

# תוכנה 1 – אביב 2019/20

## תרגיל מספר 1

### 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 aviv1 יקרא הקובץ aviv1\_hw1.zip). שימו לב שלא מדובר בשם שלכם, אלא במשתמש האוניברסיטאי, איתו אתם מתחברים למודל, למשל. קובץ ה-zip יכיל:
  - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז. (אפשר לכתוב בכל שפה – עברית או אנגלית. לא נדרש להיצמד לפורמט ספציפי. רק שהפרטים יהיו ברורים).
  - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש. מתוך התוכנית שתכתבו יש לשים **רק את תיקיית ה-src** ובתוכה שני קבצי java בשם Triangles ו-Strings. הקפידו שכאשר פותחים את הזיפ, מיד רואים את תיקיית ה-src והיא לא נמצאת בעוד תיקיה, למשל תיקיה ששמה זהה לשם ה-zip או הפרוייקט (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 src שבתוכה הקבצים Triangles.java ו-Strings.java.

### הנחיות ליצירת פרוייקט חדש ב-Eclipse:

יש ליצור פרוייקט חדש ב-Eclipse ע"י בחירה ב- File-> New -> Java Project. רשמו את שם התוכנית בחלון שמופיע, וודאו שמסומנת הבחירה Use default location, ושימו לב שמתחתיה מופיע המיקום של ה-workspace בו ה-Eclipse שומר את התוכנית שלכם. 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).

שימו לב שיש להגיש בזיפ מתוך התוכנית את תיקיית ה-src בלבד!

## 1. תוכנית לדוגמא - משולשים

כתבו תכנית ג'אוהה המקבלת בשורת הפקודה שלושה מספרים ובודקת האם הם יכולים לשמש כאורכי צלעות במשולש שווה שוקיים או במשולש ישר זווית.

להזכירכם, הצלעות  $x$   $y$   $z$  מגדירות משולש ישר זווית חוקי שבו  $x$  ו  $y$  הם הניצבים ו  $z$  הוא היתר, אם מתקיים  $x^2 + y^2 = z^2$ .

משולש עם צלע שאינה גדולה מאפס לא נחשב משולש תקני. במידה ואחד המספרים הוא אי-חיובי התוכנית תדפיס הודעה מתאימה.

ניתן להניח שכל המספרים שלמים (כלומר אין צורך להתייחס או לטפל במקרה של קלט שאינו מספרים שלמים בלבד).

מותר להשתמש במבנה הבקרה if-else ([קישור לתיעוד רלוונטי](#)) עליו נלמד בקרוב, כמו גם באופרטורי השוואה (שכוללים בין היתר == ו- != שמציינים בהתאמה שוויון ואי שוויון) ואופרטורים לוגיים. האופרטור הלוגי && בין שני תנאים מציין קשר של "and", ואילו האופרטור || מציין קשר של "or". האופרטור ! מציין קשר "not". אסור (וגם אין צורך) להשתמש בלולאות.

שם התוכנית הוא Triangles.

להזכירכם משולש ישר זווית נקרא באנגלית right triangle, ומשולש שווה שוקיים נקרא isosceles triangle. כמו כן, משולש שווה צלעות הוא כמובן בפרט משולש שווה שוקיים.

שימו לב, כי משולש שאורכי צלעותיו הם מספרים שלמים אינו יכול להיות גם שווה שוקיים וגם ישר זווית (ויש לכך הוכחה פשוטה וקצרה).

הודעה מתאימה מודפסת במקרה בו כל אחד מהתנאים מתקיים לבדו, כמו גם במקרה בו אף אחד משניהם לא מתקיים.

כמו כן, שימו לב שסדר הארגומנטים לא משנה. כלומר לא מובטח שהגדלים ניתנים בסדר עולה או יורד, ובפרט לא מובטח במשולש ישר זווית איזה מהארגומנטים מייצג את היתר. יש לבדוק את כל האפשרויות.

בתרגיל זה נתעלם, לשם פשטות, מדרישות חיוניות לכך שהצלעות מייצגות משולש תקני, כמו אי שוויון המשולש.

דוגמאות לקלטים אפשריים ולפלט הנדרשים:

עבור הקלט לתוכנית:

```
3 4 5
```

יודפס הפלט הבא למסך:

```
The input (3,4,5) defines a right triangle!
```

עבור הקלט לתוכנית:

```
5 3 4
```

יודפס הפלט הבא למסך:

```
The input (5,3,4) defines a right triangle!
```

עבור הקלט לתוכנית:

2 3 2

יודפס הפלט הבא למסך:

The input (2,3,2) defines an isosceles triangle!

עבור הקלט לתוכנית:

7 7 7

יודפס הפלט הבא למסך:

The input (7,7,7) defines an isosceles triangle!

עבור הקלט לתוכנית:

1 9 2

יודפס הפלט הבא למסך:

The input (1,9,2) does not define an isosceles or a right triangle!

עבור הקלט לתוכנית:

2 -1 3

יודפס הפלט הבא למסך:

Invalid input!

עבור הקלט לתוכנית:

2 0 3

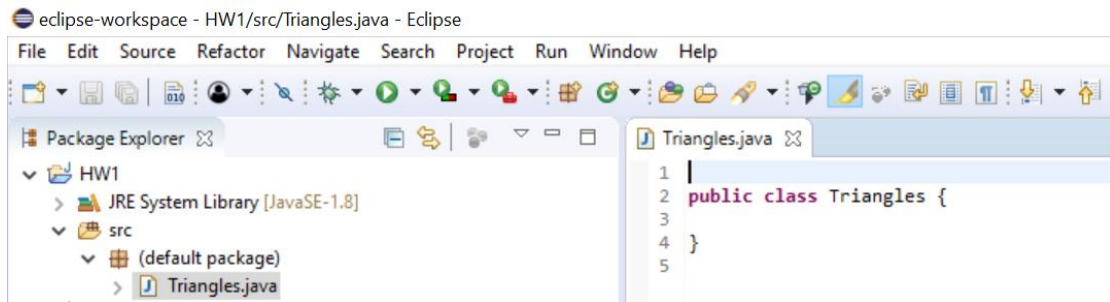
יודפס הפלט הבא למסך:

Invalid input!

### הדרכה:

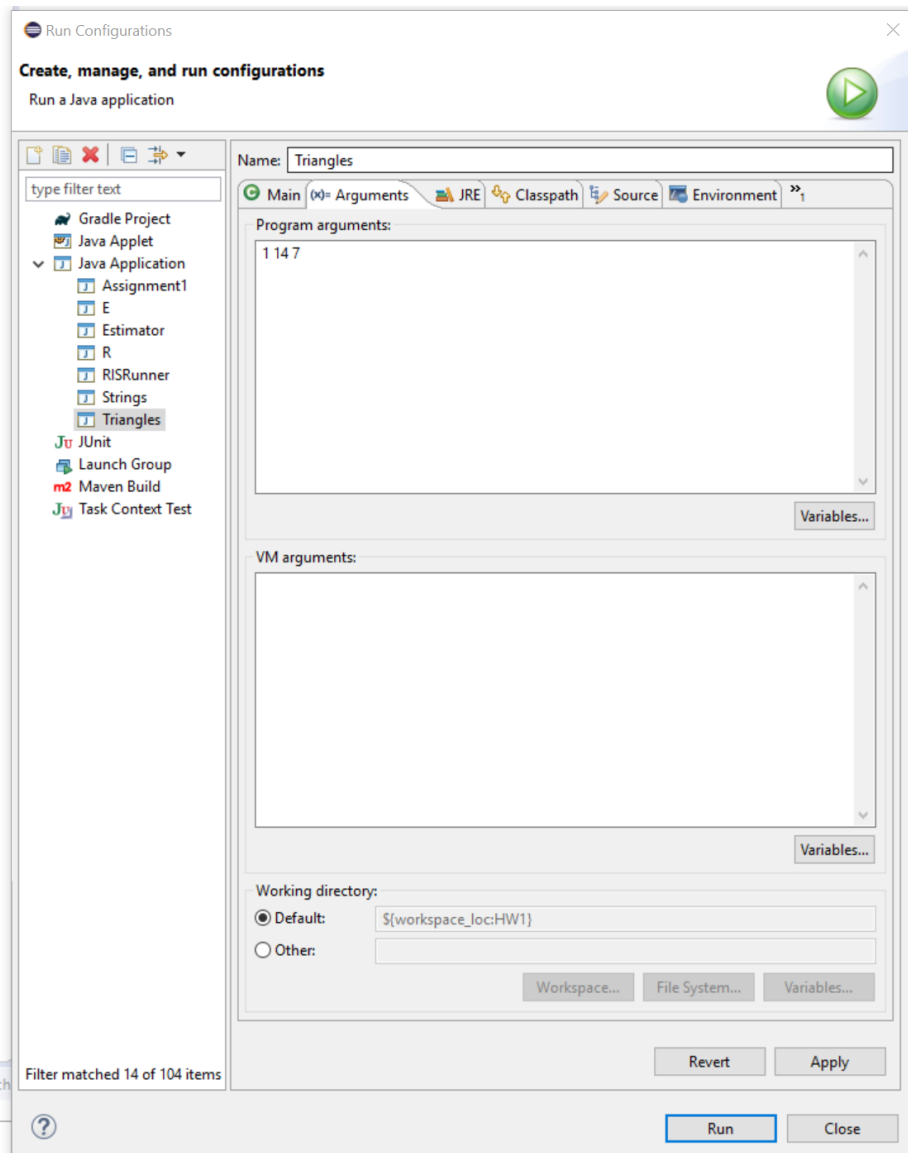
- ראשית צרו פרויקט חדש באקליפס ובו מחלקה בשם Triangles.
- בפונקציית ה-main של המחלקה, השתמשו במערך המחרוזות args על מנת לקבל את הארגומנטים שנשלחו בשורת הפקודה (המערך args הינו הפרמטר היחיד של פונקציית ה-main).
- ניתן להניח שהתוכנית מקבלת שלושה ארגומנטים, ושלושתם ניתנים להמרה ל int. במצגת התרגול ניתן למצוא דוגמא להמרה ממחרוזת למספר שלם.
- בפלט התוכנית יש רווח בודד בין כל שתי מילים סמוכות, אין להשתמש בטאבים (t).
- בסוף הדפסת השורה ניתן לרדת שורה, אך זו לא חובה.

לאחר יצירת הפרוייקט והמחלקה, האקליפס נראה כך:



הערה: היות וחלק מבידיקת תרגילי הבית מתבצע ע"י מערכת אוטומטית, יש להקפיד על פלטים מדויקים בהתאם לדוגמא, עד לרמת הפיסוק והריווח. כמו כן, הקפידו על שמות מחלקות כפי שמוגדרות בתרגיל.

תזכורת: נידון בתרגול על העברת ארגומנטים לתוכנית (מוקדש לכך גם שקף במצגת). בנוסף, לשם הבהירות, אנו מצרפים צילום מסך של העלאת שלושת הארגומנטים (1 14 7) באקליפס.



## 2. תוכנית לדוגמא - מחרוזות

כתבו תוכנית ג'אווה המקבלת בשורת הפקודה שלוש מחרוזות, ובודקת האם המחרוזת השלישית יכולה להיווצר על ידי "שרשור" בסדר כלשהו של שתי המחרוזות האחרות, עם בדיוק אות אחת כלשהי שמפרידה בין שתי המחרוזות המשורשרות (כלומר אנחנו בודקים מצב של "כמעט שרשור" כאשר בשרשורים שלנו תמיד יש תו אחד מפריד, כך שזה למעשה שרשור של שלוש מחרוזות, כאשר הראשונה והאחרונה התקבלו כקלט, והאמצעית היא אות כלשהי, שיכולה להיות כל אות).

למשל, אם שלושת המחרוזות בקלט הן "abg" "ht" "abg", אז המחרוזת abght היא אכן שרשור של 2 המחרוזות הראשונות, כאשר האות g היא האות המפרידה. שימו לב, כי המחרוזת הראשונה בקלט היא לא בהכרח המחרוזת הראשונה בשרשור. כלומר, הסדר בשרשור בין 2 המחרוזות הראשונות בקלט יכול להיות כל אחת משתי האפשרויות (מבחינתו השרשור מתקיים אם לפחות אחת מתוך שתי האפשרויות מתאימה).

ניתן להניח כי הקלט לתוכנית הוא מערך של שלושה ארגומנטים שכל אחד מהם הוא מחרוזת המורכבת מאותיות אנגליות בלבד (כלומר לא צריך לבדוק שאין תווים מסוג אחר כמו רווחים). כמו כן, ניתן להניח שכל מחרוזת מכילה לפחות אות אחת. עם זאת, המחרוזת יכולה להכיל אותיות קטנות וגדולות, אך בבדיקת השרשור אנחנו נתייחס לגירסה הגדולה והקטנה של אות כלשהי בתור בדיוק אותה האות, כלומר נתעלם מהגודל.

למשל המחרוזת "AnDNoW" היא מבחינתנו שרשור תקין של המחרוזות "aNd" ו-"Ow" עם התו המפריד "N" בין השרשורים.

אם מספר האותיות במחרוזת השלישית שמפרידות בין שתי המחרוזות הראשונות הוא לא בדיוק 1, אז לא מדובר בשרשור תקין לפי ההגדרה שלנו.

שם התוכנית הוא Strings. אנו נכתוב אותה בתור מחלקה נוספת **באותו הפרוייקט** בו מופיעה גם המחלקה מהתרגיל הקודם.

מותר להשתמש במבנה הבקרה if-else ([קישור לתיעוד רלוונטי](#)) עליו נלמד בקרוב, כמו גם באופרטורי השוואה (שכוללים בין היתר == ו- != שמציינים בהתאמה שוויון ואי שוויון) ואופרטורים לוגיים. האופרטור הלוגי && בין שני תנאים מציין קשר של "and", ואילו האופרטור || מציין קשר של "or". האופרטור ! מציין קשר "not". אסור (וגם אין צורך) להשתמש בלולאות.

כמו כן, את כל הפעולות על המחרוזות עליכם לבצע באמצעות המתודות המובנות של המחלקה String, אותן תוכלו למצוא [בתיעוד הרשמי](#). במיוחד, אם בשלב מסוים הינכם בודקים שוויון בין 2 מחרוזות, הקפידו להשתמש במתודות מתאימות מהתיעוד, במקום באופרטור השוויון == (מסיבות שנלמד עליהן בקרוב).

יש הודעה מתאימה שמודפסת במקרה בו הקלט מקיים את תכונת השרשור שהגדרנו, והודעה מתאימה אחרת במקרה שלא, כפי שתוכלו לראות מדוגמאות הקוד שלפניכם.

דוגמאות לקלטים אפשריים ולפלטים הנדרשים:

עבור הקלט לתוכנית:

```
CrY Don dontcry
```

יודפס הפלט הבא למסך:

```
dontcry is a concatenation.
```

עבור הקלט לתוכנית:

cry dont dOncry

יודפס הפלט הבא למסך:

dOncry is not a concatenation.

עבור הקלט לתוכנית:

do tcRy Dontcry

יודפס הפלט הבא למסך:

Dontcry is a concatenation.

עבור הקלט לתוכנית:

Abc a c

יודפס הפלט הבא למסך:

c is not a concatenation.

עבור הקלט לתוכנית:

ABB cC AbBC

יודפס הפלט הבא למסך:

AbBC is not a concatenation.

עבור הקלט לתוכנית:

ab bcd abcd

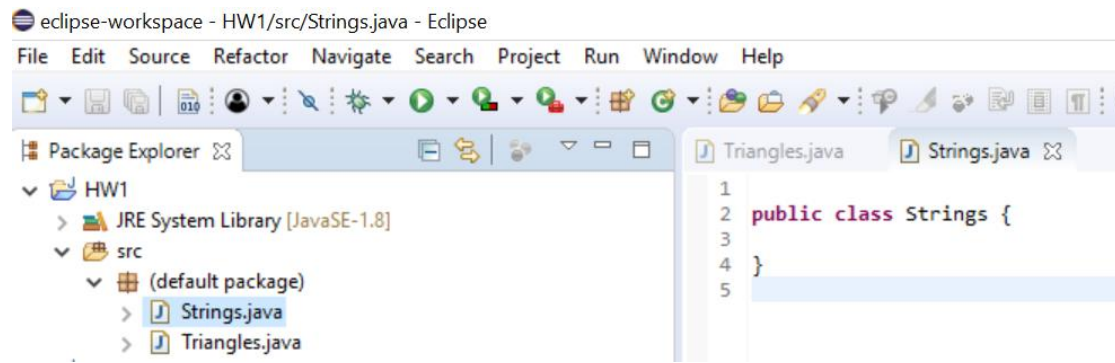
יודפס הפלט הבא למסך:

abcd is not a concatenation.

## הדרכה:

- בפרויקט שיצרתם עבור התרגיל הקודם, הוסיפו מחלקה בשם Strings.
- בפונקציה ה-main של המחלקה, השתמשו במערך המחרוזות args על מנת לקבל את הארגומנטים שנשלחו בשורת הפקודה (המערך args הינו הפרמטר היחיד של פונקציית ה-main).
- בפלט התוכנית יש רווח בודד בין כל שתי מילים סמוכות, אין להשתמש בטאבים (\t).
- בסוף הדפסת השורה ניתן לרדת שורה, אך זו לא חובה.

לאחר יצירת הפרוייקט, המחלקה בתרגיל הקודם, והמחלקה בתרגיל הנוכחי, האקליפס נראה כך:



הערה: שימו לב כי יש מתודת main (שונה) בכל אחד משני הקבצים בפרויקט. כאשר אתם לוחצים על הרצה באקליפס, תורץ ה-main המתאימה בהתאם לטאב שפתוח באותה עת באקליפס.

נזכיר שוב: היות וחלק מבדיקת תרגילי הבית מתבצע ע"י מערכת אוטומטית, יש להקפיד על פלטים מדויקים בהתאם לדוגמא, עד לרמת הפיסוק והריווח. כמו כן, הקפידו על שמות מחלקות כפי שמוגדרות בתרגיל.

# בהצלחה !