

האוניברסיטה הפתוחה

20554

תכנות מתקדם

בשפת Java

חוברת הקורס – סתיו 2013

כתבה : תמר בניה

אוקטובר 2012 – סמסטר סתיו- תשע"ג

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
ו	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 9:00 - 11:00 בטלפון 09-7781265 או בדואר אלקטרוני tamar@openu.ac.il. פגישות יש לתאם מראש. לצורך בירורים בנושאים אדמיניסטרטיביים יש לפנות בכתב או טלפונית למחלקת האוניברסיטה הפתוחה.

בברכת לימוד מהנה,

תמר בניה

מרכזת ההוראה בקורס

1. לוח זמנים ופעילויות (מס' 20554 /א2013)

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
1	19.10.2012-14.10.2012	פרק 1-3	מפגש 1	
2	26.10.2012-21.10.2012	פרק 4-6		
3	2.11.2012-28.10.2012	פרק 7-8	מפגש 2	ממ"ן 11 2.11.2012
4	9.11.2012-4.11.2012	פרק 9-10		
5	16.11.2012-11.11.2012	פרק 10-11	מפגש 3	ממ"ן 12 16.11.2012
6	23.11.2012-18.11.2012	פרק 11		
7	30.11.2012-25.11.2012	פרק 14	מפגש 4	
8	7.12.2012-2.12.2012	פרק 15-16		ממ"ן 13 7.12.2012

* 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)
9	14.12.2012-9.12.2012 (א-ו חנוכה)	פרק 17 , 20	מפגש 5	
10	21.12.2012-16.12.2012	פרק 21 - 22		ממ"ן 14 21.12.2012
11	28.12.2012-23.12.2012	פרק 23 - 24	מפגש 6	
12	4.1.2013-30.12.2012	פרק 25-26		
13	11.1.2013-6.1.2013	פרק 26	מפגש 7	ממ"ן 15 11.1.2013
14	18.1.2013-13.1.2013	פרק 27		
15	25.1.2013-20.1.2013	פרק 27	מפגש 8	ממ"ן 16 25.1.2013

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. בראש כל מטלה מצוין אם היא מטלת חובה. רצוי שתגישו גם מטלות שאינן מטלות חובה.

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. מומלץ להגיש את המטלות דרך מערכת המטלות האלקטרונית באופן הבא:

מטלה תהייה מורכבת מקובץ אחד המכוון 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 ולישומונים (applets) קבצי html (עם סיומת html).

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) של יישום יכול את השורות :

```
java ClassFileName
```

```
pause
```

קובץ ההרצה (run.bat) של אפלט יכול את השורות :

```
appletviewer HtmlFileName.html
```

אין צורך לכלול בקבצי ההרצה את פקודת ה**הידור** `javac SourceFileName`

מטלה המוגשת דרך הדואר או ישירות למנחה תכלול תקליטור הכולל את הקבצים שתוארו לעיל וכן הדפסה של קבצי המקור. יש לוודא שהתכניות רצות ישירות מהתקליטור. 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 26 כולל	4
16	עד פרק 27 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 (כשהמוצע 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.

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: 2.11.2012

סמסטר: 2013א

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)

ברצוננו לכתוב יישום 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.

כיוון הרובוט – הרובוט יכול לפנות לאחד מארבעה כיוונים – למעלה, למטה, ימינה ושמאלה.

הגדירו טיפוס בן מנייה (Enumeration) המכיל ארבע ליטרלים (LEFT, RIGHT, DOWN, UP, ו-LEFT) המייצגים כיוונים אלו.

כתבו במחלקה בנאי המקבל שני פרמטרים – זיהוי הרובוט וכיוונו – ומאתחל איתם את הרובוט.

הוסיפו למחלקה את המתודות הבאות:

public void turnLeft() – מתודה המסובבת את הרובוט ב-90 מעלות שמאלה.

public void turnRight() – מתודה המסובבת את הרובוט ב-90 מעלות ימינה.

public String toString() – מתודה המחזירה מחרוזת המייצגת את זיהוי הרובוט והכיוון שלו.

אפשר לציין את הכיוון באמצעות התווים הבאים: ^, v, <, >.

הוסיפו מתודות גישה לקבלת נתוני הרובוט.

ב. כתבו מחלקה בשם **RobotsWorld** המיצגת את עולם הרובוטים. העולם ייוצג כמערך דו מימדי

ריבועי של רובוטים (מסוג Robot שהוגדר לעיל).

כתבו במחלקה בנאי המקבל כפרמטר את מימדי העולם, ומאתחל אותו כעולם ריק מרובוטים.

הוסיפו למחלקה את המתודות הבאות:

public boolean addRobot(Robot r, Point p) – המתודה תקבל כפרמטר אובייקט מסוג Robot

ואובייקט מסוג Point המייצג את קואורדינטות ה-x וה-y של המיקום המבוקש. אם המיקום

נמצא בתוך גבולות העולם ובמיקום זה לא קיים רובוט אחר, המתודה תוסיף את הרובוט

לעולם במיקום המבוקש ותחזיר true, אחרת יוחזר false.

`public boolean moveRobot(Robot r)` – מתודה תקבל כפרמטר אובייקט מסוג רובוט ותזיז אותו צעד אחד קדימה בכיוון אליו הוא מסתכל. אם הרובוט לא נמצא בעולם או שהוא אינו יכול לזוז כנדרש, התזוזה לא תבוצע והמתודה תחזיר `false` אחרת יוחזר `true`.

`public boolean removeRobot(Robot r)` – המתודה תקבל כפרמטר אובייקט מסוג רובוט ותסיר אותו מהעולם ותחזיר `true`. אם הרובוט לא נמצא בעולם, המתודה תחזיר `false`.

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

`public Point getPosition(Robot r)` – המתודה תקבל כפרמטר אובייקט מסוג רובוט ותחזיר אובייקט מסוג `Point` המציין את המיקום שבו הרובוט נמצא לא נמצא בעולם, המתודה תחזיר `null`.

`public String toString()` – מתודה המחזירה מחרוזת המייצגת את עולם הרובוטים. למשל העולם הבא מייצג עולם בגודל 5×5 עם רובוטים שמספרם 1 עד 4 במקומות (1,1), (2,2), (3,1) ו- (3,4):

```

— — — — —
— 1 ^ — — —
— — 2 > — —
— 3 < — — 4v
— — — — —

```

- ניתן לראות בתרשים את מיקום וכיוון הרובוטים. תא ריק מצוין באמצעות (מקף תחתון). מעבר שורה מתבצע באמצעות סדרת התווים "`\n`".
- ג. כתבו תכנית ראשית הבודקת את עולם הרובוטים באופן הבא:
- צרו עולם ריבועי בגודל המתקבל מתיבת דו-שיח (ניתן להניח פלט תקין).
 - צרו 3 רובוטים עם מספרי זיהוי עוקבים וכיוונים רנדומליים. מקמו את הרובוטים במיקומים רנדומליים בתוך תחומי העולם והציגו את העולם.
 - הזיזו כל רובוט 6 צעדים באמצעות המתודה `moveRobot` באופן הבא:
 - אם הרובוט לא הצליח לזוז, סובבו אותו ימינה ונסו שוב להזיזו. חזרו על פעולה זו לפי הצורך עד שהרובוט יוכל לזוז או עד שניסה את כל הכיוונים.
 - לוו את התזוזות בהדפסות כדי שתוכלו לעקוב אחר פעילות הרובוטים

שאלה 2 (50 נקודות)

כתבו יישום (application) המממש משחק ניחוש הפועל באופן הבא: **המשתמש** בוחר מספר בין ארבע ספרות **בלי חזרות** (המספר הנבחר) **והמחשב** מנסה לנחש את המספר הנבחר. עבור כל ספרה שהמחשב מנחש, המשתמש מספק תשובה המורכבת ממחרוזת בעלת 4 תווים המתייחסת לארבעת הספרות באופן הבא:

- אם הספרה שהמחשב ניחש מופיעה במספר הנבחר במקום הנכון, המשתמש מספק את האות 'B' (Bool),
- אם הספרה שהמחשב ניחש מופיעה במספר במקום אחר המשתמש מספק את האות 'H' (Hit),
- ואם הספרה אינה מופיעה כלל במספר הנבחר, המשתמש מספק את האות 'X',
דוגמה: בהינתן שהמספר הנבחר הוא 1307, והניחוש 7401 התשובה של המשתמש תהיה המחרוזת: "HXBH".
- הסבר: האות H הראשונה מציינת שהספרה 7 נמצאת במספר הנבחר אך מופיעה במיקום אחר, האות X מציינת שהספרה 4 לא נמצאת במספר הנבחר, האות B מציינת שהספרה 0 נמצאת במספר הנבחר בדיוק במיקום זה, והאות H האחרונה מציינת שהספרה 1 נמצאת במספר הנבחר אך מופיעה במיקום אחר.
- המחשב ממשיך לנחש עד אשר הוא מצליח לגלות את המספר. לבסוף יש להציג את מספר הניחושים להן נזקק המחשב עד שהצליח לנחש את המספר.
- הקלט והפלט יתבצעו באמצעות תיבת דו-שיח.
- לאחר סיום המשחק, יש לשאול את המשתמש (באמצעות תיבת דו-שיח) אם הוא מעוניין לשחק משחק חדש.
- הדרכה:**
- כתבו מחלקה המממשת את לוגיקת המשחק, המחלקה תכיל מתודה המחזירה את הניחוש הבא של המחשב ומתודה המקבלת מחרוזת המייצגת את התשובה של המשתמש.
- התכנית הראשית תנהל את המשחק, כלומר תציג את הניחושים של המחשב ותקלוט את התשובות של המשתמש. התכנית תשתמש במחלקת המשחק ותציג הודעות למשתמש.
- אם הקלט אינו תקין, עליכם לבקש מהמשתמש שיזין את התשובה מחדש.
- עליכם לחפש במחלקה String מתודות המתאימות לפירוק מחרוזת הקלט.

מטלת מנחה (ממ"ן) 12

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 11 (כולל)

משקל המטלה: 4

מספר השאלות: 2

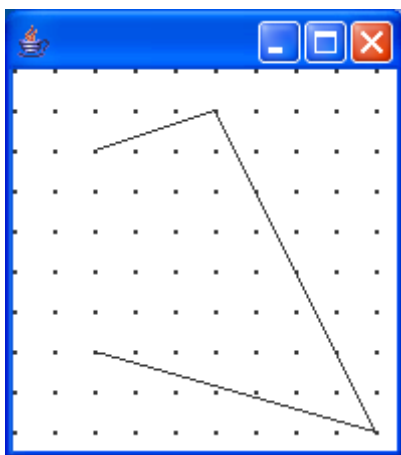
מועד אחרון להגשה: 16.11.2012

סמסטר: 2013א

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 חלונאי המשמש כמשטח ציור המורכב ממטריצה של 100 נקודות (10 נקודות בכל שורה). התכנית תאפשר למשתמש להציג קו המחבר בין נקודות לפי בחירתו. תחילה התכנית מציגה לוח ריק שכולל רק את הנקודות. לאחר מכן המשתמש מזין באמצעות תיבות דו-שיח מספרים בין 0 ל-99 המייצגים נקודות על משטח הציור. הנקודות ממוספרות ברצף משמאל לימין כאשר המספר 0 מייצג את הנקודה בפינה השמאלית העליונה, ו-99 מייצג את הנקודה בפינה הימנית תחתונה של משטח הציור. עבור כל מספר שהמשתמש מזין, התכנית מציירת קו המחבר בינו לבין הנקודה הקודמת. הציור מסתיים כאשר המשתמש מזין מחרוזת ריקה. לדוגמה, בשרטוט המוצג לעיל, המשתמש הזין את סדרת המספרים: 22, 15, 99 ו-72.

הדרכה:

התכנית תהייה מורכבת מהמחלקות הבאות:

- מחלקה מסוג JPanel המשמשת להצגה הגרפית של משטח הציור. מחלקה זו תכיל את

האלמנטים הבאים:

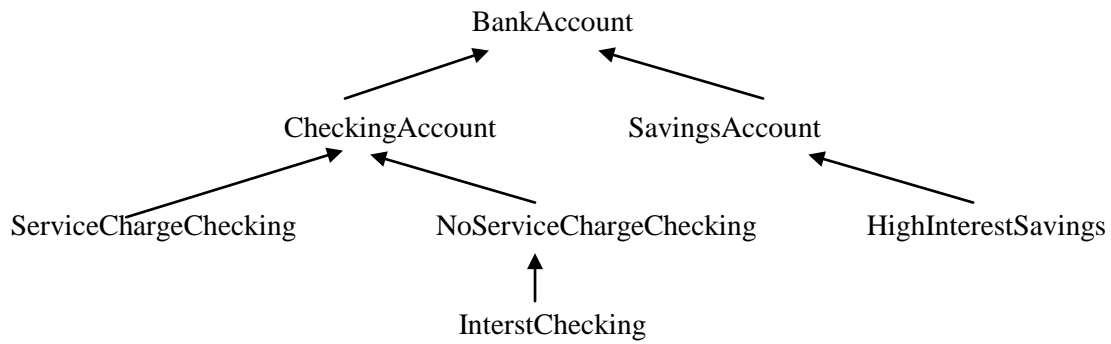
- אובייקט מסוג ArrayList שיכיל את סדרת המספרים שהמשתמש מזין.
- מתודת הציור paintComponent שתציג את השרטוט לפי המידע שנמצא ב- ArrayList. שימו לב שעליכם להציג את הציור בשלמותו ולא רק את הקו האחרון שנוסף. אפשר לקבוע רווח של 20 פיקסלים בין נקודה לנקודה.

○ מתודה להוספת קו לציור. המתודה תעדכן את ה-ArrayList ותדאג לעדכון הציור. עדכון הציור מתבצע ע"י קריאה למתודה repaint() של מחלקת ה-JPanel. מתודה זו מוגדרת במחלקה JPanel (אין צורך לכתוב אותה) והיא גורמת לתהליך האחראי על הממשק הגרפי לקרוא למתודה paintComponent (שבה כתוב קוד הציור).

- מחלקה ראשית שתיצור את החלון עם הפנל הגרפי ותכיל מתודה לקליטת ערכים מהמשתמש. עליכם לבדוק שהמשתמש מזין מחרוזות המייצגות מספרים חוקיים ושהמספרים הינם בתחום 0-99.

שאלה 2 (50 נקודות)

נתונה היררכיה של מחלקות המייצגות חשבונות בנק מסוגים שונים:



א. הגדירו את המחלקות הבאות:

המחלקה המופשטת BankAccount מייצגת חשבון בנק. המחלקה כוללת את התכונות הבאות המשותפות לכל סוגי החשבונות: מס' חשבון (מחרוזת), שם בעל החשבון (מחרוזת), ת.ז. (מחרוזת) ויתרה (מספר ממשי).

המחלקה תספק בנאי לאתחול התכונות ואת המתודות הבאות:

- מתודות גישה לקריאה ועדכון על פי הצורך.
- מתודת הפקדה המקבלת סכום ומוסיפה אותו ליתרה.
- מתודת משיכה המקבלת סכום ומפחיתה אותו מהיתרה. במקרה שבחשבון אין מספיק כסף, המשיכה לא תתבצע והמתודה תעורר מצב חריג מסוג IllegalBalance. עליכם להגדיר מצב חריג זה כתת מחלקה של Exception.
- מתודה מופשטת לניהול חודשי.
- מתודה toString להחזרת פרטי החשבון.

המחלקה המופשטת CheckingAccount מייצגת חשבון עו"ש המאפשר כתיבת המחאות. מחלקה זו כוללת מתודה נוספת בשם writeCheck המקבלת סכום המחאה כפרמטר. במקרה שלא ניתן לרשום המחאה בסכום המבוקש, מתודה תעורר מצב חריג מסוג IllegalBalance (כפי שהוגדר לעיל).

המחלקות ServiceChargeChecking ו-NoServiceChargeChecking הן מחלקות קונקרטיות המייצגות סוגים שונים של חשבונות עו"ש.

המחלקה ServiceChargeChecking מייצגת חשבון עו"ש המחויב בעמלה חודשית. המחלקה כוללת קבוע המייצג את סכום העמלה החודשית שבו החשבון יחויב. סכום זה נקבע כברירת מחדל. בנוסף יש להגדיר תכונה עבור סכום העמלה החודשית כדי לאפשר חשבונות עם סכומים שונים. המחלקה תכלול בנאי נוסף המאפשר לאתחל את סכום העמלה וכן מתודות גישה לקריאה ועדכון סכום העמלה. מתודת הניהול החודשי תבצע את חיוב העמלה.

המחלקה NoServiceChargeChecking מייצגת חשבון עו"ש ללא עמלה חודשית עם יתרת מינימום שמתחתיה אין לבצע משיכות. המחלקה כוללת קבוע המייצג את יתרת המינימום המוגדרת כברירת מחדל וכן תכונה עבור יתרת המינימום כדי לאפשר חשבונות עם יתרות מינימום שונות. המחלקה תכלול מתודות גישה לקריאה ועדכון תכונה זו.

המחלקה InterestChecking מייצגת חשבון עו"ש נושא ריבית. חשבון כזה הוא חשבון ללא עמלה חודשית עם יתרת מינימום (גבוהה יותר). המחלקה כוללת קבוע (ברירת מחדל) ותכונה עבור שער הריבית, מתודות גישה לתכונה זו ומתודה לחישוב הריבית. מתודת הניהול החודשי תזכה את החשבון בסכום הריבית החודשית (המחושבת על-פי יתרת החשבון בזמן ביצוע הניהול החודשי).

המחלקה SavingsAccount מייצגת חשבון חסכון נושא ריבית. מחלקה זו כוללת קבוע (ברירת מחדל) ותכונה עבור שער הריבית, מתודות גישה לתכונה זו ומתודה לחישוב הריבית. מתודת הניהול החודשי תזכה את החשבון בסכום הריבית החודשית (המחושבת על-פי יתרת החשבון בזמן ביצוע הניהול החודשי).

המחלקה HighInterestSavings מייצגת חשבון חסכון עם יתרת מינימום הנושא ריבית גבוהה יותר מהריבית של חשבון חסכון רגיל. המחלקה כוללת קבוע (ברירת מחדל) עבור שער הריבית וכן קבוע (ברירת מחדל) ותכונה עבור יתרת המינימום, ומתודות גישה לתכונה זו.

הערות :

- במחלקות הכוללות תכונות נוספות כגון יתרת מינימום או שער ריבית, יש לכלול בנאי אחד המאפשר לאתחל את פרטי החשבון מבלי לספק ערך עבור התכונה הנוספת (במקרה זה יש להשתמש בערך הקבוע המשמש כברירת מחדל), ובנאי נוסף המאפשר לספק ערך לתכונה הנוספת.
- עליכם לממש מחדש את המתודה toString בכל תת המחלקות בהן יש תכונות נוספות, המתודה תשקף את המידע הנוסף.
- ב. הגדירו תכנית ראשית הכוללת מערך של חשבונות מסוגים שונים. צרו חשבונות מסוגים שונים והציגו את פרטי החשבונות. בצעו הפקדות ומשיכות על החשבונות השונים ולבסוף בצעו ניהול חודשי על כל החשבונות. הציגו את פרטי החשבונות לאחר כל פעולה.

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: 7.12.2012

סמסטר: 2013א

קיימות שתי חלופות להגשת מטלות:

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

שאלה 1 (50 נקודות)

כתבו יישום המממש פזל המכיל את המספרים 1-15

המפוזרים בצורה אקראית בתוך מטריצה (4×4) .

כאשר תא אחד נשאר ריק, למשל:

1	2	3	4
5	6	7	8
9		11	12
13	14	15	10

בכל צעד המשתמש בוחר (ע"י לחיצה) באחד השכנים של התא הפנוי - שכן מוגדר כתא הגובל עם התא הפנוי מלמעלה, מלמטה, מימין או משמאל (אך לא באלכסון). הלחיצה גורמת להחלפה בין התא הפנוי לתא שנלחץ. למשל, בהינתן הלוח שהוצג לעיל, ניתן ללחוץ על המספר 6, 9, 11, או 14 כדי להעבירו לתא הפנוי, התא הפנוי יחליף את התא הנבחר. המטרה היא לסדר את המספרים בסדר עולה באופן הבא:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

סיום המשחק: המשחק מסתיים כאשר מתקבל לוח מסודר, המשתמש יקבל הודעה על הצלחתו.
הערה: לא כל מטריצה ניתנת לסידור, לכן כדי לקבל משחק פתיר, ניתן ליצור מטריצה התחלתית על-ידי "ערבוב" מטריצה מסודרת. כלומר, יש לבצע מספר כלשהו של צעדים חוקיים שבהם השכן המוזז נבחר בצורה רנדומלית.
- יש לספק כפתור בתחתית היישומון המאפשר להתחיל משחק חדש.

שאלה 2 (50 נקודות)

כתבו תכנית המשמשת להזמנת כרטיסים לבית קולנוע באופן הבא :

- התכנית תאפשר למשתמש לבחור סרט מתוך רשימה של 4 סרטים, את רשימת הסרטים יש לקרוא מקובץ.
- לאחר מכן המשתמש יבחר את שעת ההקרנה הרצויה. אפשר להניח 3 שעות הקרנה קבועות (כגון 16:00, 19:00 ו-22:00).
- לאחר בחירת המופע הרצוי, המשתמש יבחר את המושבים הרצויים. אפשר להניח שכל האולמות הם באותו מבנה הכולל n שורות ו-m כסאות בכל שורה. התכנית תאפשר לבחור מושבים בצורה גרפית למשל באמצעות מטריצה של כפתורים עם icons מתאימים או צבעים שיבדילו בין מושבים פנויים לתפוסים. התכנית לא תאפשר למשתמש לבחור מושבים תפוסים.
- לאחר סיום הבחירה, המשתמש יתבקש להזין את פרטיו (שם מספר טלפון ומייל).
- לבסוף המשתמש יקבל את סיכום ההזמנה הכולל את שם הסרט, וזמן ההקרנה, את מספרי הכרטיסים שהוזמנו והמחיר לתשלום, (אפשר להניח מחיר קבוע לכרטיס).
- לאחר שהמשתמש מאשר את ההזמנה, התכנית תאפשר לו לבצע הזמנה נוספת. הממשק הגרפי של התכנית צריך לכלול רכיבים שונים לפי שיקול דעתכם.

אפשר לקרוא מתוך קובץ טקסט באמצעות המחלקות `java.util.Scanner` ו-`java.io.File` באופן הבא :

פתיחת הקובץ `data.txt` לקריאה :

```
Scanner input = new Scanner(new File("data.txt"));
```

קריאת מחרוזות :

```
while (input.hasNext()){  
    String st = input.next();  
}
```

סגירת הקובץ :

```
input.close();
```

שימו לב, כי פתיחת הקובץ לקריאה עלולה לעורר מצב חריג הדורש טיפול.

הדרכה :

הגדירו מחלקה המייצגת מופע, המחלקה תכלול את שם הסרט, השעה ומטריצה המייצגת את המושבים.

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: 21.12.2012

סמסטר: 2013א

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)

ברצוננו לכתוב מחלקה ג'נרית לרשימה מקושרת בעלות איברים מטיפוס כלשהו. לצורך כך הגדירו את המחלקות הבאות:

א. מחלקה ג'נרית להגדרת תא ברשימה, המחלקה כוללת את האלמנטים הבאים:

- שדה מטיפוס ג'נרי כלשהו המייצג את תוכן התא.
- שדה המשמש כמצביע לתא הבא ברשימה.
- בנאי ליצירת תא, המקבל את תוכן התא ויוצר תא עם התוכן הרצוי ומצביע null.
- בנאי ליצירת תא, המקבל את תוכן התא ומצביע לתא הבא ויוצר את התא המבוקש.
- מתודות set ו-get לעדכון והחזרת תוכן התא.
- מתודות set ו-get לעדכון והחזרת המצביע לתא הבא.

ב. מחלקה ג'נרית להגדרת רשימה מקושרת. המחלקה כוללת את האלמנטים הבאים:

- שדה head המצביע לתא שנמצא בראש הרשימה.
- שדה tail המצביע לתא האחרון ברשימה.
- בנאי חסר פרמטרים ליצירת רשימה ריקה.
- מתודת גישה להחזרת התא הראשון והתא האחרון ברשימה.
- מתודת add המקבלת איבר המייצג את הנתון (תוכן התא) שיש להוסיף לרשימה. המתודה יוצרת את התא המבוקש ומוסיפה אותו לסוף הרשימה.
- מתודת remove המסירה את האיבר הראשון מהרשימה ומחזירה את תוכן התא. אם הרשימה ריקה יש לזרוק מצב חריג מסוג EmptyListException, עליכם להגדיר מצב חריג זה כמחלקה היורשת מ-Exception.
- מתודת toString המחזירה מחרוזת המייצגת את איברי הרשימה.

- ג. כתבו תכנית ראשית הבודקת את הרשימה המקושרת באופן הבא :
- צרו רשימה מקושרת של מחרוזות, קלטו מהמשתמש 6 מחרוזות והוסיפו אותן לרשימה. הציגו את הרשימה.
 - לאחר מכן צרו רשימה נוספת שתהייה מורכבת מאיברי הרשימה הראשונה כאשר סדר האיברים הוא הפוך והציגו את הרשימה השנייה. אין צורך לשמור על הרשימה המקורית.
 - צרו רשימה מקושרת של אובייקטים מסוג `java.awt.Point`, הוסיפו לה 10 אובייקטים בעלי ערכי x ו-y אקראיים בין 0 ל-100 והציגו את הרשימה.
- ד. כתבו מתודה ג'נרית המקבלת רשימה מקושרת בעלת איברים מטיפוס כלשהו המממש את הממשק **Comparable**. המתודה תחזיר את האיבר המינימלי ברשימה. עליכם לבצע זאת מבלי להרוס את תוכן הרשימה.
- הגדירו מחלקה בשם `CompPoint` היורשת מ-`Point` המממשת את הממשק `Comparable<CompPoint>`. נקודה תחשב קטנה מנקודה אחרת אם סכום ה-x וה-y שלה קטן מסכום ה-x וה-y של הנקודה האחרת. נקודה תחשב גדולה מנקודה אחרת אם סכום ה-x וה-y שלה גדול מסכום ה-x וה-y של הנקודה האחרת. הנקודות ייחשבו שוות אם סכומי ה-x וה-y שלהם שווים. לדוגמה הנקודה (3,5) תחשב כשווה לנקודה (7,1) וגדולה מהנקודה (4,1).
- צרו רשימה עבור איברים מטיפוס `CompPoint` והוסיפו לה 10 אובייקטים בעלי ערכי x ו-y אקראיים בין 0 ל-100 והציגו את הרשימה. הפעילו את המתודה עם הרשימה והציגו את האיבר המינימלי המוחזר מהמתודה.

שאלה 2 (50 נקודות)

- ברצוננו לכתוב יישום המשמש כמילון מונחים המכיל מונחים ופרושים. המערכת תציג רשימה ממוינת של המונחים והפרושים שלהם. המערכת תאפשר להוסיף מונח ופירוש, למחוק מונח ופירוש ולעדכן פירוש. לאחר כל הוספה ומחיקה יש לעדכן את הרשימה. בנוסף אפשר יהיה לחפש מונח במילון. אפשרו למשתמש לשמור ולטעון את מילון המונחים. אפשר להסתפק במילון (קובץ) אחד. הגדירו ממשק משתמש גרפי המאפשר את הפונקציונליות הנדרשת.
- הגדירו מחלקה המייצגת מילון. מחלקה זו תשתמש במבנה נתונים מתוך ה- `Java Collection Framework` לצורך שמירת המונחים. אפשר להשתמש במבנה נתונים מסוג `HashMap` כפי שמוצג בסעיף 20.11 בספר הלימוד.

מטלת מנחה (ממ"ן) 15

הקורס: 20554 – תכנות מתקדם בשפת Java

חומר הלימוד למטלה: עד פרק 26 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 11.01.2013

סמסטר: 2013א

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)

ברצוננו לכתוב יישום המבצע סימולציה להמראות ונחיתות של מטוסים בשדה תעופה.

המערכת תהייה מורכבת מהאלמנטים הבאים:

- שדה תעופה המאוחזל עם שם, ומספר מסלולים וכולל את המתודות הבאות:
 - depart: המתודה מקבלת את מספר הטיסה המבקשת להמריא ומחזירה מספר מסלול פנוי שממנו אפשר להמריא. במקרה שאין מסלול פנוי המתודה תשהה את המבקש עד אשר יתפנה מסלול.
 - land: המתודה מקבלת את מספר הטיסה המבקשת לנחות ומחזירה מספר מסלול פנוי שבו אפשר לנחות. במקרה שאין מסלול פנוי המתודה תשהה את המבקש עד אשר יתפנה מסלול.
 - freeRunway: מתודה המשמשת לשחרור מסלול שהיה בשימוש לצורך המראה או נחיתה. המתודה מקבלת את מספר הטיסה ואת מספר המסלול שהתפנה.

הערות:

- עליכם לממש את הקצאת המסלולים בשדה התעופה כך שאירועי ההמראה והנחיתה יקבלו מסלולים לפי הסדר שבו פנו לשדה.
- שימו לב שכל המסלולים משמשים הן להמראה והן לנחיתה.
- את המתודות יש ללוות עם הדפסות מתאימות כדי לעקוב אחר פעילות המערכת.
- מחלקה המייצגת טיסה: הבנאי מקבל את מספר הטיסה, את שדה התעופה שממנו יש להמריא ואת השדה שבו יש לנחות. מחלקה זו מוגדרת כ**כתהליך** (Thread) שמבצע את הפעולות הבאות:
 - פונה למתודת ההמראה (depart) בשדה המקור ומקבל מסלול פנוי,
 - מבצע סימולציה של המראה באמצעות השהייה רנדומלית של שניות (למשל בין 2 ל- 5).

- פונה לשדה ומשחרר את המסלול.
- משהה את עצמו למשך זמן הטיסה (זמן רנדומלי של כמה שניות לפי שיקול דעתכם).
- פונה למתודת הנחיתה (land) בשדה היעד ומקבל מסלול פנוי,
- מבצע סימולציה של נחיתה באמצעות השהייה רנדומלית של שניות (למשל בין 2 ל- 5).
- פונה לשדה ומשחרר את המסלול.
- מחלקה ראשית המבצעת סימולציה לפעילות ההמראות והנחיתות בשדה תעופה. התכנית תיצור שני שדות תעופה, ועשר טיסות בין שני השדות (כיווני הטיסות יוגרל בצורה רנדומלית). יש לעקוב אחר פעילות המערכת באמצעות הדפסות מתאימות.

שאלה 2 (50 נקודות)

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

```
public class SharedData{
    private double x = 0;
    private double y = 0;
    public synchronized double calc(){
        return (x+y)/2;
    }
    public synchronized void store(double x, double y){
        this.x = x;
        this.y = y;
    }
}
```

- הגדירו שני תהליכים המקבלים reference לאובייקט מסוג SharedData (אותו אובייקט).
 - התהליך הראשון מייצר 10 זוגות של מספרים אקראיים ומאחסן אותם בזה אחר זה באובייקט מסוג SharedData (באמצעות המתודה store). עליכם לבצע השהייה בין עדכון לעדכון ולהציג בפלט הסטנדרטי את ערכי המספרים.
 - התהליך השני פונה 10 פעמים לאובייקט בכדי לקבל את תוצאות החישוב (באמצעות המתודה calc). עליכם לבצע השהייה בין פנייה לפנייה ולהציג בפלט הסטנדרטי את התוצאות שהתקבלו.
 - הפעילו את התהליכים במקביל ובדקו האם חישוב הממוצע מתבצע עבור כל זוג.
 - הוסיפו למחלקה SharedData מנגנון המסנכרן בין שני התהליכים כך שיפעלו לסירוגין והממוצע יחושב עבור כל זוג מספרים. הפעילו את התהליכים במקביל ובדקו את התוצאות.
 - שנו את SharedData כך שתאפשר "ריבוי קוראים וכותב יחיד" באופן הבא:
 - מספר תהליכים יוכלו לבצע חישוב (calc) בו זמנית, ובזמן זה לא יתבצע אף עדכון (store),
 - יש לוודא שתהליך המבצע עדכון (store) יהיה היחיד שניגש באותו זמן לנתונים המשותפים, כלומר חישובים ועדכונים נוספים לא יתבצעו בזמן שהנתונים מתעדכנים.
- את השינוי יש לבצע 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 27 (כולל)

משקל המטלה: 4

מספר השאלות: 2

מועד אחרון להגשה: 25.1.2013

סמסטר: 2013א

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 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 (50 נקודות)

כתבו מערכת שרת לקוח להמרת מטבעות באמצעות **תקשורת UDP**.

תכנית הלקוח:

התכנית כוללת ממשק משתמש גרפי המאפשר למשתמש לבחור מטבע מקור ומטבע יעד (מתוך רשימות בחירה JComboBox (הכוללת סוגי מטבעות שונים), ושדה טקסט המאפשר למשתמש להזין את סכום ההמרה.

הלקוח ישלח לשרת הודעת UDP הכוללת את פרטי ההמרה המבוקשת

הלקוח ימתין עד 5 שניות לקבלת תשובה.
חשבו כיצד תוכלו לוודא שהתשובה שהתקבלה היא תשובה לבקשה שנשלחה .

אם תשובה התקבלה, היא תוצג ללקוח אחרת תוצג הודעה על כך שיש בעיה בשרת.
השרת מאזין ב-port 4444 וכתובת ה-ip של השרת תתקבל משורת הפקודה.

תכנית השרת:

השרת כולל את הפונקציות הבאה :

- השרת ממתין לבקשות מלקוחות.
- כאשר בקשה מתקבלת השרת מחשב את התשובה ושולח אותה בחזרה ללקוח.

הערות :

- אפשר להסתפק במספר קטן (למשל 5) של סוגי מטבעות.
- אפשר לאתחל את שערי החליפין בקוד השרת. אפשר להסתפק בשערי החליפין בין ₪ לסוגי המטבעות השונים ולבצע את ההמרה דרך ₪. כלומר אם מתבקשים להמיר יורו לדולר, אפשר לבצע זאת ע"י המרה של יורו ל- ₪ והמרה נוספת של ₪ לדולר.
- אפשר להריץ את השרת והלקוח על אותו המחשב.