוט עדיין תקוע, יש להמשיך ולסובבו ימינה עד שיוכל להמשיך בדרכו. לאחר כל הזזה של הרובוט יש לעדכן את תצוגת העולם. עדכון התצוגה מתבצע באמצעות קריאה למתודה repaint() של הפנל. עליכם לבצע השהייה של שנייה בין הזזה להזזה כדי שיהיה אפשר להבחין בעדכון התצוגה. את ההשהייה ממשו באמצעות המתודה הסטטית sleep של המחלקה java.lang.Thread המקבלת פרמטר המציין את מספר המילישניות שבהם יש להשהות את התכנית. המתודה מוגדרת באופן הבא:

```
public static void sleep(long millis)
```

```
throws InterruptedException
```

שימו לב שיכול להיות מצב שהרובוט תקוע. עליכם לזהות את המקרה ולסיים את הסימולציה תוך הצגת הודעה מתאימה.

מטלת מנחה (ממ"ן) 13

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 15 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 18.4.2019

סמסטר: 2019ב

קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
 - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

שאלה 1 (50 נקודות)

כתבו יישום המממש את המשחק "איש תלוי" (Hang-man) הפועל באופן הבא:

התכנית בוחרת מלה כלשהי מתוך מאגר מילים, והמשתמש מנסה לנחש את המלה. בכל תור מנחשים אות אחת, אם האות מופיעה במלה, יש להציבה במקומות המתאימים בתוך המלה ולהציגה. למשל המלה "polymorphism" לאחר שהמשתמש ניחש את האותיות "m", "o" ו-"o", תוצג באופן הבא:

_ o _ _ m o _ _ _ _ m

אם הניחוש שגוי התכנית מוסיפה איבר אחד ל"איש התלוי". המשחק נמשך עד שהמשתמש מצליח לנחש את המלה או עד שהציור של ה"איש התלוי" מושלם. בנוסף יש להציג את אותיות הא"ב שנבחרו עד כה. המשחק מסתיים כאשר משלימים את האיש התלוי, או כאשר מצליחים לנחש את המלה. הממשק הגרפי של המשחק יהיה מעוצב לפי שיקול דעתכם, למשל אפשר להשתמש ברכיבים הבאים:

- תוויות - להצגת המילה והאותיות שכבר נבחרו.
- משטח ציור מסוג JPanel - להצגת תרשים האיש התלוי. את האיש התלוי ניתן לצייר באמצעות גרפיקה פשוטה. כדי לעדכן את התרשים יש לקרוא למתודה repaint() של משטח הציור. מתודה זו גורמת למתודה paintComponent להתבצע מחדש ובכך לעדכן את הציור.
- סדרה של כפתורים או כפתור בחירה (JComboBox) לקבלת האות מהמשתמש. התכנית תקרא את מאגר המילים מקובץ. עליכם לאפשר למשתמש להתחיל משחק חדש (מלה חדשה מתוך המאגר). עליכם לתכנן פתרון לפי עקרונות תכנות מונחה עצמים, כלומר עליכם להגדיר את האובייקטים העיקריים ולהרכיב מהם את התכנית.

שאלה 2 (50 נקודות)

- כתבו יישום המממש מחשבון פשוט הכולל את הפעולות חיבור, חיסור, כפל וחילוק.
- המחשבון יאפשר לחשב ביטוי חשבוני המורכב מרצף של פעולות חשבון.
- שימו לב לנקודות הבאות:
- כל מקש מייצג ספרה אחת ולכן יש צורך ליצור מספר המורכב מכמה הקשות רצופות המסתיימות במקש פעולה.
 - המחשבון תומך במספרים שלמים וממשיים.
 - המחשבון תומך במספרים שליליים באמצעות המקש $+/-$ המאפשר לשנות את סימן המספר.
- הממשק הגרפי של המחשבון יכלול מטריצת כפתורים הכוללים את הספרות 0..9 ואת הסימנים $+$, $-$, $*$, $/$, $.$, $+/-$ ו- $=$. בנוסף המחשבון יכיל שדה טקסט המציג את המספר המוקש או תוצאת החישוב.

מטלת מנחה (ממ"ן) 14

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 22 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 10.5.2019

סמסטר: 2019ב

קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
 - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

שאלה 1 (50 נקודות)

א. כתבו מחלקה ג'נרית בשם `SortedGroup` המשמשת להגדרת אוסף של איברים ממוינים בסדר עולה מטיפוס כלשהו `T` המממש את הממשק `Comparable<T>`. את האוסף יש לממש באמצעות מבנה נתונים מהטיפוס `ArrayList`. המחלקה תכלול בנאי ריק שיוצר אוסף ריק. בנוסף יש לכלול ולממש את המתודות הבאות:

- `add` מקבלת איבר ומוסיפה אותו לאוסף במקומו המתאים כך שהמיון יישמר. ייתכן מצב שהאוסף יכלול איברים כפולים.
- `remove` מקבלת איבר ומסירה מהאוסף את כל האיברים השווים (על-פי `equals`) לאיבר הנתון. המתודה תחזיר מספר המציין כמה איברים הוסרו.
- `iterator` המחזירה איטרטור מהטיפוס הג'נרי.

ב. כתבו (במחלקה נפרדת) מתודה ג'נרית בשם `reduce` המקבלת אוסף ממוין `sGroup` (מהסוג `SortedGroup` שהוגדר לעיל) וכן איבר בודד `x` מטיפוס זהה לטיפוס איברי האוסף `sGroup`. המתודה מייצרת אוסף חדש מסוג `SortedGroup` המכיל את כל האיברים מהאוסף `sGroup` אשר גדולים ממש מהאיבר `x`. המתודה תחזיר את האוסף החדש. את הפעולה יש לבצע מבלי להרוס את האוסף המקורי `sGroup`.

ג. כתבו תכנית ראשית המייצרת אוסף מסוג `Person`. המחלקה `Person` תכיל שם, ת.ז. וגובה בסנטימטרים. ההשוואה בין אנשים תתבצע על פי הגובה. הוסיפו למחלקה גם מתודה `toString` המחזירה את פרטי האדם. בדקו את הפעולות השונות באמצעות הוספה והסרה של כמה איברים (מסוג `Person`) ולו את הפעולות באמצעות הדפסה של האוסף שמתקבל. הפעילו את המתודה `reduce` כך שתחזיר אוסף חדש שיכלול רק את האנשים שגובהם גדול מ-175 ס"מ. הדפיסו את תוכן האוסף לפני ואחרי פעולת ה-`reduce`.

שאלה 2 (50 נקודות)

ברצוננו לכתוב אפליקציה המאפשרת למשתמש לכתוב תזכורות ולשייך אותם לתאריכים שונים. המשתמש יוכל לשלוף ולעדכן תזכורות שנכתבו בעבר והוא יוכל להוסיף תזכורות חדשות. לכל תאריך אפשר לשייך תזכורת אחת המורכבת מטקסט חופשי.

ד. עליכם להשתמש במבנה נתונים מסוג טבלת hash כאשר מפתח הטבלה הוא התאריך והתוכן הוא מחרוזות המייצגות את התזכורת של אותו היום. הגדירו מחלקה משלכם שתייצג תאריך המורכב מיום חודש ושנה. שימו לב שעליכם להגדיר מחדש את המתודה equals ו-hashcode כדי שהתאריך יפעל בצורה תקינה כמפתח בטבלת ה-hash.

- ממשק המשתמש יכלול:

- ה. שלושה רכיבים מסוג JComboBox המשמשים להזנת תאריך (יום, חודש ושנה),
- איזור טקסט המשמש לכתובה והצגה של התזכורת המשויכת לתאריך הנבחר,
- כפתור לשמירת הטקסט (שמופיע באזור הטקסט) תחת התאריך שנבחר. טקסט זה יחליף את התזכורת הקיימת (אם יש כזו).
- כפתור לשליפת התזכורת שמשויכת לתאריך שנבחר והצגתה באזור הטקסט. אם אין עדיין תזכורת המשויכת לתאריך זה, תוצג מחרוזת ריקה. המשתמש יוכל לערוך את הטקסט שנשלף ולשמור אותו ובכך לעדכן תזכורת קיימת.
- התכנית תשמור את התזכורות בקובץ (המשתמש יספק שם לקובץ).
- המשתמש יוכל לאתחל את המידע מתוך קובץ קיים או להתחיל אוסף חדש של תזכורות.

מטלת מנחה (ממ"ן) 15

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 23 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 24.5.2019

סמסטר: 2019ב

קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
 - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

שאלה 1 (50 נקודות)

כתבו יישום המגדיר מערך של n תהליכים המתקשרים זה עם זה כפי שיתואר בהמשך. כל תהליך מייצג מספר כלשהו בתחום $1..100$, המספר מאותחל באופן רנדומלי. התהליכים יפעלו באופן הבא:

כל תהליך מברר מה המספר שנמצא אצל שכנו משמאל ואצל שכנו מימין (כלומר, השכן בתא $i-1$ והשכן בתא $i+1$) ופועל בהתאם לכללים הבאים:

- אם מספרו קטן ממש מהמספרים של שני שכניו, הוא יקדם את מספרו באחד.
- אם מספרו גדול ממש מהמספרים של שני שכניו, הוא יפחית את המספר שלו באחד.
- אם מספרו נמצא בין המספרים של שכניו או שווה למספר של אחד (או שניים) משכניו, הוא ישאיר את מספרו ללא שינוי.

המערך יפעל כמערך מעגלי כלומר התהליך הראשון והאחרון במערך יחשבו כשכנים. כל תהליך צריך להקפיד לא לשנות את מספרו לפני שהשכנים הספיקו לבדוק אותו. המערכת תבצע m סבבים של בדיקות.

מספר התהליכים n ומספר הסבבים m יתקבלו כקלט מהמשתמש. את פלט התכנית יש להציג באזור טקסט באופן הבא:

השורה הראשונה תייצג את המצב ההתחלתי של המערך.

השורה השנייה תייצג את המספרים בכל המערך לאחר שכל תהליך ביצע בדיקה אחת.

השורה השלישית תייצג את המספרים בכל המערך לאחר שכל תהליך ביצע בדיקה נוספת, וכן הלאה.

שאלה 2 (50 נקודות)

נתונה המחלקה SharedData המספקת שירותי חישוב ועדכון של נתונים המשותפים לתהליכים רבים:

```
public class SharedData{
    private int x = 0;
    private int y = 0;
    public SharedData (int x, int y){
        this.x = x;
        this.y = y;
    }
    public SharedData get(){
        return (new SharedData(x,y));
    }
    public void move(int dx, int dy){
        x = x + dx;
        y = y + dy;
    }
}
```

א. הגדירו שני תהליכים המקבלים reference לאובייקט מסוג SharedData (אותו אובייקט).

- התהליך הראשון מייצר 10 זוגות של מספרים אקראיים ומזיז באמצעותם בזה אחר זה את האובייקט מסוג SharedData (באמצעות המתודה move). עליכם לבצע השהייה בין הזזה להזזה ולהציג בפלט הסטנדרטי את ערכי המספרים.

- התהליך השני פונה 10 פעמים לאובייקט בכדי לקבל את ערכי האובייקט (באמצעות המתודה get). עליכם לבצע השהייה בין פנייה לפנייה ולהציג בפלט הסטנדרטי את התוצאות שהתקבלו.

- הפעילו את התהליכים במקביל ובדקו האם התוצאות שהתקבלו ע"י התהליך השני משקפות את כל ההזזות שהתבצעו ע"י התהליך הראשון.

ב. הוסיפו למחלקה SharedData מנגנון המסנכרן בין שני התהליכים כך שיפעלו לסירוגין. הפעילו את התהליכים במקביל ובדקו את התוצאות.

ג. שנו את SharedData כך שתאפשר "ריבוי קוראים וכותב יחיד" באופן הבא:

- מספר תהליכים יוכלו לבצע חישוב (get) בו זמנית, ובזמן זה לא יתבצע אף עדכון (move),

- יש לוודא שתהליך המבצע עדכון (move) יהיה היחיד שניגש באותו זמן לנתונים

המשותפים, כלומר חישובים ועדכונים נוספים לא יתבצעו בזמן שהנתונים מתעדכנים.

עליכם לבצע את השינוי ב- SharedData ולא בתהליכים. הפעילו את התכנית עם 4 תהליכים מכל סוג ועקבו אחר פעילותם באמצעות הדפסות מתאימות.

מטלת מנחה (ממ"ן) 16

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 28 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 7.6.2019

סמסטר: 2019ב

קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
 - שליחת מטלות באמצעות הדואר או הגשה ישירה למנחה במפגשי ההנחיה
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

שאלה 1 (60 נקודות)

כתבו תכנית שרת לקוח המממשת חדר שיחות המאפשר למספר כלשהו של משתמשים להשתתף בשיחה. התכנית כוללת את המאפיינים הבאים:

- כאשר משתמש מצטרף לשיחה, הוא מוסר את שמו ומקבל את רשימת המשתמשים שכבר משתתפים בשיחה.
 - כל המשתמשים מקבלים הודעה כאשר משתמש חדש מצטרף.
 - כל המשתמשים מקבלים הודעה כאשר משתמש עוזב.
 - כל הודעה הנשלחת ממשתמש כלשהו מופצת לכל המשתתפים.
- עליכם לתכנן את ממשק המשתמש כרצונכם, עליו לכלול שדה להקלדת טקסט, אזור להצגת ההודעות המתקבלות, רשימת המשתתפים, כפתורים להצטרפות ולהתנתקות ועוד. המערכת תמומש באמצעות תקשורת TCP. עליכם לאפשר למשתמש להזין את שם המחשב עליו רץ השרת. ניתן לצורך בדיקות להריץ את השרת ומספר לקוחות על אותו המחשב.

שאלה 2 (40 נקודות)

כתבו שרת חדשות המספק מבזקי חדשות ללקוחות הפונים לשרת.

תכנית השרת

- תכנית השרת תאזין בפורט 7777 ותקבל פניות של לקוחות המעוניינים להרשם לקבלת מבזקי חדשות ופניות של לקוחות המעוניינים להסיר עצמם מהרשימה.
- במקביל, התכנית תקבל מבזקי חדשות בקלט (מאזור טקסט) ותפיץ את המידע לכל הלקוחות הרשומים אצלה.
- תכנית השרת לא תשמור אצלה את המבזקים הישנים, כל מבזק יופץ ללקוחות שבאותו רגע רשומים.

תכנית הלקוח

- לקוח יוכל לפנות לשרת כדי להרשם אצלו לקבלת מבזקים או כדי להסיר עצמו מהרשימה.
 - המבזקים שיגיעו ללקוח יוצגו באזור טקסט יחד עם התאריך והשעה בו הם הגיעו.
 - המשתמש יוכל לנקות את אזור הטקסט באמצעות כפתור clear.
- המערכת תמומש באמצעות תקשורת UDP.
- עליכם לאפשר למשתמש לספק לתכנית הלקוח מידע הכולל את שם המחשב עליו רץ השרת. ניתן להריץ את השרת והלקוח על אותו המחשב.