

האוניברסיטה הפתוחה

20441
מבוא למדעי המחשב
ושפת Java
חוברת הקורס – אביב 2022

כתבה : תמר וילנר

פברואר 2022 – סמסטר אביב – תשפ"ב

פנימי – לא להפצה.

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

תוכן העניינים

א	אל הסטודנט
ה	1. לוח זמנים ופעילויות
ז	2. תיאור המטלות
ז	2.1 מבנה המטלות
ט	2.2 חומר הלימוד הדרוש לפתרון המטלות
ט	2.3 ניקוד המטלות
יב	3. התנאים לקבלת נקודות זכות בקורס
1	מטלת אופל 01
3	ממ"ן 11
11	ממ"ן 12
19	ממ"ן 13
23	מטלת אופל 02
25	ממ"ן 14
33	ממ"ן 15
37	מטלת אופל 03

אל הסטודנט

אנו מקדמים את פניך בברכה עם הצטרפותך אל הלומדים בקורס "מבוא למדעי המחשב ושפת Java".

הקורס מבוסס על הרצאותיהם של ד"ר אמיר גורן ותמר וילנר. ההרצאות המצולמות נמצאות באתר הקורס ומלוות במצגות. חוברות השקפים שקיבלתם הן אלו המלוות את סרטי הוידאו שבאתר.

לקורס קיים אתר באינטרנט בו תמצאו את חומרי הלמידה בקורס. בנוסף, באתר הקורס תמצאו חומרים כתובים ומצולמים במהלך הקורס. גם הם חובה ללימוד הקורס. בנוסף, האתר מהווה עבורכם ערוץ תקשורת עם צוות ההוראה ועם סטודנטים אחרים בקורס.

אתר הקורס הוא למעשה "ביתו" של הקורס. באתר תמצאו הסברים, הפניות והבהרות נוספות. כמו כן, באתר קיימת קבוצת דיון המאפשרת לכם לשאול שאלות את צוות הקורס וגם להתייעץ עם הסטודנטים האחרים בקורס. בקבוצת הדיון יינתן מקום לשאלת שאלות בעניין חומר הלימוד, המטלות, סביבת העבודה ועוד.

אתר הקורס בנוי בשיטת ה"רצפים (רצפי למידה מקוונים)", הרעיון הוא שהתכנים בקורס מחולקים למנות קטנות, המאורגנות באתר הקורס בצורה סדורה על פי רצף למידה המותאם ללוח הזמנים של הסמסטר. מנות התוכן מוצגות בסוגי מדיה מגוונים למשל וידאו, תרגילים אינטראקטיביים, סימולציות ועוד, המעשירות את חוויית הלמידה של הסטודנטים ומאפשרות לתרגל באופן אקטיבי את החומר הנלמד.

בסביבת הלמידה אוֹפֵּל פותח פורמט ייחודי המותאם לפדגוגית רצפי הלמידה. בפורמט הרצפים ממשק המשתמש והעיצוב הויזואלי מדגישים לסטודנטים ולסטודנטיות את רצף הלמידה המומלץ. האתר מוצג לכל סטודנט וסטודנטית במבט אישי בהתאם לקצב ההתקדמות במהלך הסמסטר.

מומלץ להיכנס לאתר לפחות פעמיים בשבוע, לעקוב אחר ההודעות בלוח ההודעות ובקבוצת הדיון. בבעיות טכניות כגון קושי בכניסה לחלקים שונים באתר וכד' אנא פנו אל המוקד בטלפון 09-7782222 או בדואר. אנא עדכנו / בדקו את פרטיכם על-מנת שתיכללו ברשימת התפוצה של דואר אלקטרוני של הקורס.

הלימוד מלווה בפתרון מטלות. משימות הלימוד לכל שבוע והתאריך האחרון למשלוח כל אחת ממטלות הקורס, רשומים ב"לוח זמנים ופעילויות" שבהמשך.

אתר הקורס הוא חלק בלתי נפרד מהקורס, והסטודנטים מחויבים להיכנס אליו ולהתעדכן בכל מה שקורה בו.
לא יישלחו הודעות/הוראות כתובות בדואר רגיל או אלקטרוני. כל ההודעות המתפרסמות באתר, כל השינויים וכל ההסברים מחייבים את כל הסטודנטים.

גם המטלות שתכתבו יישלחו אלינו דרך אתר הקורס תחת הכפתור "מערכת שליחת מטלות".
הסבר מפורט על השימוש במערכת זו יישלח אליכם בנפרד.

פרטים על למידה מתוקשבת ואתר הקורס, תמצאו באתר שה"ם בכתובת:

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

התכניות בקורס "מבוא למדעי המחשב ושפת Java" נכתבות בשפת התכנות Java, ואנו נעבוד בסביבת העבודה BlueJ שפותחה במיוחד להוראת תכנות בשפת Java. BlueJ פותחה כחלק מפרויקט מחקר אוניברסיטאי על הוראת תכנות מונחה-עצמים למתחילים. המערכת פותחה ומתוחזקת על-ידי קבוצת מחקר משותפת של אוניברסיטת Deakin ממלבורן, אוסטרליה, עם מכון Mærsk באוניברסיטה של דרום דנמרק ועם האוניברסיטה של Kent ב-Canterbury, אנגליה. הפרויקט נתמך על-ידי חברת Sun Microsystems.

מתוך הנאמר באתר האינטרנט של BlueJ: (כתובת האתר היא www.bluej.org)

The aim of BlueJ is to provide an easy-to-use teaching environment for the Java language that facilitates the teaching of Java to first year students. Special emphasis has been placed on visualization and interaction techniques to create a highly interactive environment that encourages experimentation and exploration.

כפי שאמרנו, אנו נעבוד בקורס עם סביבה זו, ותמיכה טכנית תינתן לסביבת עבודה זו בלבד. יחד עם זאת, אתם יכולים לעבוד עם כל סביבה אחרת שתמצאו, אבל עליכם להשתמש אך ורק במרכיבי השפה שאנחנו לומדים אותם בקורס (דהיינו בהרצאות המצולמות של ד"ר אמיר גורן של תמר וילנר).

הורדת סביבת העבודה מהאתר של BlueJ היא חינם. הוראות התקנה ל-BlueJ ול-Java Platform נמצאות באתר שלהם, בנוסף, באתר הקורס, ביחידה 1, בתוך "מדריכי עזר וקישורים" תוכלו למצוא מדריך להתקנת Java וסביבת העבודה BlueJ, שימו לב שמדריך זה הוא קצת ישן (ובוודאי הגרסאות המצוינות בו), ולכן עדיף לעבוד לפי ההסברים והקישורים שיש באתר של BlueJ עצמו. www.bluej.org

בתוך "הרצאות מצולמות" שביחידה 1 תמצאו סרט על סביבת העבודה BlueJ. בסרט זה תוכלו לראות איך עובדים עם הסביבה, ואיך כותבים ומריצים בה תכניות ב-Java.

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

בשאלות הקשורות ללימודי מדעי המחשב באופן כללי תוכל לפנות ליועצים האקדמיים מתחום מדעי המחשב, על-פי הרשימה המופיעה בידיעון האקדמי.

כדי לקבל ייעוץ בכל הנוגע לקורס זה, תוכלו לפנות אל תמר וילנר, מרכזת ההוראה של הקורס, בדואר אלקטרוני בכתובת tami@openu.ac.il. **בכל מכתב חובה לכתוב שם מלא ומספר ת"ז, אחרת לא אוכל לטפל בפניה!**
בשל מגפת הקורונה, אני לא נמצאת במשרד, ולכן אני לא מקיימת שעות ייעוץ טלפוניות בסמסטר זה.

לתשומת לב הסטודנטים הלומדים בחו"ל:

למרות הריחוק הפיסי הגדול, נשתדל לשמור אתכם על קשרים הדוקים ולעמוד לרשותכם ככל האפשר.
תוכלו להצטרף לשידור החי ולצפות במפגשים המוקלטים של קבוצת הנחיה מוגברת אחת, לפי מה שמשויך לכם באתר הקורס.

פרטים נוספים על הקורס נכללים בחוברת הקורס וכן באתר הקורס.
מומלץ מאד להשתמש באתר הקורס ובכל אמצעי העזר שבו וכמובן לפנות אלינו במידת הצורך.

בחוברת זו תמצא את לוח הזמנים של הקורס, תנאים לקבלת נקודות זכות בקורס ומטלות.

אני מאחלת לך לימוד פורה ומהנה.

בברכה,
תמר וילנר
מרכזת ההוראה בקורס

1. לוח זמנים ופעילויות (20441 / 2022ב)

שבוע לימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח ממ"ן (למנחה)
1	04.03.2022-27.02.2022	יחידות 1 ו-2 – הקדמה ויסודות השפה	מפגש 1	
2	11.03.2022-06.03.2022	יחידה 2 – יסודות השפה	מפגש 2	מטלת אופל 01 12.3.2022
3	18.03.2022-13.03.2022 (ד-ו פורים)	יחידה 3 – שימוש במחלקות נתונות		
4	25.03.2022-20.03.2022	יחידה 4 – כתיבת מחלקות	מפגש 3	ממ"ן 11 26.3.2022
5	01.04.2022-27.03.2022	יחידה 4 – כתיבת מחלקות		
6	08.04.2022-03.04.2022	יחידה 5 – לולאות	מפגש 4	ממ"ן 12 9.4.2022
7	15.04.2022-10.04.2022	יחידה 6 – מערכים		
8	22.04.2022-17.04.2022 (א-ו פסח)	יחידה 6 – מערכים	מפגש 5	ממ"ן 13 23.4.2022
9	29.04.2022-24.04.2022 (ה יום הזכרון לשואה)	יחידה 7 – ירושה		

* 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	06.05.2022-01.05.2022 (ד יום הזיכרון, ה יום העצמאות)	יחידה 8 – פולימורפיזם	מפגש 6	
11	13.05.2022-08.05.2022	יחידה 9 – יעילות	מפגש 7	מטלת אופל 02 14.5.2022
12	20.05.2022-15.05.2022 (ה ל"ג בעומר)	יחידות 9 ו-10 – יעילות ורקורסיה		
13	27.05.2022-22.05.2022	יחידה 10 – רקורסיה	מפגש 8	ממ"ן 14 28.5.2022
14	03.06.2022-29.05.2022	יחידה 11 – רשימות	מפגש 9	
15	10.06.2022-05.06.2022 (א שבועות)	יחידה 12 – מחסנית, תור ועצים בינריים	מפגש 10	ממ"ן 15 11.6.2022
16	לאחר תום הסמסטר	חזרה		מטלת אופל 03 18.6.2022

מועדי בחינות הגמר יפורסמו בנפרד

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

2. תיאור המטלות

קרא היטב עמודים אלו לפני שתתחיל לענות על השאלות

בקורס זה **שמונה** מטלות שעליך לפתור ולהגיש במהלך הקורס. שלוש מטלות הן מטלות מחשב (ממ"ח), אותן עליך להגיש **במערכת שאלות"א**. הממ"ח נבדק בצורה ממוחשבת ואין לשלוח את פתרון הממ"ח למנחה. השאר הן מטלות מנחה (ממ"ן), אותן עליך להגיש באמצעות **מערכת המטלות המקוונת שנמצאת באתר הקורס**.

להלן תמצא הסבר על אופן הפתרון וכיצד לשלוח את המטלה למנחה. אם שאלה במטלה אינה ברורה לך, אל תהסס להתקשר אל מרכזת ההוראה או אל אחד המנחים (בשעות הייעוץ הטלפוני שלהם **בלבד**) לצורך קבלת הסבר.

2.1 מבנה המטלות

השאלות במטלות שבקורס הן משני סוגים: שאלות "יבשות" ושאלות "הרצה".

להלן הסבר מפורט על אופן הגשת הפתרונות לשאלות:

א. **שאלות "יבשות"** הן שאלות שאינן דורשות הרצת תכניות במחשב. הן נועדו לבדוק את הבנתך בחומר הלימוד.

אופן הגשת שאלה "יבשה":

בשאלה שבה הנך מתבקש לכתוב מה מבצעת תכנית מסוימת, יש לנסח באופן כללי **באמצעות משפט אחד או שניים** (בלבד) **מהי המשימה העיקרית אותה מבצעת התכנית לכל קלט אפשרי**. בשום אופן **אין** לפרט **איך** התכנית מתבצעת, כלומר אין להסביר את מהלך ביצועה. כמו כן, אין להריץ תכניות אלה במחשב אלא ל"הריץ" אותן "על יבשה", ידנית.
זכור! במבחן אין לצדך מחשב!

ב. **שאלת הרצה:** בשאלה זו יש לכתוב תכנית ולהריץ אותה במחשב.

הקפדה על שמות מחלקות ושיטות (ציבוריות), לפי הנדרש, היא הכרחית. כל חריגה מההגדרות (אפילו החלפה בודדת של אות גדולה בקטנה, למשל) תגרום לבדיקה האוטומטית שלנו להיכשל וכתוצאה מכך לנזק בלתי הפיך בציון.

אופן הגשת שאלת הרצה:

עליך לבדוק שהיא מבצעת את הנדרש ממנה ללא טעויות. תכנית שאינה רצה נכון לא תיבדק!

תיעוד

בכל תכנית הוסף תיעוד בתחילת התכנית המסביר את האלגוריתם בו השתמשת ואת מבנה התכנית. בגוף התכנית הוסף תיעוד המסביר מהו תפקידו של כל משתנה, מה מבצע כל קטע חשוב בתכנית וכל הסבר נוסף החשוב להבנת מהלך פעולתה של התכנית (את התיעוד יש לכתוב באנגלית בלבד). יש להקפיד על בחירת שמות משמעותיים למשתנים (באנגלית).

במשך הקורס, כאשר נגיע ליחידה הרלוונטית, נלמד איך לתעד את התכניות שלנו בעזרת תיעוד שנקרא API. מרגע זה תצטרכו להגיש את כל הממ"נים מתועדים לפי הנחיות ה-API, כפי שיילמד.

המטלות בקורס זה יוגשו בעזרת מערכת שליחת המטלות שבאתר הקורס.

שאלות "יבשות" ייכתבו בעזרת מעבד תמלילים Word.

שאלות הרצה יוגשו כקובצי Java.

הקפידו על אופן שליחת מטלה – קובץ דחוס מסוג ZIP בלבד (לא RAR!), המכיל את כל הקבצים הרלוונטיים לפתרון. במידה ומדובר בפתרון שאלה "יבשה", יש לענות עליה בקובץ מסוג DOC או RTF (לא DOCX!) – מי שלא בטוח, שיפנה למנחה מבעוד מועד. במידה ומדובר בפתרון שאלה "להרצה" יש לצרף את הקבצים מסוג JAVA בלבד (לא CLASS או כל קובץ אחר). כאמור, את כל הקבצים הרלוונטיים (מסוג JAVA ו/או RTF/DOC), יש לארוז בקובץ דחוס אחד מסוג ZIP. שוב, חריגה תגרור הורדה בציון.

הסבר מפורט לגבי שליחת הקבצים המתאימים יינתן בכל מטלה בנפרד.

הסברים מפורטים על דרך שליחת המטלות בעזרת המערכת יישלחו במכתב נפרד.

2.2 חומר הלימוד הדרוש לפתרון המטלות

בטבלה שלהלן תמצא מהו חומר הלימוד הנדרש (לפי פרקי הספר) לפתרון כל אחת מהמטלות.

מטלה	חומר הלימוד הנדרש לפתרונה
מטלת אופל 01	יחידות 1 - 2
ממ"ן 11	יחידות 1 - 2
ממ"ן 12	יחידות 3 - 4
ממ"ן 13	יחידות 5 - 6
מטלת אופל 02	יחידות 7 - 8
ממ"ן 14	יחידות 9 - 10
ממ"ן 15	יחידה 11
מטלת אופל 03	יחידה 12

שימו לב!

אין להשתמש לפתרון המטלות בידע הנרכש בפרקי לימוד מתקדמים יותר מהפרקים בהם עוסקת המטלה.

2.3 ניקוד המטלות

ניתן לצבור עד 25 נקודות. חובה להגיש מטלות במשקל מינימלי של 15 נקודות לפחות.

להלן פירוט הניקוד לכל מטלה:

ממ"ן/ממ"ח	ניקוד
01	1
11	3
12	4
13	5
02	1
14	5
15	5
03	1

כדי לעבור את הקורס, צריך (בין היתר) להגיש מטלות במשקל של 15 נקודות לפחות. סטודנט שמגיש את כל המטלות, משקל המטלות שלו הוא 25 נקודות. בהתאם למשקל המטלות, נקבע אחוז הציון של בחינת הגמר בציון הסופי.

חישוב אחוז הציון של הבחינה – אם הוגשו מטלות במשקל 15 נקודות, הבחינה שווה 85% מהציון הסופי של הקורס. אם המטלות שהוגשו הן במשקל 25, הבחינה שווה 75%. וכך גם לכל המספרים ביניהם. לדוגמא, אם הוגשו מטלות 01, 12, 13, 02, 14, 03 המשקל שלהן הוא:

$17 = 1 + 5 + 1 + 5 + 4 + 1$ (ללא קשר לציון שהתקבל בכל מטלה), ואז משקל הבחינה בציון הסופי

הוא 83%.

איך מחושב ציון הקורס ?

בכל מטלה, מכפילים את הציון שלה במספר הנקודות שהיא שוקלת. מכפילים גם את ציון הבחינה במשקל שלה (לפי משקל המטלות). מחברים את כל המכפלות האלו, ומחלקים ב- 100. זה הציון הסופי בקורס.

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 (**עז 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.

לדוגמא, אם הציונים במטלות ובבחינה הם אלו :

מטלה	משקל	ציון
01	1	70
12	4	68
13	5	80
02	1	100
14	5	90
03	1	85
בחינת הגמר	83	73

שימו לב שבמטלות 01, 12 הציון נמוך מציון הבחינה. אם נוריד את שתיהן, יישארו 12 נקודות. אם נוריד את מטלה 12 יישארו 13 נקודות. לכן אפשר להוריד רק את מטלה 01 (כלומר להניח כאילו לא הוגשה כלל).

חישוב הציון ייעשה, אפוא, כך :

$$(68 \times 4 + 80 \times 5 + 100 \times 1 + 90 \times 5 + 85 \times 1 + 73 \times 84) / 100 = 74.39 \rightarrow 74$$

כלומר, הציון הסופי הוא 74.

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 שאלתא <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.

הכנת המטלות חייבת להיעשות על-ידי כל סטודנט בנפרד.

מטלות שלא יבוצעו באופן עצמאי – ייפסלו!!!

אי אפשר לעשות בזוגות (או בחבורות גדולות יותר). מי שיגיש מטלה שאנחנו נחשוד בה כמועתקת (או ככזו שהעתיקו ממנה), יועלה לוועדת משמעת. שימו לב, אפשר להתייעץ זה עם זה במהלך הכנת המטלות, אך ההתייעצות חייבת להיות בעל-פה (ללא כתיבת חומר כלשהו). לאחר ההתייעצות כל אחד חייב לכתוב את המטלה בנפרד.

מועדי הגשת המטלות

בעמוד הראשון של כל מטלה מצוין מועד הגשתה. הממ"ן ייבדק ויוחזר לך תוך שלושה שבועות מהתאריך האחרון להגשת הממ"ן. אם הממ"ן לא יוחזר אליך במועד זה, אנא התקשר עם המנחה לברר סיבת העיכוב.

דחייה בהגשת מטלות

- אנחנו נאשר איחורים רק כאשר יש בקשה מראש לאיחור כזה. בקשה מנומקת לאיחור של עד שבוע יש להפנות למנחה הקבוצה. במקרים חריגים ביותר של דחייה של למעלה משבוע, תצטרכו לפנות למרכזת ההוראה לקבלת אישור כזה. שוב, האישור צריך להתקבל מלכתחילה ולא בדיעבד.
- מי שיאחר בהגשת המטלה ללא קבלת אישור מראש, יורדו לו 3 נקודות מהציון לכל יום איחור, וגם זה רק עד שבוע. לאחר שבוע המטלה תיבדק למשוב בלבד, ולא תחשב בשקלול הציון.
- לגבי מילואימניקים – אתם יכולים לפנות אל המנחים שלכם ולסכם איתם את מועדי ההגשה למטלות המתאימים לכם ולמנחים. לכל מטלה המוגשת באיחור צרף מכתב/אישור המנמק את סיבת האיחור.

נזכיר שוב:

לבחינת הגמר רשאי לגשת רק סטודנט שצבר 15 נקודות לפחות בהגשת המטלות.

3. התנאים לקבלת נקודות זכות בקורס

- א) צבירת משקל של 15 נקודות לפחות במטלות.
- ב) ציון של 60 נקודות לפחות בבחינת הגמר.
- ג) ציון סופי בקורס של 60 נקודות לפחות.

מטלת אופל 01

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 1-2 **נושא המטלה:** הקדמה ויסודות השפה

מספר השאלות: 20 **משקל המטלה:** 1 נקודות

סמסטר: 2022 **מועד אחרון להגשה:** 12.3.2022

(ת)

סטודנטים יקרים,
מטלת אופל, היא שאלון רב ברירה (שאלון אמריקאי, ממ"ח) שאתם עונים עליו באתר הקורס.
כל הפרטים הנדרשים נמצאים באתר הקורס.
הכניסה למטלה והגשתה אפשרית מתאריך 6.3 בשעה 10:00 עד לתאריך 12.3 בשעה 23:59.
אתם מוזמנים להתנסות.

המטלה נבדקת בצורה ממוחשבת. אין לשלוח את פתרונה למנחה!

שימו לב, המנחה לא יכול לדחות לכם את תאריך הגשת מטלת אופל. זה אוטומטי והוא לא יכול להשפיע על כך. לכן אין טעם לפנות למנחה בבקשות כאלו.

זכרו לשמור את האסמכתא שאתם מקבלים מהמחשב לאחר הגשת מטלת אופל. אם לא קיבלתם מספר אסמכתא כאילו לא הגשתם את המטלה.

הנחיות לפתרון מטלת אופל

יש לקרוא כל שאלה פעמים מספר ולהתייחס לכל מלה בה. קריאה זהירה והבנה מדויקת של משמעות כל משפט בשאלה הן תנאי ראשון להצלחתכם במטלה.
לכל שאלה יש רק תשובה נכונה אחת. קראו תחילה את כל האפשרויות הנתונות, החליטו מהי האפשרות הנכונה ביותר מבין כל האפשרויות ואז סמנו אפשרות זו.
אם נדמה לכם שיש לשאלה אחת שתי תשובות נכונות, או אף שלוש, ייתכן כי תגלו, לאחר קריאת כל התשובות, תשובה אחת האומרת "שלוש התשובות הקודמות נכונות". במקרה כזה, מובן שתסמנו תשובה זו ואותה בלבד כנכונה. אם לא מופיע משפט מסוג זה, הרי רק אחת התשובות נכונה. קיימת גם אפשרות שאין כל תשובה נכונה, ובמקרה כזה תינתן לכם אפשרות לסמן כנכונה את התשובה: "אין אף תשובה נכונה".

מטלת מנחה (ממ"ן) 11

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 1-2

נושאי המטלה: יסודות השפה

מספר השאלות: 1

משקל המטלה: 3 נקודות

סמסטר: 2022

מועד אחרון להגשה: 26.3.2022


(ת)

שימו לב:

- יש להקפיד על שמות המחלקות בדיוק כמו שנכתבו.
- יש לתעד את התכניות בתיעוד פנימי באנגלית בלבד (בתחילת התכנית התיעוד מסביר מה מבצעת התכנית באופן כללי ובמהלך התכניות התיעוד מסביר את הקוד).
- אין להוסיף שיטות מעבר לאלה הנדרשות במטלה במפורש.
- אין להשתמש בחומר מתקדם ובפרט לא בלולאות.
- אין להשתמש בפקודה `exit()`.
- יש להשתמש בקבועים היכן שאפשר.
- יש להקפיד על הזחה (אינדנטציה - עימוד) נכונה, ועל שמות משתנים בעלי משמעות (באנגלית) ולפי המוסכמות בקורס.
- יש להקפיד על פורמט הפלט בדיוק כפי שמצוין בשאלה: איות נכון, אותיות גדולות וקטנות, רווחים, וכו'.
- באתר הקורס תוכלו למצוא קובץ הנחיה לפתרון המטלות התכנותיות. כדאי מאד לעיין בו ולפעול לפיו. הקובץ נמצא באתר הקורס בתוך "יחידה 1" ב"מדריכי עזר וקישורים" והוא נקרא "הנחיות לכתיבת תכניות ומטלות בקורס".
- הגשת המטלה נעשית אך ורק בעזרת מערכת המטלות המקוונת שבאתר הקורס.
- אל תשכחו לשמור את מספר האסמכתא שתקבלו מהמערכת לאחר ההגשה.

שאלה 1 - להרצה (100%)

לוח השחמט מחולק ל-64 משבצות: 8 שורות ו-8 עמודות. כל שורה ממוספרת מ-1 עד 8 וכן כל עמודה ממוספרת מ-1 עד 8. מיקום של כלי על לוח השחמט מוגדר באמצעות מספר שורה ומספר עמודה. כך, למשל, המשבצת (1,1) היא המשבצת השמאלית העליונה בלוח, והמשבצת (8,8) היא המשבצת הימנית התחתונה בלוח.

	1	2	3	4	5	6	7	8
1								
2				X		X		
3			X				X	
4								
5			X				X	
6				X		X		
7								
8								

מסע פֶּרֶשׁ (Knight) על לוח שחמט מורכב ממעבר על שתי משבצות רצופות בכוון מאוזן ואחת בכוון מאונך, או שתי משבצות רצופות בכוון מאונך ואחת בכוון מאוזן. שימו לב כי אם הפרש נמצא על אחת המשבצות שבמסגרת הלוח או קרוב לה, הוא אינו יכול לבצע את כל שמונת המסעות.

התאים אליהם יכול לנוע הפרש המסומן על לוח השחמט שלעיל מסומנים בתו X.

סעיף א (40%):

כתבו תכנית הקוראת מהקלט את מיקומו של פרש על לוח השחמט ומדפיסה את כל המקומות על הלוח שהפרש יכול לנוע אליהן.

שימו לב:

- יש לכל היותר 8 אפשרויות בפלט.
- סדר ההדפסה אינו משנה.

לדוגמה, עבור הקלט: 5 7 יודפס הפלט הבא המתאר את המיקומים האפשריים:

Moves:

3 8

3 6

4 5

6 5

7 6

7 8

שימו לב, במקרה של קלט לא חוקי (כלומר, מיקום החורג מגבולות הלוח) התוכנית תדפיס הודעת שגיאה כזו: `input is illegal` ותסיים. הניחו שהקלט הוא אכן מספרים שלמים בלבד, אין צורך לבדוק זאת.

שם קריאה מהקלט השתמשו במחלקה `Scanner`.
כדי להשתמש בה צריך לכתוב בראשית התכנית את השורה

```
import java.util.Scanner;
```

אפשר למצוא את הממשק של המחלקה `Scanner` בתוך סעיף 2.6 Interactive Programs מהספר `Java Software Solutions` שנמצא לאחר הסרטון 2.4, שם מובאות חלק מהשיטות. הסברים על המחלקה `Scanner` והשימוש בה אפשר למצוא באתר הקורס בתוך "יחידה 2" ב"מדריכי עזר וקישורים" בקובץ "מדריך לעבודה עם המחלקה `Scanner` לקבלת קלט מהמשתמש".

התכנית שכתבתם צריכה להיות במחלקה בשם `Knight`.

כתבנו כאן חלקים מהמחלקה. עליכם להשתמש בקוד זה ולהשלים את החסר (גם את התיעוד החסר).

אל תשכחו להוסיף קבועים לפי הצורך.

שימו לב – פלט התוכנית חייב להיות זהה לפלט המופק מהתוכנית המופיעה כאן (למטה) ואז ההודעה "`input is illegal`" או המילה "`Moves:`" ואז רשימת המיקומים האפשריים לפרש. עבור רשימת המיקומים – כל מיקום אפשרי יוצג בשורה נפרדת עם רווח בין סיפורה לסיפורה.

להלן תחילת המחלקה Knight:

```
import java.util.Scanner;
public class Knight
{
    public static void main (String [] args)
    {
        // כאן צריכים להופיע קבועים מתאימים...

        Scanner scan = new Scanner (System.in);
        System.out.println ("This program reads two integers which " +
            "represent the knight's location on the chess board: ");
        System.out.println ("Please enter the number of row");
        int row = scan.nextInt();
        System.out.println ("Please enter the number of column");
        int col = scan.nextInt();

        // כאן עליכם להמשיך...

    } // end of method main
} //end of class Knight
```

סעיף ב (60%):

כתבו תכנית הבודקת "איומים" של כלים על לוח השחמט. אנחנו מתמקדים בכלים הבאים:

1. רץ – bishop, יכול ללכת רק באלכסון לכל ארבעת הכיוונים האפשריים. הוא יכול לנוע משבצת אחת או מספר משבצות (עד לקצה גבול הלוח). אם הוא נתקל בכלי אויב, הוא יכול להכותו - ולהתייצב במקומו.
 2. צריח – rook, יכול ללכת רק על אותה שורה או על אותה עמודה. הוא יכול לנוע משבצת אחת או מספר משבצות (עד לקצה גבול הלוח). אם הוא נתקל בכלי אויב, הוא יכול להכותו - ולהתייצב במקומו.
 3. פרש – knight, את מסעותיו תיארונו לעיל. אם הפרש מסיים את מסעו במיקום שבו נמצא כלי אויב, הוא יכול להכותו - ולהתייצב במקומו.
- כשכלי מסוים יכול להגיע לכלי אחר ולהכות אותו לפי הכללים שציינו לעיל, אנו אומרים שהוא "מאיים" עליו.

התכנית צריכה לקרוא מהקלט שמות של שני כלים לפי האות הראשונה שלהם: פרש ('k'), רץ ('b') וצריח ('r') ואת מיקומיהם. הניחו כי האותיות בקלט הן אכן 'k', 'b' או 'r' והמספרים הם אכן מספרים שלמים. אין צורך לבדוק זאת.

התכנית צריכה לבדוק האם הכלים שונים זה מזה (מבחינת סוג הכלי), אם מיקומיהם של שני הכלים חוקיים (כלומר הם בגבולות הלוח) וכן אם המיקומים שונים זה מזה. אם כן, התכנית צריכה לבדוק אם יש איום כלשהו בין שני הכלים, ולהדפיס הודעה מתאימה.

ההודעות יהיו בדיוק אלו:

- אם הכלים זהים, יודפס: Chessmen should be different from each other
- אם לפחות אחד המיקומים לא חוקי, יודפס: Position is not legal
- אם המיקומים זהים, יודפס: Chessmen positions should not be identical
- אם כלי אחד מאיים על אחר (נניח, פרש מאיים על צריח), יודפס: knight threatens rook
- אם אין איומים, יודפס: no threat

דוגמאות:

- עבור הקלט:

b 2 3

r 4 5

יודפס: bishop threatens rook

- עבור הקלט:

b 2 3

r 5 5

יודפס: no threat

- עבור הקלט:

k 10 3

r 7 5

יודפס: Position is not legal

שימו לב, הבדיקות יתבצעו לאחר קריאת כל הקלט. ברגע שהתגלתה שגיאה מסוימת, אין צורך לבדוק הלאה, וצריך לסיים את התכנית. לכן אין צורך להדפיס את כל השגיאות שיש אלא רק את הראשונה. סדר הבדיקות צריך להיות לפי הסדר המתואר בהודעות למעלה. כמו כן, שימו לב שבשל אופן תנועת הכלים לא יתכן מצב בו שני כלים שונים יאיימו זה על זה בו זמנית.

גם כאן, כמו בסעיף א, לשם קריאה מהקלט השתמשו במחלקה Scanner.

התכנית שכתבתם צריכה להיות במחלקה בשם Chess.

כתבנו כאן חלקים מהמחלקה. עליכם להשתמש בקוד זה ולהשלים את החסר (גם את התיעוד החסר).

אל תשכחו להוסיף קבועים לפי הצורך.

שימו לב – פלט התוכנית חייב להיות זהה לפלט המופק מהתוכנית המופיעה כאן (למטה) ואז ההודעה המתאימה.

להלן תחילת המחלקה Chess:

```
import java.util.Scanner;
public class Chess
{
    public static void main (String [] args)
    {
        // כאן צריכים להופיע קבועים מתאימים ...

        Scanner scan = new Scanner (System.in);
        System.out.println("Please enter the type"+
                           " of the first chessman");
        char first = scan.next().charAt(0);
        System.out.println ("Please enter the number of row");
        int row1 = scan.nextInt();
        System.out.println ("Please enter the number of column");
        int col1 = scan.nextInt();
        System.out.println("Please enter the type"+
                           " of the second chessman");
        char second = scan.next().charAt(0);
        System.out.println ("Please enter the number of row");
        int row2 = scan.nextInt();
        System.out.println ("Please enter the number of column");
        int col2 = scan.nextInt();

        // כאן עליכם להמשיך ...

    } // end of method main
} //end of class Chess
```

בממ"ן זה אתם צריכים לכתוב שתי מחלקות. מחלקה Knight עבור השאלה הראשונה ומחלקה Chess עבור השאלה השניה. בכל מחלקה תהיה שיטה אחת – main. אין להוסיף שיטות נוספות.

במטלה זו אסור להשתמש בלולאות ו/או במערכים!

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הפתרון לסעיף א כולל את הקובץ Knight.java.
3. הפתרון לסעיף ב כולל את הקובץ Chess.java.
4. פתרון שיכלול שגיאה בשמות אחת המחלקות או השיטה main לא יבדק וציונו יהיה 0.
5. ארזו את שני הקבצים בקובץ zip (ולא rar) יחיד ושלחו אותו בלבד.
6. **אל תשכחו לשמור את מספר האסמכתא שקיבלתם מהמערכת לאחר ההגשה. אם לא קיבלתם מספר אסמכתא, סימן שההגשה לא התקבלה.**
7. שימו לב, אתם יכולים לשלוח שוב ושוב את המטלה במערכת, אם אתם רוצים לתקן משהו בה. כל הגשה דורסת את ההגשה הקודמת. **אבל עשו זאת אך ורק עד לתאריך ההגשה.** אחרי התאריך, ייחשב לכם כאילו הגשתם באיחור, גם אם ההגשה הראשונה היתה בזמן!
8. כמו כן, אם המנחה הוריד כבר את המטלה שלכם מהמערכת, לא תוכלו לשלוח עותק מעודכן יותר.

ב ה צ ל ח ה

מטלת מנחה (ממ"ן) 12

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 3 - 4 נושאי המטלה: שימוש במחלקות נתונות וכתובת מחלקות

מספר השאלות: 3 משקל המטלה: 4 נקודות

סמסטר: 2022 מועד אחרון להגשה: 9.4.2022

(ת)

מטרת מטלה זו היא להקנות לכם את עיקרי התכנות מונחה-העצמים. תתבקשו לממש מחלקות שונות המייצגות זמן וטיסה. כדי לעמוד על ההבדל בין המימוש לממשק של מחלקה, תתבקשו לכתוב שני מימושים שונים למחלקה המייצגת זמן.

שאלה 1 - 35 נקודות

המחלקה Time1 מייצגת זמן -

למחלקה Time1 יש את התכונות הפרטיות (instance variables) הבאות:

- `int _hour` – שמייצגת את השעה (בין 0 ל-23);
- `int _minute` – שמייצגת את הדקה (בין 0 ל-59).

למחלקה Time1 הוגדרו שני בנאים (constructors):

- האחד - בנאי המקבל שני פרמטרים (שעה ודקה)

```
public Time1(int h, int m)
```

אם אחד הפרמטרים שהתקבל אינו בתחום הנכון, הוא צריך להיות מאותחל ל-0.

- השני - בנאי העתקה המקבל זמן אחר, ומעתיק את ערכיו.

```
public Time1 (Time1 other)
```

בנוסף הוגדרו במחלקה השיטות:

- שיטות האחזור:

- `int getHour()` המחזירה את השעות.
- `int getMinute()` המחזירה את הדקות.

- השיטות הקובעות:

- `void setHour (int num)` המשנה את ערכה של השעה להיות `num`. אם `num` הוא לא בתחום 0-23, הערך של `_hour` לא משתנה.
- `void setMinute (int num)` המשנה את ערכה של הדקה להיות `num`. אם `num` הוא לא בתחום 0-59, הערך של `_minute` לא משתנה.

- השיטה toString() שמחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המקובל - hh:mm. כך, המחרוזת "07:30" מייצגת את הזמן של שבע שעות ושלושים דקות. **שימו לב לדייק במחרוזת לפי הכתוב כאן, ללא רווחים וללא תווים נוספים.** יש להקפיד שהשיטה תחזיר מחרוזת בת 5 תווים בדיוק.

```
public String toString()
```

- השיטה minFromMidnight שמחזירה כמה דקות עברו מאז חצות הלילה. למשל, אם הזמן המאוחר באובייקט הוא 07:30, יוחזר הערך 450.

```
public int minFromMidnight()
```

- השיטה equals המקבלת כפרמטר זמן מסוים ובודקת אם הוא זהה בערכיו לזמן שמיוצג על ידי האובייקט עליו מופעלת השיטה. אם כן, השיטה תחזיר true ואם לא, יוחזר false.

```
public boolean equals (Time1 other)
```

- השיטה before המקבלת כפרמטר זמן מסוים ובודקת אם האובייקט שעליו מופעלת השיטה **קודם** בזמן לאובייקט שמתקבל כפרמטר. אם כן, השיטה תחזיר true ואם לא, יוחזר false. למשל, 07:30 הוא לפני 13:24.

```
public boolean before (Time1 other)
```

- השיטה after המקבלת כפרמטר זמן מסוים ובודקת אם האובייקט שעליו מופעלת השיטה **מאוחר** בזמן לאובייקט שמתקבל כפרמטר. אם כן, השיטה תחזיר true ואם לא, יוחזר false. השיטה הזו משתמשת אך ורק בשיטה before שהוגדרה לעיל. **אסור להשתמש בשום אופרטור אחר או שיטה אחרת.**

```
public boolean after (Time1 other)
```

- השיטה difference המקבלת כפרמטר זמן מסוים ומחזירה את ההפרש בדקות בין האובייקט שעליו מופעלת השיטה לאובייקט שמתקבל כפרמטר. **שימו לב, אתם יכולים להניח שהאובייקט שעליו מופעלת השיטה מייצג זמן מאוחר יותר מהזמן שבאובייקט שהתקבל כפרמטר. כמו כן אפשר להניח ששני האובייקטים מייצגים זמנים באותה יממה.**

```
public int difference(Time1 other)
```

- השיטה addMinutes המקבלת כפרמטר מספר שלם num המייצג מספר דקות, ומוסיפה אותו לזמן המיוצג על-ידי האובייקט עליו מופעלת השיטה (this). השיטה מחזירה אובייקט חדש מהמחלקה Time1 המייצג את הזמן החדש. האובייקט עליו מופעלת השיטה addMinutes לא משתנה. אם הפרמטר num הוא שלילי, השיטה מחסירה את מספר הדקות הזה מהאובייקט. שימו לב לדאוג לכך שאם ישנה גלישה ליום הבא (או יותר) או אם יש גלישה אחורה בזמן ליום אחד או יותר, עדיין התכונות צריכות להיות בטווחים של 0-23 ו-0-59. **אין צורך לציין באיזושהי צורה את העובדה שעבר יום.**

```
public Time1 addMinutes(int num)
```

עליכם לכתוב את המחלקה Time1 לפי ההגדרות לעיל.

הגדרות מדויקות לפי API תמצאו באתר הקורס ביחידה 4 בתת-פרק העוסק במטלה 12.

שימו לב שאינכם יכולים להגדיר תכונות נוספות על התכונות `_hour` ו-`_minute`. מותר להגדיר קבועים נוספים למחלקה. אתם יכולים להגדיר שיטות פרטיות נוספות על אלו שהוגדרו לעיל, אבל לא שיטות ציבוריות ולא תכונות נוספות.

שאלה 2 - 35 נקודות

המחלקה `Time2` מייצגת זמן, לפי מספר הדקות שעברו מאז חצות הלילה ועד לזמן שמיוצג האובייקט.

לדוגמא, אם הזמן שמיוצג על ידי האובייקט הוא: 07:30 (כלומר 7 בבוקר, 30 דקות), הוא ייוצג על-ידי הערך 450 שכן, $450 = 30 + 7 * 60$

למחלקה `Time2` יש, אם כן, תכונה פרטית (instance variable) אחת, והיא מספר הדקות הזה.

`int _minFromMid`

עליכם לכתוב את המימוש של המחלקה `Time2`.

ה-API של שתי המחלקות `Time1` ו-`Time2` זהה לחלוטין! רק הייצוג הפנימי של האובייקטים (התכונות) שונה.

<code>Time2 (int h, int m)</code>	בנאי המקבל שני פרמטרים (שעה ודקה). אם אחד הפרמטרים שהתקבל אינו בתחום הנכון, הוא צריך להיות מאותחל ל-0.
<code>Time2 (Time2 other)</code>	בנאי העתקה המקבל זמן אחר, ומעתיק את ערכיו.
<code>int getHour()</code>	שיטות מאחזרות
<code>int getMinute()</code>	
<code>void setHour (int num)</code>	שיטות קובעות בשיטות הקובעות אם הפרמטר אינו תקין יש להשאיר את התכונה ללא שינוי
<code>void setMinute (int num)</code>	
<code>int minFromMidnight ()</code>	שיטה המחזירה כמה דקות עברו מאז חצות הלילה
<code>boolean equals(Time2 other)</code>	שיטה הבודקת האם הזמנים שווים
<code>boolean before(Time2 other)</code>	השיטה בודקת האם הזמן שעליו הופעלה השיטה מוקדם מהזמן שהתקבל כפרמטר

<i>boolean after(Time2 other)</i>	השיטה בודקת האם הזמן שעליו הופעלה השיטה מאוחר לזמן שהתקבל כפרמטר (השיטה הזו יכולה להשתמש אך ורק בשיטה <i>before</i> שהוגדרה במחלקה <i>Time2</i> ולא בשום אופרטור אחר או שיטה אחרת).
<i>int difference(Time2 other)</i>	שיטה המחזירה את ההפרש בדקות בין האובייקט שעליו מופעלת השיטה לאובייקט שמתקבל כפרמטר. אתם יכולים להניח שהאובייקט שעליו מופעלת השיטה מייצג זמן מאוחר יותר מהזמן שבאובייקט שהתקבל כפרמטר. כמו כן אפשר להניח ששני האובייקטים מייצגים זמנים באותה יממה.
<i>String toString()</i>	שיטה המחזירה את תוכן האובייקט כמחרוזת תווים לפי הייצוג המקובל - hh:mm. כך, המחרוזת "07:30" מייצגת את הזמן של שבע שעות ושלושים דקות. שימו לב לדייק במחרוזת לפי הכתוב כאן, ללא רווחים וללא תווים נוספים. יש להקפיד שהשיטה תחזיר מחרוזת בעלת 5 תווים בדיוק.
<i>Time2 addMinutes(int num)</i>	שיטה המקבלת כפרמטר מספר שלם <i>num</i> המייצג מספר דקות, ומוסיפה אותו לזמן המיוצג על-ידי האובייקט עליו מופעלת השיטה (<i>this</i>). השיטה מחזירה אובייקט חדש מהמחלקה <i>Time2</i> המייצג את הזמן החדש.

שימו לב שאינכם יכולים להגדיר תכונות נוספות על התכונה `_minFromMid`.

מותר להגדיר קבועים נוספים למחלקה.

במילים אחרות, חתימות השיטות של המחלקה *Time1* זהות לחלוטין לאלו של *Time2*, לבד מהמקרים בהם מתקבל זמן כפרמטר לשיטה, ואז במקום שכתוב *Time1* צריך להיות *Time2*.

שימו לב, גם השיטה *toString* של המחלקה *Time2* צריכה להיות לפי זו של המחלקה *Time1*, כלומר להדפיס את הזמן לפי שעות ודקות ולא לפי מספר הדקות מאז חצות.

אין להשתמש בשיטות ובבנאים של המחלקה *Time1* במחלקה *Time2*. מדובר במימושים חלופיים למחלקה של הזמן.

שאלה 3 - 30 נקודות

המחלקה Flight מייצגת טיסה.

למחלקה Flight התכונות הפרטיות (instance variables) הבאות:

- String _origin – שם העיר ממנה ממריאה הטיסה.
- String _destination – שם העיר בה נוחתת הטיסה.
- Time1 _departure – זמן ההמראה של הטיסה.
- int _flightDuration – משך זמן הטיסה בדקות.
- int _noOfPassengers – מספר הנוסעים בטיסה.
- boolean _isFull – האם הטיסה מלאה?
- int _price – מחיר לכרטיס טיסה לאדם.

כמו כן קיים במחלקה קבוע שלם MAX_CAPACITY המציין את המספר המקסימלי של נוסעים על טיסה - 250.

למחלקה Flight יש שני בנאים:

- בנאי אחד שמקבל כפרמטרים: שם עיר ההמראה, שם עיר הנחיתה, שני מספרים שלמים המהווים את זמן ההמראה של הטיסה (שעות ודקות), מספר שלם המייצג את משך זמן הטיסה בדקות, מספר שלם המייצג את מספר הנוסעים בטיסה ומספר שלם המייצג מחיר כרטיס טיסה לאדם.

שימו לב להנחות הבאות:

- אם מספר הנוסעים גדול מהמספר המקסימלי של נוסעים האפשרי, הערך שיינתן לתכונה של מספר הנוסעים יהיה המספר המקסימלי ולא הפרמטר.
 - אם מספר הנוסעים קטן מ-0, ינתן הערך 0.
 - את התכונה הבוליאנית צריך לקבוע לפי מספר הנוסעים והקבוע המציין את מספר הנוסעים המקסימלי האפשרי.
 - אם הפרמטר המייצג את משך זמן הטיסה בדקות קטן מ-0, הערך שיינתן לתכונה של זמן הטיסה יהיה 0.
 - אם הפרמטר המייצג את מחיר כרטיס טיסה קטן מ-0, הערך שיינתן לתכונה זו יהיה 0.
 - אם המספרים המהווים את זמן ההמראה של הטיסה שגויים, הטיפול בכך יתבצע במחלקה Time1.
- בנאי העתקה המקבל טיסה אחרת, ומעתיק את ערכיה.

במחלקה הוגדרו פעולות get ו-set לפי השמות המקובלים. ראו פרטים מדוייקים ב-API וראו שם גם הנחיות לטיפול במקרי קצה.

כמו כן הוגדרו השיטות הבאות :

- equals שיטה המקבלת טיסה אחרת כפרמטר ומחזירה true אם הטיסה שעליה השיטה מופעלת והטיסה שהתקבלה כפרמטר זהות. הזהות תיקבע לפי שמות הערים של ההמראה והנחיתה וזמן ההמראה.
- getArrivalTime שיטה המחשבת את זמן הנחיתה של הטיסה ומחזירה זמן זה.
- addPassengers שיטה בוליאנית המקבלת מספר של נוסעים num, ומוסיפה אותם לטיסה, אם יש בה מקום. אם יש, היא מחזירה true, אם אין מקום לכולם, היא לא מוסיפה אף אחד, ומחזירה false. שימו לב שצריך לעדכן גם את התכונה הבוליאנית isFull במקרה והיא אמורה להשתנות. ניתן להניח שהשיטה מקבלת מספר חיובי.
- isCheaper שיטה המקבלת טיסה אחרת כפרמטר ומחזירה true אם כרטיס לטיסה עליה מופעלת השיטה הוא זול יותר מכרטיס לטיסה שהתקבלה כפרמטר, אחרת יוחזר false.
- totalPrice שיטה המחשבת את התשלום הכולל שהתקבל מכל נוסעי הטיסה, ומחזירה ערך זה.
- landsEarlier שיטה המקבלת טיסה אחרת כפרמטר ומחזירה true אם הטיסה עליה מופעלת השיטה נוחתת בזמן מוקדם יותר מאשר זמן הנחיתה של הטיסה שהתקבלה כפרמטר, אחרת יוחזר false.
- toString שיטה המחזירה מחרוזת ובה נתוני הטיסה הבאים (בלבד), לפי הפורמט הבא :
Flight from *_origin* to *_destination* departs at *_departure*. Flight is full.
אם הטיסה לא מלאה, יודפס Flight is not full.

הערות כלליות:

שימו לב, אסור להוסיף תכונות פרטיות.

מותר להוסיף שיטות פרטיות.

אין להשתמש במספרים בקוד. יש להוסיף קבועים (final) עבור כל מספר קבוע ולהשתמש בקבוע בקוד.

בכל השיטות במטלה שמקבלות אובייקט כפרמטר אפשר להניח שמתקבל אובייקט שאותחל ואינו שווה ל- null.

הגדרות מדויקות לבנאים ולשיטות הנדרשות לפי API תמצאו באתר הקורס.

שימו לב ששמנו טסטרים לשלוש המחלקות באתר הקורס. חובה שטסטרים אלו ירוצו ללא שגיאות קומפילציה עם המחלקות שלכם. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטרים ירוצו עם המחלקות ללא שגיאות קומפילציה. אם הטסטרים לא ירוצו ללא שגיאות קומפילציה הציון במטלה יהיה אפס.

שימו לב לא לבצע aliasing במקומות המועדים.

עליכם לתעד את כל המחלקות שתכתבו ב-API וגם בתיעוד פנימי. אפשר כמובן להשתמש בהערות ה-API שנמצאות באתר.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות המחלקות והשיטות יהיו בדיוק כפי שמוגדר בממ"ן. **אחרת יורדו לכם הרבה נקודות!**
3. חובה להריץ את הטסטרים שנמצאים באתר הקורס על המחלקות שכתבתם. שימו לב שהטסטרים לא מכסים את כל האפשרויות, ובפרט לא את מקרי הקצה. הם רק בודקים את השמות של השיטות במחלקות. מאד מומלץ להוסיף להם בדיקות. שימו לב שאם הטסטרים לא יעברו קומפילציה מול המחלקות שכתבתם, לא יקבלו נקודות בכלל. אם יש שיטה שאתם מעוניינים לדלג עליה, עלכם לרשום את חתימת השיטה ולהחזיר ערך סתמי על מנת שהטסטרים יעברו קומפילציה.
4. את התשובות לשאלות יש להגיש בשלושה קובצי Java הבאים : Time2.java, Time1.java, Flight.java
5. ארזו את כל הקבצים בקובץ zip יחיד ושלחו אותו בלבד.

ב ה צ ל ח ה

מטלת מנחה (ממ"ן) 13

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 5 – 6 נושא המטלה: לולאות ומערכים

מספר השאלות: 1 משקל המטלה: 5 נקודות

סמסטר: 2022 מועד אחרון להגשה: 23.4.2022

(ת)

במטלה זו אנו משתמשים במחלקות Time1 ו-Flight שכתבנו בממ"ן 12.

אתם יכולים להשתמש במחלקות Time1 ו-Flight שכתבתם או בקבצים Time1.class ו-Flight.class שיהיו באתר ביחידה 6 בצמוד למטלה 13. נשים את הקבצים האלו באתר רק אחרי ההגשה של מטלה 12.

שימו לב שהקבצים שיש באתר הם קובצי class ולא הקוד ב-Java. אי אפשר לפתוח אותם אלא להשתמש בהם בלבד. לצידם יש קובצי html שהם ה-API של המחלקות Time1 ו-Flight.

אנא קראו את הכתוב במדריך מדריך עזר קצר לשימוש בקובצי מחלקות (class) הקיימים בפרויקט שאתם יוצרים ב-Blue. המדריך נמצא בלשונית "מדריכי עזר" ביחידה 3 באתר הקורס. כך תדעו איך להשתמש במחלקה שכבר כתובה, וניתנת לכם כקובץ class ללא הקוד. שמנו באתר טסטר בסיסי לבדיקה ראשונית של המטלה. חובה להריץ את המטלה מול הטסטר ולבדוק שאין טעויות קומפילציה.

שאלה 1 - להרצה (100%)

המחלקה Airport מייצגת את לוח הטיסות בשדה התעופה ביממה.

הייצוג נעשה על-ידי מערך ששומר את רשימת הטיסות. התכונות במחלקה הן:

- מערך של הטיסות `Flight [] _flightsSchedule`
 - מספר הטיסות בלוח הטיסות (במערך) `int _noOfFlights`
 - שם העיר בה נמצא שדה התעופה `String _city`
- כמו כן קיים במחלקה קבוע שלם `MAX_FLIGHTS` המציין את המספר המקסימלי של טיסות ביממה – 200.

הטיסות (כלומר האובייקטים מהמחלקה Flight) נמצאים במערך ברצף, ללא "חורים" מתחילת המערך. המערך צריך להישאר כך (ללא חורים) לאחר כל פעולה.

עליכם לכתוב את המימוש ב-Java של המחלקה Airport. מימוש המחלקה כולל את הסעיפים שלהלן:

1. הגדרת הקבוע של המחלקה.
2. הגדרת התכונות של המחלקה.
3. בנאי שמקבל את שם העיר בה נמצא שדה התעופה ומאתחל את תכונות המחלקה כך שמערך הטיסות יהיה בגודל מקסימלי ומספר הטיסות הוא 0.
4. שיטה (addFlight) בוליאנית המוסיפה טיסה ללוח הטיסות. השיטה מקבלת את הטיסה כפרמטר. השיטה מחזירה ערך true אם ההוספה התבצעה כשורה, אם לא, השיטה תחזיר false. שימו לב שהמקור או היעד של הטיסה חייבים להיות העיר בה נמצא שדה התעופה. אתם יכולים להניח שהטיסה הזו לא קיימת כבר בלוח הטיסות. אין צורך לבדוק זאת.
5. שיטה (removeFlight) בוליאנית המוחקת טיסה מלוח הטיסות. השיטה מקבלת את הטיסה כפרמטר. השיטה מחזירה ערך true אם המחיקה התבצעה כשורה, אם לא, השיטה תחזיר false.
6. שיטה (firstFlightFromOrigin) המקבלת עיר כלשהי place, מחזירה את הזמן בו ממריאה הטיסה הראשונה באותו יום מהמקום place. אם אין אף טיסה באותו יום מהמקום place יוחזר null.
7. שיטה (howManyFullFlights) המחזירה מספר האומר כמה טיסות מלאות יש באותו יום.
8. שיטה (howManyFlightsBetween) המקבלת עיר place ומחזירה מספר האומר כמה טיסות יש באותו יום הממריאות משדה התעופה city ונוחתות ב-place, או ממריאות מ-place ונוחתות ב-city.
9. שיטה (mostPopularDestination) המחזירה את העיר הכי פופולרית באותו יום (כלומר העיר בה נוחתות הכי הרבה טיסות). אם אין טיסות בכלל בלוח הטיסות, יוחזר null. אם יש כמה ערים שהן פופולריות ביותר באותה מידה, תוחזר העיר הפופולרית הראשונה שנמצאה בלוח הטיסות.
10. שיטה (mostExpensiveTicket) המחזירה את הטיסה שהכרטיס שלה הוא היקר ביותר. אם אין טיסות בכלל בלוח הטיסות, יוחזר null. אם יש כמה טיסות שהן יקרות ביותר באותה מידה, תוחזר הטיסה היקרה ביותר הראשונה שנמצאה בלוח הטיסות.
11. שיטה (longestFlight) המחזירה את הטיסה הארוכה ביותר במערך הטיסות. אם אין טיסות בכלל בלוח הטיסות, יוחזר null. אם יש כמה טיסות שהן ארוכות ביותר באותה מידה, תוחזר הטיסה הארוכה ביותר הראשונה שנמצאה בלוח הטיסות.
12. שיטה (toString) המחזירה מחרוזת המתארת את כל הטיסות במערך הטיסות כסדרן לפי המערך (אין צורך למיין לפי זמנים או משהו אחר). כל טיסה תהיה בשורה נפרדת. ובתחילה תהיה כותרת. אם אין טיסות בכלל בלוח הטיסות, יוחזר null.

ראו את הדוגמא הבאה :

The flights for airport Tel-Aviv today are:

Flight from Tel-Aviv to London departs at 12:00. Flight is full.

Flight from New York to Tel-Aviv departs at 10:50. Flight is full.

Flight from Tel-Aviv to Paris departs at 11:35. Flight is not full.

בכל השיטות לעיל, אם מועבר אובייקט כפרמטר, אפשר להניח שהוא אינו null.

שימו לב לא לבצע aliasing במקומות המועדים.

תזכורת – השוואה בין אובייקטים (ומחרוזות) צריכה להיעשות בעזרת השיטה equals ולא על-ידי סימן השוויון ==.

מותר להוסיף שיטות נוספות (פרטיות בלבד), לפי ראות עיניכם, אבל אי אפשר להוסיף תכונות נוספות. כמו כן, כל התכונות חייבות להיות פרטיות!

לפניכם רשימת החתימות של הבנאי ושיטות המחלקה :

- `public Airport(String city)`
- `public boolean addFlight(Flight f)`
- `public boolean removeFlight(Flight f)`
- `public Time1 firstFlightFromOrigin (String place)`
- `public String toString()`
- `public int howManyFullFlights()`
- `public int howManyFlightsBetween (String city)`
- `public String mostPopularDestination()`
- `public Flight mostExpensiveTicket()`
- `public Flight longestFlight()`

אתם צריכים לכתוב בעצמכם API למחלקה, לבנאים ולשיטות לפי הנהוג בכתיבת API. כמו כן, עליכם לתעד בתיעוד פנימי כל מה שדורש הבהרה ואינו פשוט.

שימו לב,

באתר הקורס תמצאו גם טסטר לבדיקת האיות והפרמטרים של השמות של השיטות והמחלקה שאתם צריכים לכתוב. חובה עליכם לבדוק את המחלקה שכתבתם בטסטר זה, ולהגיש אותה רק אם הטסטר עובר קומפילציה. שימו לב שהטסטר לא מכסה את כל האפשרויות, ובפרט לא את מקרי הקצה. הוא רק בודק את השמות של השיטות במחלקות כלומר שגיאות קומפילציה. מאד מומלץ להוסיף לו בדיקות. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטר ירוצו עם המחלקה ללא שגיאות קומפילציה. אם הטסטר לא ירוץ ללא שגיאות קומפילציה הציון במטלה יהיה אפס.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו לתעד בתיעוד פנימי וב-API את כל השיטות שיש במחלקות השונות.
3. הקפידו ששמות השיטות יהיו בדיוק כפי שכתוב במטלה. וכן שההדפסות יהיו בדיוק כפי שמופיע במטלה.
4. עליכם להגיש את הקובץ Airport.java, עטפו אותו בקובץ zip ושלחו. אין לשלוח קבצים נוספים.

מטלת אופל 02

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 7 – 8

נושא המטלה: ירושה ופולימורפיזם

מספר השאלות: 20

משקל המטלה: 1 נקודות

סמסטר: 2022

מועד אחרון להגשה: 14.5.2022

(ת)

סטודנטים יקרים,
מטלת אופל, היא שאלון רב ברירה (שאלון אמריקאי, ממ"ח) שאתם עונים עליו באתר הקורס.
כל הפרטים הנדרשים נמצאים באתר הקורס.
הכניסה למטלה והגשתה אפשרית מתאריך 8.5 בשעה 10:00 עד לתאריך 14.5 בשעה 23:59.
אתם מוזמנים להתנסות.

המטלה נבדקת בצורה ממוחשבת. אין לשלוח את פתרונה למנחה!

שימו לב, המנחה לא יכול לדחות לכם את תאריך הגשת מטלת אופל. זה אוטומטי והוא לא יכול להשפיע על כך. לכן אין טעם לפנות למנחה בבקשות כאלו.

זכרו לשמור את האסמכתא שאתם מקבלים מהמחשב לאחר הגשת מטלת אופל. אם לא קיבלתם מספר אסמכתא כאילו לא הגשתם את המטלה.

הנחיות לפתרון מטלת אופל

יש לקרוא כל שאלה פעמים מספר ולהתייחס לכל מלה בה. קריאה זהירה והבנה מדויקת של משמעות כל משפט בשאלה הן תנאי ראשון להצלחתכם במטלה.
לכל שאלה יש רק תשובה נכונה אחת. קראו תחילה את כל האפשרויות הנתונות, החליטו מהי האפשרות הנכונה ביותר מבין כל האפשרויות ואז סמנו אפשרות זו.
אם נדמה לכם שיש לשאלה אחת שתי תשובות נכונות, או אף שלוש, ייתכן כי תגלו, לאחר קריאת כל התשובות, תשובה אחת האומרת "שלוש התשובות הקודמות נכונות". במקרה כזה, מובן שתסמנו תשובה זו ואותה בלבד כנכונה. אם לא מופיע משפט מסוג זה, הרי רק אחת התשובות נכונה. קיימת גם אפשרות שאין כל תשובה נכונה, ובמקרה כזה תינתן לכם אפשרות לסמן כנכונה את התשובה: "אין אף תשובה נכונה".

מטלת מנחה (ממ"ן) 14

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 9-10 נושא המטלה: יעילות ורקורסיה

מספר השאלות: 4 משקל המטלה: 5 נקודות

סמסטר: 2022 מועד אחרון להגשה: 28.5.2022

השאלות במטלה זו לקוחות מבחינות גמר שונות או דומות לשאלות של בחינות גמר. אנו ממליצים מאד, בשלב הראשון, לענות עליהן ללא הרצה במחשב (כפי שמקובל בבחינת הגמר).

את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם Ex14.java (בדיוק).
את התשובות לשאלות על הסיבוכיות כתבו כחלק מה-API.

שאלה 1- 25 נקודות

בשאלה זו נניח ש-m הוא מערך דו ממדי ריבועי (מספר השורות שלו שווה למספר העמודות).
התבוננו בשיטות הבאות:

```
public static boolean what(int [][] m)
{
    int n=m.length;
    for(int x=0; x<n; x++)
        for (int y=0; y<(n-1); y++)
            if(m[x][y] > m[x][y+1]) return false;
    for(int x=0; x<n; x++)
        for (int y=0; y<(n-1); y++)
            if(m[y][x] > m[y+1][x]) return false;
    return true;
}
```

```

public static boolean riddle(int [][] m, int val)
{
    int n = m.length;
    for(int x=0; x<n; x++)
        for (int y=0; y<n; y++)
            if(m[x][y] == val) return true;
    return false;
}

public static boolean test(int [][] m)
{
    int n=m.length;
    for(int r=0; r<(n-1); r++)
        for (int c=0; c<n; c++)
            for (int i=0; i<n; i++)
                if(m[r][c] > m[r+1][i]) return false;
    return true;
}

```

סעיף א - (6 נקודות)

עליכם לסמן את כל הטענות הנכונות:

1. בהינתן מערך m , אם השיטה `test` תחזיר `true` עבור המערך m אזי גם השיטה `what` תחזיר `true` עבור המערך m .
2. בהינתן מערך כל-שהוא m , אם השיטה `what` תחזיר `true` עבור המערך m אזי גם השיטה `test` תחזיר `true` עבור המערך m .
3. זמן הריצה של השיטה `what` הוא $O(n^2)$ אם n הוא מספר השורות במערך m .
4. זמן הריצה של השיטה `test` הוא $O(n^2)$ אם n הוא מספר השורות במערך m .
5. זמן הריצה של השיטה `riddle` הוא $O(n^2)$ אם n הוא מספר השורות במערך m .
6. זמן הריצה של השיטה `riddle` הוא $O(n)$ אם n הוא מספר התאים במערך m .

סעיף ב - (19 נקודות)

כתבו את השיטות הבאות :

1. נתון ש-m הוא מערך שכשנעביר אותו כפרמטר לשיטה what השיטה תחזיר true. השיטה findValWhat צריכה לבדוק האם הערך val נמצא במערך m בזמן ריצה $O(n)$ כאשר n הוא מספר השורות במערך m. (7 נקודות) (פתרון לא רקורסיבי)

חתימת השיטה תהיה:

```
public static boolean findValWhat (int [][] m, int val)
```

2. נתון ש-m הוא מערך שכשנעביר אותו כפרמטר לשיטה test השיטה תחזיר true. השיטה findValTest צריכה לבדוק האם הערך val נמצא במערך m בזמן ריצה $O(n)$ כאשר n הוא מספר השורות במערך m. (12 נקודות) (פתרון לא רקורסיבי)

חתימת השיטה תהיה:

```
public static boolean findValTest (int [][] m, int val)
```

שימו לב:

השיטות שתכתבו צריכות להיות יעילות ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.

ניתן להשתמש בשיטות עזר ככל הנדרש. בחישוב הסיבוכיות צריך לחשב גם את הזמן והמקום של שיטות העזר.

כתבו מה סיבוכיות זמן הריצה וסיבוכיות המקום של השיטות שכתבתם.
הסבירו תשובתכם.

אל תשכחו לתעד את מה שכתבתם!

שאלה 2- 25 נקודות

נתון מערך חד-ממדי a המלא במספרים שלמים.

כתבו שיטה סטטית, המקבלת כפרמטר מערך כזה, ומחזירה את מספר התת-מערכים בתוכו שהם מסודרים בסדר עולה ממש (כלומר ללא מספרים שווים). שימו לב שאנו מדברים על תאים סמוכים אחד לשני. אורך של תת-מערך מסודר בסדר עולה ממש הוא לפחות 2.

לדוגמא,

- אם נתון המערך a הבא :

		0	1	2	3	4
a	=	1	2	4	4	5

השיטה צריכה להחזיר את הערך 4.

אלו ארבעת התת-מערכים המסודרים בסדר עולה ממש :

{1, 2}, {1, 2, 4}, {2, 4}, {4, 5}

- אם נתון המערך a הבא :

		0	1	2
a	=	1	3	2

השיטה צריכה להחזיר את הערך 1. התת-מערך {1, 3}

- אם נתון המערך a הבא :

		0	1	2	3	4
a	=	5	4	3	2	1

השיטה צריכה להחזיר את הערך 0.

חתימת השיטה היא:

```
public static int strictlyIncreasing (int[] a)
```

שימו לב:

השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.

ניתן להשתמש בשיטות עזר ככל הנדרש. בחישוב הסיבוכיות צריך לחשב גם את הזמן והמקום של שיטות העזר.

מה סיבוכיות זמן הריצה וסיבוכיות המקום של השיטה שכתבתם?
הסבירו תשובתכם. אל תשכחו לתעד את מה שכתבתם!

שאלה 3- 25 נקודות

נגדיר **סדרה שטוחה** (flat sequence) של מספרים שלמים אם כל המספרים בה הם אותו מספר או שהסדרה מורכבת משני מספרים עוקבים (consecutive). לדוגמא,

הסדרות הבאות הן שטוחות :

- 4, 4, 4, 4, 4

- 2, 3, 2, 2, 2, 3, 3, 2

הסדרות הבאות אינן שטוחות :

- 2, 3, 4, 3, 4, 3, 3, 2 כי היא מורכבת משלושה מספרים ולא רק שניים

- 2, 4, 2, 2, 2, 4, 4, 2 כי המספרים 2 ו-4 אינם מספרים עוקבים.

סדרה ריקה, או סדרה שיש בה איבר אחד נחשבת כסדרה שטוחה.

כתבו שיטה סטטית **רקורסיבית** המקבלת מערך של מספרים שלמים arr, ומחזירה את אורכו של התת-מעריך המקסימלי המהווה סדרה שטוחה.

שימו לב, בתת-מעריך מודבר על האיברים בתאים סמוכים אחד לשני.

לדוגמא, עבור המעריך הבא

0	1	2	3	4	5
4	5	6	5	4	3

השיטה תחזיר 3. שכן, התת-מערכים שמהווים סדרות השטוחות במעריך הם :

- {4, 5} באורך 2

- {5, 6, 5} באורך 3

- {5, 4} באורך 2

- {4, 3} באורך 2

מכאן שהתת-סדרה השטוחה הארוכה ביותר במעריך היא באורך 3, וזה הערך שיוחזר.

חתימת השיטה היא:

```
public static int longestFlatSequence (int[] arr)
```

עזרה:

כדאי שתכתבו שיטת עזר שמחשבת ומחזירה את אורך הסדרה השטוחה החל מאינדקס מסוים במעריך. חתימה של השיטה יכולה להיות כזו :

```
public static int lengthFlat (int [] arr, int index)
```

כמובן שהחתימה יכולה להכיל גם פרמטרים נוספים, כמו המונה (counter) של אורך התת-סדרה עד עכשיו, שני הערכים שיכולים להיות בסדרה, ועוד.

השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות. תשובה שתכיל לולאה, אפילו במקום שולי בתשובתכם לשאלה זאת, תקבל אפס נקודות.

מותר להשתמש בשיטה Math.max

שאלה 4- 25 נקודות

נתון מערך דו-ממדי mat, שערכיו הם המספרים השלמים 1, 0 או -1. מספר השורות לא בהכרח שווה למספר העמודות. מספר העמודות בכל השורות שווה.

נגדיר **מסלול** במערך אם הוא מקיים את התנאים הבאים:

1. התא הראשון במסלול הוא mat[0][0] (שורה ראשונה ועמודה ראשונה)
2. המסלול עובר אך ורק בתאים שמכילים 1 או 0.
3. בשורה עם אינדקס זוגי, אפשר ללכת אך ורק לתא **מימין** באותה שורה או לרדת לשורה הבאה באותה עמודה.
4. בשורה עם אינדקס אי-זוגי, אפשר ללכת אך ורק לתא **משמאל** באותה שורה או לרדת לשורה הבאה באותה עמודה.

כל מעבר של מסלול בתא שבו מופיע הערך 1 מעניק נקודה אחת. מעבר של המסלול בתא שבו מופיע הערך 0 לא מעניק אף נקודה. כזכור אסור למסלול לעבור בתאים שבהם הערך הוא -1.

עליכם לכתוב שיטה סטטית רקורסיבית המקבלת מערך דו-ממדי כנ"ל, המחזירה את סכום הנקודות הגבוה ביותר אליו אפשר להגיע במסלול כלשהו במערך הדו מימדי mat.

לדוגמא, מספר הנקודות המקסימלי במערך משמאל הוא 9.

במערך מימין מסומן המסלול, ומודגשים התאים במסלול שערכם הוא 1.

	0	1	2	3	4
0	1	1	-1	1	1
1	1	0	0	-1	1
2	1	1	1	1	-1
3	-1	-1	1	1	1
4	1	1	-1	-1	1

	0	1	2	3	4
0	1	1	-1	1	1
1	1	0	0	-1	1
2	1	1	1	1	-1
3	-1	-1	1	1	1
4	1	1	-1	-1	1

אם בתא [0][0] הערך הוא -1 השיטה תחזיר -1.

חתימת השיטה היא:

```
public static int findMaximum(int[][] mat)
```

השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.

אפשר להשתמש בהעמסת-יתר (overloading).

מותר לשנות את המערך במהלך השיטה, אבל בסופה הוא צריך לחזור למצבו המקורי.

אין צורך לדאוג ליעילות השיטה, אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות! אל תשכחו לתעד את מה שכתבתם!

שימו לב:

בשאלות 3 ו-4:

- אין צורך לדאוג ליעילות השיטה שתכתבו! אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!
- אפשר להשתמש בהעמסת-יתר (overloading).
- מותר לשנות את המערך במהלך השיטה, אבל בסופה הוא צריך לחזור למצבו המקורי.

בכל השאלות:

- אסור להשתמש במשתנים גלובליים שהוגדרו מחוץ לשיטות עצמן!
- אל תשכחו לתעד את מה שכתבתם!

שימו לב ששמנו טסטר באתר הקורס. חובה שהטסטר ירוץ עם המחלקה שכתבתם ללא שגיאות קומפילציה. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטר ירוצו עם המחלקה ללא שגיאות קומפילציה. אם בהרצת הטסטר יתגלו שגיאות קומפילציה הציון במטלה יהיה אפס, ללא אפשרות ערעור.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות השיטות יהיו **בדיוק** כפי שמוגדר בממ"ן.
3. עליכם לתעד את כל השיטות שאתם כותבים בתיעוד API ובתיעוד פנימי המסביר מה עשיתם בשיטה. בתיעוד זה כתבו גם מה הסיבוכיות של השיטות (בשאלות 1 ו-2).
4. **את התשובות לכל השאלות עליכם לכתוב במחלקה אחת בשם Ex14.java (בדיוק). ארזו אותו בתוך קובץ zip. אין לשלוח קבצים נוספים.**

בהצלחה

מטלת מנחה (ממ"ן) 15

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידה 11 נושא המטלה: רשימות מקושרות

מספר השאלות: 1 משקל המטלה: 5 נקודות

סמסטר: 2022 מועד אחרון להגשה: 11.6.2022

רשימה מקושרת בנויה מקדודים. ברשימה שנממש בממ"ן זה תוכן כל קדקוד הוא מספר שלם, והוא מחזיק "מצביע" לקדקוד הבא אחריו.

להלן נתון קוד המחלקה `IntNode` המייצג קדקוד ברשימה מקושרת חד-סטרית.

```
public class IntNode
{
    private int _value;
    private IntNode _next;

    public IntNode(int v, IntNode n)
    {
        _value = v;
        _next = n;
    }

    public int getValue() {
        return _value;
    }

    public IntNode getNext() {
        return _next;
    }

    public void setValue(int v) {
        _value = v;
    }

    public void setNext(IntNode n) {
        _next = n;
    }
}
```

שאלה 1 - להרצה (100%)

נגדיר :

קבוצה (set) היא אוסף לא סדור של עצמים המכונים **האיברים** (element) של הקבוצה.

נתאר כמה ממאפייני **קבוצה** :

- בהינתן קבוצה A ועצם x כלשהו, מתקיימת בדיוק אחת משתי האפשרויות : או x הוא איבר של A (ובמילים אחרות נאמר : x שייך ל- A , x נמצא ב- A), או x אינו איבר של A .
- בכל קבוצה מתקיים שכל האיברים שונים זה מזה. כלומר, אין חזרות של איברים.
- שתי קבוצות A ו- B נחשבות שוות זו לזו, כאשר לשתיהן יש בדיוק אותם איברים. **אין חשיבות לסדר האיברים בקבוצה**. לדוגמא, הקבוצה $\{1, 2, 4\}$ זהה לקבוצה $\{2, 4, 1\}$.
- קיימות קבוצות סופיות, כגון : קבוצת הפתרונות של המשוואה $x^2 - 7x + 12 = 0$ (הקבוצה מכילה את המספרים 3 ו-4). במקרה כזה נגיד כי **מספר האיברים** בקבוצה הוא 2.
- קיימות קבוצות אינסופיות, כגון : כל המספרים הממשיים בין 0 ל-1 (כולל קצוות). במקרה כזה לא נדבר על מספר האיברים בקבוצה.
- קבוצה A תיקרא **קבוצה ריקה** אם אין בה איברים בכלל (כגון, קבוצת כל המספרים בין 10 ל-15 המתחלקים ב-9). במקרה כזה נגיד כי מספר האיברים הוא 0.
- בהינתן שתי קבוצות A ו- B , A נקראת **תת-קבוצה** של B , אם כל איבר של A הוא איבר של B . (ובמילים אחרות נאמר : A חלקית ל- B , או A מוכלת ב- B).
- בהינתן שתי קבוצות A ו- B , קבוצת **החיתוך** של שתי הקבוצות היא אוסף האיברים שנמצאים גם בקבוצה A וגם בקבוצה B . מסומן $A \cap B$.
לדוגמא, אם $A = \{6, 2, 4\}$ ו- $B = \{2, 3, 8, 6\}$ אז $A \cap B = \{2, 6\}$
- בהינתן שתי קבוצות A ו- B , קבוצת **האיחוד** של שתי הקבוצות היא אוסף האיברים שנמצאים לפחות באחת הקבוצות A או B . מסומן $A \cup B$.
לדוגמא, אם $A = \{6, 2, 4\}$ ו- $B = \{2, 3, 8, 6\}$ אז $A \cup B = \{2, 3, 4, 6, 8\}$
- בהינתן שתי קבוצות A ו- B , קבוצת **הפרש** $A \setminus B$ היא אוסף האיברים שנמצאים בקבוצה A ולא נמצאים בקבוצה B .
לדוגמא, אם $A = \{6, 2, 4\}$ ו- $B = \{2, 3, 8, 6\}$ אז $A \setminus B = \{4\}$

עליכם להגדיר מחלקה בשם Set המייצגת את קבוצת המספרים הטבעיים האי-זוגיים (טבעיים = שלמים חיוביים). כלומר {1, 3, 5, 7, ...} (עד אינסוף).

שימו לב, כל אחד מהאובייקטים שיוצרו מהמחלקה Set שתכתבו יוכל להחזיק קבוצה סופית משלו. לדוגמא, אובייקט אחד ייצג את הקבוצה {1, 25, 3}, אובייקט אחר ייצג למשל את הקבוצה {9, 5, 25, 17} ואובייקט שלישי ייצג את קבוצת כל המספרים האי-זוגיים מ-100 עד 1000.

מימוש קבוצה צריך להיות באמצעות רשימה מקושרת.

המחלקה צריכה לייצג את הקבוצה בצורה היעילה ביותר למימוש. לכן חשבו היטב איך תצרו את הרשימה שתחזיק את המספרים שבקבוצה, כך שלפחות חלק מהפעולות המוזכרות לעיל ייעשו באופן היעיל ביותר.

עליכם לכתוב את הגדרת המחלקה: תכונות (instance variables) - לפי בחירתכם.

השיטות הנדרשות במחלקה הן:

- `public boolean isEmpty ()`
המחזירה true אם הקבוצה ריקה ו- false אחרת;
- `public boolean isMember (int num)`
המחזירה true אם הערך num הוא איבר בקבוצה ו- false אחרת;
- `public boolean equals (Set other)`
המחזירה true אם הקבוצה other שווה לקבוצה שבאובייקט ו- false אחרת;
- `public int numElements ()`
המחזירה את מספר האיברים בקבוצה;
- `public boolean subSet (Set other)`
המחזירה true אם הקבוצה other היא תת-קבוצה של הקבוצה שבאובייקט ו- false אחרת;
- `public void addToSet (int x)`
המקבלת מספר שלם x ומוסיפה אותו לקבוצה. שימו לב לכל מקרי הקצה האפשריים (חשבו מהם);
- `public void removeFromSet (int x)`
המקבלת מספר שלם x ומוציאה אותו מהקבוצה. שימו לב לכל מקרי הקצה האפשריים (חשבו מהם);
- `public String toString()`
המחזירה כמחרוזת את איברי הקבוצה בפורמט הבא: {a1, a2, a3}
הקפידו לשרשר בין האיברים פסיק בודד, ללא רווחים.

- `public Set intersection (Set other)`

המחזירה את קבוצת החיתוך של הקבוצה `other` עם הקבוצה שבאובייקט; השיטה מחזירה קבוצה חדשה ולא משנה את הקבוצה עליה הופעלה השיטה ולא את הקבוצה `other`.

- `public Set union (Set other)`

המחזירה את קבוצת האיחוד של הקבוצה `other` עם הקבוצה שבאובייקט; השיטה מחזירה קבוצה חדשה ולא משנה את הקבוצה עליה הופעלה השיטה ולא את הקבוצה `other`.

- `public Set difference (Set other)`

המחזירה את קבוצת ההפרש של הקבוצה שבאובייקט פחות הקבוצה `other`; השיטה מחזירה קבוצה חדשה ולא משנה את הקבוצה עליה הופעלה השיטה ולא את הקבוצה `other`.

כתבו מה סיבוכיות הזמן וסיבוכיות המקום של כל שיטה שכתבתם.

שימו לב: אסור להשתמש במחלקות מוכנות כבר של Java. מותר ורצוי להשתמש במחלקות שניתנו בהרצאה ונמצאות בחוברת השקפים.

שימו לב לא לכתוב קוד מיותר (שכבר נכתב) אלא להשתמש במחלקות המתאימות.

שימו לב:

הקפידו ששמות המחלקות והשיטות יהיה **בדיוק** כפי שהוגדרו.

שימו לב,

באתר הקורס תמצאו גם טסטר לבדיקת האיות והפרמטרים של השמות של השיטות והמחלקה שאתם צריכים לכתוב. חובה עליכם לבדוק את המחלקה שכתבתם בטסטר זה, ולהגיש אותה רק אם הטסטר עובר קומפילציה. שימו לב שהטסטר לא מכסה את כל האפשרויות, ובפרט לא את מקרי הקצה. הוא רק בודק את השמות של השיטות במחלקות כלומר שגיאות קומפילציה. מאד מומלץ להוסיף לו בדיקות. אם יש שיטה שלא כתבתם, כתבו חתימה והחזירו ערך סתמי כדי שהטסטר ירוצו עם המחלקה ללא שגיאות קומפילציה. אם הטסטר לא ירוץ ללא שגיאות קומפילציה הציון במטלה יהיה אפס.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו ששמות השיטות והמחלקות יהיו בדיוק לפי הוראות הממ"ן.
3. את התשובות לשאלות יש להגיש בשני קבצי Java הבאים: `Set.java`, `IntNode.java`, ארוזים יחד בתוך קובץ zip יחיד. אין לשלוח קבצים נוספים.

בהצלחה

מטלת אופל 03

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידה 12 **נושא המטלה:** מחסנית, תור, עצים בינריים

מספר השאלות: 20 **משקל המטלה:** 1 נקודות

סמסטר: 2022 **מועד אחרון להגשה:** 18.6.2022

(ת)

סטודנטים יקרים,
מטלת אופל, היא שאלון רב ברירה (שאלון אמריקאי, ממ"ח) שאתם עונים עליו באתר הקורס.
כל הפרטים הנדרשים נמצאים באתר הקורס.
הכניסה למטלה והגשתה אפשרית מתאריך 12.6 בשעה 10:00 עד לתאריך 18.6 בשעה 23:59.
אתם מוזמנים להתנסות.

המטלה נבדקת בצורה ממוחשבת. אין לשלוח את פתרונה למנחה!

שימו לב, המנחה לא יכול לדחות לכם את תאריך הגשת מטלת אופל. זה אוטומטי והוא לא יכול להשפיע על כך. לכן אין טעם לפנות למנחה בבקשות כאלו.

זכרו לשמור את האסמכתא שאתם מקבלים מהמחשב לאחר הגשת מטלת אופל. אם לא קיבלתם מספר אסמכתא כאילו לא הגשתם את המטלה.

הנחיות לפתרון מטלת אופל

יש לקרוא כל שאלה פעמים מספר ולהתייחס לכל מלה בה. קריאה זהירה והבנה מדויקת של משמעות כל משפט בשאלה הן תנאי ראשון להצלחתכם במטלה.
לכל שאלה יש רק תשובה נכונה אחת. קראו תחילה את כל האפשרויות הנתונות, החליטו מהי האפשרות הנכונה ביותר מבין כל האפשרויות ואז סמנו אפשרות זו.
אם נדמה לכם שיש לשאלה אחת שתי תשובות נכונות, או אף שלוש, ייתכן כי תגלו, לאחר קריאת כל התשובות, תשובה אחת האומרת "שלוש התשובות הקודמות נכונות". במקרה כזה, מובן שתסמנו תשובה זו ואותה בלבד כנכונה. אם לא מופיע משפט מסוג זה, הרי רק אחת התשובות נכונה. קיימת גם אפשרות שאין כל תשובה נכונה, ובמקרה כזה תינתן לכם אפשרות לסמן כנכונה את התשובה: "אין אף תשובה נכונה".