

תוכנה 1 - סמסטר א' 2020/21

תרגיל מספר 2

הנחיות כלליות:

קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.

- הגשת התרגיל תעשה במערכת ה-moodle בלבד (<http://moodle.tau.ac.il>).
- יש להגיש קובץ **zip** יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש aviv2 יקרא הקובץ aviv2_hw2.zip). שימו לב שלא מדובר בשם שלכם, אלא במשתמש האוניברסיטאי, איתו אתם מתחברים למודל למשל.
- קובץ ה-zip יכיל:
 1. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז. (אפשר לכתוב בכל שפה – עברית או אנגלית. אתם לא נדרשים להיצמד לפורמט ספציפי, רק שהפרטים יהיו ברורים).
 2. קבצי ה-java של התוכניות אותם התבקשתם לממש. יש לשים את תיקיית ה-src ובתוכה כל קבצי הג'אווה של סעיפים התרגיל **כשהם נמצאים בתוך היררכיית התיקיות כפי שקיבלתם בשלד הקוד – כלומר בתוך src תהיה התיקייה il, ובתוכה ac וכן הלאה.**הקפידו ששמות הקבצים יהיו זהים לשמות שהתבקשתם ושימו לב שכולם מתחילים במילה Assignment. אין להוסיף קבצים או תיקיות נוספות! הקפידו שכאשר פותחים את הזיפ, מיד רואים את תיקיית ה-src והיא לא נמצאת בתוך עוד תיקיה, למשל תיקיה ששמה זהה לשם ה-zip או הפרויקט (טעות נפוצה).

שימו לב: חשוב מאד להקפיד על פורמט ההגשה.

כלומר אין להגיש קבצי rar, ויש לקרוא לקבצים בדיוק לפי ההנחיות שקיבלתם. כמו כן אין לצרף תיקיות או קבצים נוספים. **אי עמידה בהנחיות ההגשה תגרור הורדה משמעותית בניקוד!** נדגיש שוב: בזיפ יש רק קובץ details.txt, ותיקיית ה-src שבתוכה כל קבצי הג'אווה שהתבקשתם לכתוב בתרגיל.

בנוסף שימו לב: לפני ההגשה, עליכם להריץ כל תכנית על מספר קלטים שונים (ובפרט לפי הדוגמאות המפורטות בתרגיל) כדי לוודא שהיא אכן פועלת כראוי. זכרו שהארגומנטים בשורת הפקודה (כלומר, הארגומנטים של פונק' ה-main) תמיד מתקבלים כמערך מחרוזות. לכן, אם ברצונכם לעבוד עם מספרים או תווים יש לבצע המרה – אפשר למשל בעזרת השיטות שנלמדו בתרגול.

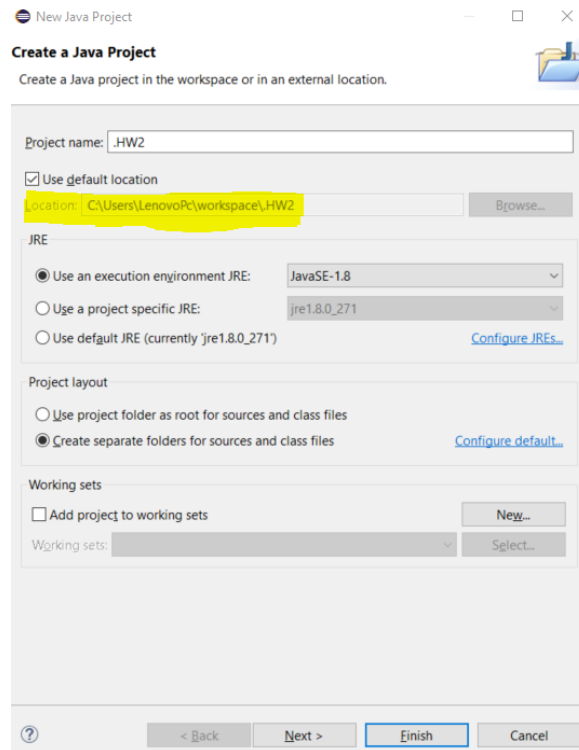
ניתן להניח כי הקלט תקין, למשל, אם נדרשתם לבצע פעולה על הארגומנט הרביעי - שקיים אחד כזה. כמו כן, **הקפידו על שמות המחלקות במדויק. שם מחלקה לא נכון יגרור הורדה בציון התרגיל.**

בכל אחד מהסעיפים הבאים עליכם לכתוב תוכנית(=מחלקה עם שירות main) המבצעת משימה מסוימת על הארגומנטים משורת הפקודה. לתרגיל בית זה יש שלד קוד קיים ועבור כל שאלה יש תוכנית קיימת עם שם מתאים לשאלה. לדוגמא, עבור שאלה 1 יש את התוכנית בשם Assignment02Q01 ועליכם לכתוב את הפיתרון אליה בקובץ שלה. עבור שאלה 2 כתבו בתוכנית בשם Assignment02Q02 וכדומה. יש למלא את הקוד המתאים בתוכנית המתאימה לשאלה הרלוונטית לפי השם. שימו לב שאין למחוק או לערוך קוד קיים בשלד בשאלות שבו יש כזה. הקוד הקיים נמצא שם על מנת שתוכלו להוציא את הפלט התקין בתצורה נכונה.

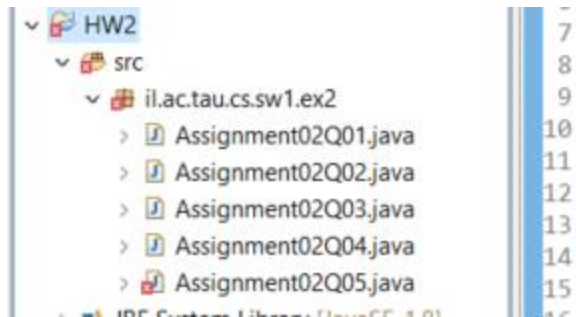
הנחיות ליצירת פרוייקט חדש ב-Eclipse וייבוא השלד של תרגיל 2:

יש ליצור פרוייקט חדש ב-Eclipse ע"י בחירה ב- File-> New -> Java Project. רשמו את שם הפרוייקט (השם הוא לבחירתכם) בחלון שמופיע, וודאו שמומנת הבחירה Use default location, ושימו לב שמתחתיה מופיע המיקום של ה-workspace בו ה-Eclipse שומר את התוכנית שלכם (מסומן בצהוב בצילום מסך המצורף).

זה המיקום בו נמצאת התיקיה של התוכנית ובתוכה תיקיית ה-src שצריך להגיש (לאחר סיום כתיבת התוכנית, כמובן). כמו כן, וודאו באותו החלון כי הגירסה של ה-JRE היא 1.8 (כלומר ג'אוה 8). לחצו על Finish.



כעת, נווטו (מחוץ לאקליפס) אל התיקיה של הפרוייקט שיצרתם ב-workspace, והיכנסו בתוכה לתיקיית ה-src (שאמורה להיות ריקה). העתיקו את תוכן תיקיית ה-src (רק את התוכן, לא כולל התיקיה עצמה) שחילצתם מקובץ הזיפ אל תוך תיקיית ה-src המקומית ב-workspace. לסיום, חזרו לאקליפס, לחצו על הפרוייקט שלכם לחיצה ימנית, ובחרו refresh. כעת מחלקות שסופקו לכם אמורות להופיע בפרוייקט באקליפס. מצורף צילום נוסף להמחשה (התעלמו מכך ששם הפרוייקט לא תואם לשם שנקבע בצילום מסך הקודם).



שאלה 1

[20 נק'] חימום:

בג'אווה כל char מיוצג ע"י ערך מספרי. כדי לדעת מהו הערך של כל תו בג'אווה צריך להסתכל [בטבלת אסקי table](#), שזהו סטנדרט בינלאומי למיפוי אותיות למספרים. כך לפי טבלת אסקי הערך של התו 'A' הוא 65, של 'b' הוא 98 והערך של '!' הוא 33.

כיתבו תכנית במחלקה בשם Assignment02Q01 בשלד קוד הנתון הקוראת את כל המחרוזות בקלט, בודקת מהו ערך האסקי של התו הראשון במחרוזת, ובמידה והוא מתחלק ב-3 והוא גם מספר זוגי מדפיסה אותו (את התו) בשורה נפרדת למסך. ניתן להיעזר באופרטור [%](#). לדוגמא עבור קלט:

Before a E f

התכנית תדפיס את:

B

f

(כיוון שערך האסקי של B הינו 66, של a הינו 97, של E 69 ושל f 102).

לא ניתן להניח כי כל הארגומנטים הינם תווים בודדים (כלומר מחרוזת אשר מכילה תו יחיד). במקרה והמחרוזת היא איננה תו יחיד, למשל "ab", יש לחלץ רק את התו הראשון של המחרוזת (וגם להדפיס רק אותו), במקרה הזה a.

היעזרו בתירגול בהרצאה ובעבודה העצמית. (10 נק')

שאלה 2

[20 נק'] כתבו תכנית במחלקה בשם Assignment02Q02 בשלד הנתון אשר משמשת כמחשבון לחישוב המספר פאי.

התכנית מקבלת בשורת הפקודה מחרוזת אשר מייצגת מספר טבעי (ניתן להניח כי הועברה מחרוזת אחת בלבד אשר אכן מייצגת מספר טבעי חיובי גדול מ-0) ומחשבת את המספר פאי באמצעות הביטוי המתמטי הבא:

$$\pi = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \dots \right)$$

כאשר הארגומנט שהתכנית קיבלה יסמן כמה איברים סכמנו.

התכנית תדפיס למסך את הערך שהתקבל ואת הערך של המספר PI כפי שניתן על ידי ה-jdk, שלו ניקרא באמצעות הפקודה Math.PI.

לדוגמא עבור הקלט 4, התכנית תחשב את הביטוי: $4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7}\right)$, ותדפיס למסך:

3.141592653589793 2.8952380952380956 (שני הערכים מודפסים באותה השורה, מופרדים ברווח יחיד). שימו לב להשתמש בפקודת ההדפסה אשר נתונה לכם בשלד הקוד.

לדוגמא עבור הקלט 100 התכנית תדפיס למסך: 3.141592653589793 3.1315929035585537

לצורך החישוב השתמשו במשתנים מסוג double (לא float)

שאלה 3

[20 נק'] כתבו תכנית במחלקה בשם Assignment02Q03 בשלד הנתון אשר מקבלת כקלט בשורת הפקודה מספר טבעי (יש להניח כי המחרוזת אכן מייצגת מספר טבעי גדול או שווה ל-3), ומדפיסה מספר כזה של איברי סדרת פיבונצ'י (כלומר אם המספר הוא 5, נדפיס את חמשת המספרים הראשונים בסדרה) ואת מספר האיברים מתוכם שהם זוגיים. שני האיברים הראשונים הינם 1.

לדוגמא, עבור המספר 20 פלט התכנית יהיה (בפורמט הבא):

The first 20 Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765

The number of even numbers is: 6

שימו לב לפורמט ההדפסה: בשורה הראשונה מודפס כמה איברים הצגנו, בשורה נפרדת כל הסדרה מופרדת ברווח יחיד (ללא רווח בסוף השורה), ובשורה האחרונה את **מספר הזוגיים מתוכם**. יש להקפיד על פורמט ההדפסה כולל המלל המדויק שמוצג כאן. בשלד הקוד נתונות פקודות ההדפסה למלל, יש רק להוסיף את הדפסת החישובים שלכם לקוד.

דוגמא נוספת עבור הקלט 10:

The first 10 Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55

The number of even numbers is: 3

שאלה 4

[20 נק'] התבוננו בפונקציית ה-main הנתונה בתוכנית Assignment02Q04 אשר אמורה לייצר מערך עם 20 המספרים הראשונים הראשונים. המס' הראשון הראשון 2 נוסף למערך בשלב האתחול. לאחר מכן, הפונקציה עוברת על מספרים אי-זוגיים החל מ-3 ובודקת עבור כל מספר אם הוא ראשוני. כדי לגלות זאת, הפונקציה מנסה לחלק כל מספר בכל המספרים הראשונים שהתגלו עד כה, עד לגודל השורש של מספר זה (מכיוון שלא ייתכן שהמספר הוא מכפלה של שני שלמים הגדולים מהשורש שלו...).

למרבה הצער, נפל באג בתכנית, והיא מדפיסה את הפלט

[2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49]

אשר מכיל כמה מספרים שאינם ראשוניים (מודגשים בצהוב).
עליכם לגלות מדוע. למשל, התכנית מצאה ש-3 הוא מספר ראשוני, אבל לא מצאה ש-9 מתחלק ב-3.
קראו את [המדריך לשימוש ב-Eclipse של debugger](#), פרקים 1-3.

הריצו את התכנית במצב דיבאג :



ועקבו אחרי שלבי הריצה, עד שתמצאו את הגורם לבעיה.

תקנו את התכנית (יתכן שצריך לתקן במספר מקומות) והגישו את התכנית המתוקנת.
אם הצלחתם, התכנית תדפיס את המערך הנכון:

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71]

שימו לב כי אין אמנם אין מגבלה על אופי התיקון, אך התיקון הנדרש לא דורש שינוי גדול בקוד.

שאלה 5

[20 נק'] ראינו שמערכים יכולים להיות בעלי יותר ממימד אחד. לכן, ניתן להשתמש במערך דו-מימדי על מנת לייצג מטריצה. כיתבו תוכנית אשר מקבלת מערך דו-מימדי של מספרים אשר מייצג מטריצה ריבועית ומדפיסה את המטריצה ובנוסף מדפיסה את אותה המטריצה לאחר שעברה סיבוב של 90 מעלות ימינה (ראו דוגמאות למטה).

לנוחיותכם, בשלד הקוד קיימות שורות קוד שמקבלות את הקלט ומכניסות אותו למערך דו-מימדי. כך שהחלק של קבלת הקלט בargs נעשה עבורכם מראש ויש להשתמש בקוד הקיים ולא לשנות אותו.

הסבר על הקוד הקיים בשלד-

הקלט מתקבל בargs באופן הבא: האיבר הראשון args[0] יכיל מספר N שמייצג את מימד המטריצה, ולאחר מכן יתקבל N^2 מספרים שמייצגים את ערכי המטריצה. N האיברים הבאים בargs מייצגים את השורה הראשונה במטריצה, N האיברים שלאחר מכן ייצגו את השורה השנייה במטריצה וכן הלאה.

לדוגמה בהינתן האיברים הבאים בargs שמקבלת מתודת main:

3 1 2 3 4 5 6 7 8 9

הקוד הקיים בשלד יוצר מערך דו מימדי בגודל 3 על 3 (3 שורות ו3 עמודות)

[1, 2, 3]

[4, 5, 6]
[7, 8, 9]

עבור מטריצה כזאת על התוכנית להדפיס את המטריצה אחרי שעברה סיבוב של 90 מעלות:

[7, 4, 1]
[8, 5, 2]
[9, 6, 3]

דוגמא נוספת עבור הקלט:

4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

הקוד הקיים בשלד יוצר מערך דו מימדי בגודל 4 על 4 (4 שורות ו4 עמודות)

[1, 2, 3, 4]
[5, 6, 7, 8]
[9, 10, 11, 12]
[13, 14, 15, 16]

עבור מטריצה כזאת על התוכנית להדפיס את המטריצה אחרי סיבוב של 90 מעלות:

[13, 9, 5, 1]
[14, 10, 6, 2]
[15, 11, 7, 3]
[16, 12, 8, 4]

ניתן להניח כי הקלט תקין – המספר הראשון N מייצג את מימד המטריצה הוא מספר שלם גדול מ0, ושאריו מגיעים כל ה- N^2 מספרים שלמים שמייצגים את ערכי המטריצה כפי שתואר.

אתגר - נסו להתייעל בשימוש בזיכרון. האם תוכלו לעשות זאת ללא יצירה של מערך נוסף מלבד המערך הדו-מימדי שמייצג את המטריצה? כלומר שהשינויים יהיו במטריצה הנתונה in place ובקוד שלכם לא יהיה שימוש ב new מעבר לקוד הנתון. זו לא חובה וניתן להגיש קוד שמשתמש בכל כמות זיכרון ומערכים שתרצו.

בהצלחה!!