

# תוכנה 1 – חורף 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 stav1 יקרא הקובץ stav1\_hw1.zip). שימו לב שלא מדובר בשם שלכם, אלא במשתמש האוניברסיטאי, איתו אתם מתחברים למודל למשל. קובץ ה-zip יכול:
  - 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 קובץ java יחיד בשם Assignment1. 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).
  - ג. קובץ pdf בשם answers.pdf המכיל את התשובות לשאלות 2-3 (מותר לענות גם באנגלית).

**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, קובץ answers.pdf, ותיקיית src שבתוכה קובץ Assignment1.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. הכרות עם Eclipse

- ✓ התקינו Java ואת סביבת העבודה של Eclipse על פי ההוראות בקישור הבא:  
<http://www.cs.tau.ac.il/courses/software1/1718b/misc/workenv.pdf>
- ✓ עיינו בסעיפים 5-9 ו-15 של מדריך האקליפס שבקישור הבא:  
<http://www.vogella.com/tutorials/Eclipse/article.html>

## 2. מונחים בסיסיים ב-Java

שימו לב, בחלק מהנושאים המכוסים בשאלה זו נגענו בתרגול, והשאר יכוסו בהרצאה הבאה.

- עבור כל אחד מהמשפטים הבאים ציינו אם הוא נכון או לא. אין צורך לנמק.
1. ב-Java ישנם בדיוק שמונה טיפוסים פרימיטיביים.
  2. המפרש של Java מייצר קבצי exe מקבצי ה-byte-code.
  3. האיבר הראשון במערך args שמקבלת מתודת ה-main הוא תמיד שם התוכנית.
  4. לאחר הידור (compilation) נוצרים קבצים בסיומת java.
  5. פורמט byte-code מסייע בין היתר להריץ את אותו הקוד על סביבות שונות.
  6. לא ניתן להגדיר טיפוס הפניה חדש מעבר לטיפוסים הקיימים בשפה.
  7. מערך של int הוא טיפוס פרימיטיבי.
  8. קובץ class הוא למעשה קובץ טקסט אשר מכיל פקודות בשפת java וניתן לפתוח אותו בכל עורך טקסט.
  9. המהדר של Java קורא קבצים בפורמט byte-code.
  10. המהדר (Java compiler) מכיל את ה-JVM.
  11. המהדר של Java הינו חלק מה-JRE.
  12. ניתן להריץ קובץ byte-code שנוצר במערכת הפעלה אחת על מערכות הפעלה אחרות, ובלבד שיהיה מותקן JRE בגירסה המתאימה.
  13. המהדר של Java בודק את תחביר הקוד.
  14. ניתן להצהיר על משתנה מבלי לתת לו ערך התחלתי.
  15. מערך הארגומנטים שהמתודה main מקבלת יכול להיות ריק.
  16. המתודה Integer.parseInt מקבלת int וממירה אותו ל-String.
  17. תמיד ניתן להציב משתנה מטיפוס פרימיטיבי אחד בטיפוס פרימיטיבי אחר.
  18. String הוא מערך של char.
  19. ב-Java יש הבדל במשמעות כשמחליפים בשם של משתנה בין אות גדולה לאות קטנה (כלומר Java היא case-sensitive).

## 3. שגיאות קומפילציה

בסעיפים הבאים מובאות אפשרויות שונות לתכנית בשם MyClass. בכל סעיף עליכם להריץ את קוד התוכנית ולבדוק האם הקוד עובר קומפילציה. במידה והקוד מתקמפל, הריצו את התוכנית וכתבו מהו הפלט שמתקבל (האם יש בכלל פלט תקין או שיש שגיאת זמן ריצה). במידה ולא, ציינו מהי הודעת השגיאה (שגיאת קומפילציה) המתקבלת ומדוע היא נגרמת (יתכן ויש מספר בעיות – ציינו את כולן). שימו לב כי אקליפס תחת הגדרות תקינות מסמן אפילו לפני ההרצה את כל הנקודות בהן יש שגיאת

קומפילציה (אבל לא שגיאות זמן ריצה) ואם מצביעים עם העכבר על הסימן המתאים גם רושם את הבעיה. הקפידו על תשובות קצרות וענייניות. כמו כן, שימו לב כי נלמד על הבדל בין שגיאות זמן ריצה לקומפילציה בתרגול 2, לכן אם ההבדל לא ברור לכם בשלב זה, המתינו לתרגול והתקדמו לשאלות הבאות.

.1

```
public Class MyClass {
    public static void main(String args) {
        System.out.println("Hello Java!");
    }
}
```

.2

```
public class MyClass {
    public void main(String[] args) {
        System.out.println("Java Rocks!")
    }
}
```

.3

```
public class MyClass {
    public static void main2(String[] args) {
        System.out.println("Hello Java!");
    }
}
```

.4

```
public class MyClass {
    public static void main(String[] myArgs) {
        System.out.println("null is " + null);
    }
}
```

.5

```
public class MyClass {
    public void main(String[] args) {
        System.out.println("Java is the best! " + true);
    }
}
```

.6

```
public class MyClass {
    public void main(String[] args) {
        System.out.println('g' + 1.0);
    }
}
```

## 4. תוכנית לדוגמא

כתבו תכנית ג'אוהה המקבלת בשורת הפקודה שלושה מספרים ובודקת אם הם יכולים לשמש אורכי צלעות במשולש כך שמתקיים אי שוויון המשולש, כלומר שאין שתי צלעות שסכום אורכיהן קטן ממש מאורך הצלע השלישית.

בפרט התוכנית בודקת גם האם צלעות אלה יכולות לשמש כצלעותיו של משולש 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". אסור (וגם אין צורך) להשתמש בלולאות.

שם התוכנית הוא Assignment1.

להזכירכם משולש ישר זווית נקרא באנגלית right triangle.

שימו לב שיש הודעה שונה עבור קלט שהוא משולש חוקי שמקיים את אי שוויון המשולש אך אינו מקיים את משפט פיתגורס עבור משולש ישר זווית, לעומת קלט שגם מקיים את משפט פיתגורס.

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.

דוגמאות לקלטים אפשריים ולפלטם הנדרשים:

עבור הקלט לתוכנית:

3 4 5

יודפס הפלט הבא למסך:

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

עבור הקלט לתוכנית:

5 3 4

יודפס הפלט הבא למסך:

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

עבור הקלט לתוכנית:

2 3 2

יודפס הפלט הבא למסך:

The input (2,3,2) defines a valid triangle!

עבור הקלט לתוכנית:

1 9 2

יודפס הפלט הבא למסך:

The input (1,9,2) does not define a valid triangle!

עבור הקלט לתוכנית:

2 -1 3

יודפס הפלט הבא למסך:

Invalid input!

עבור הקלט לתוכנית:

2 0 3

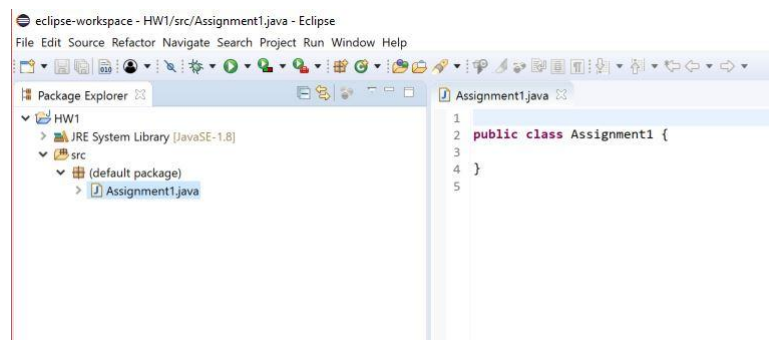
יודפס הפלט הבא למסך:

Invalid input!

## הדרכה:

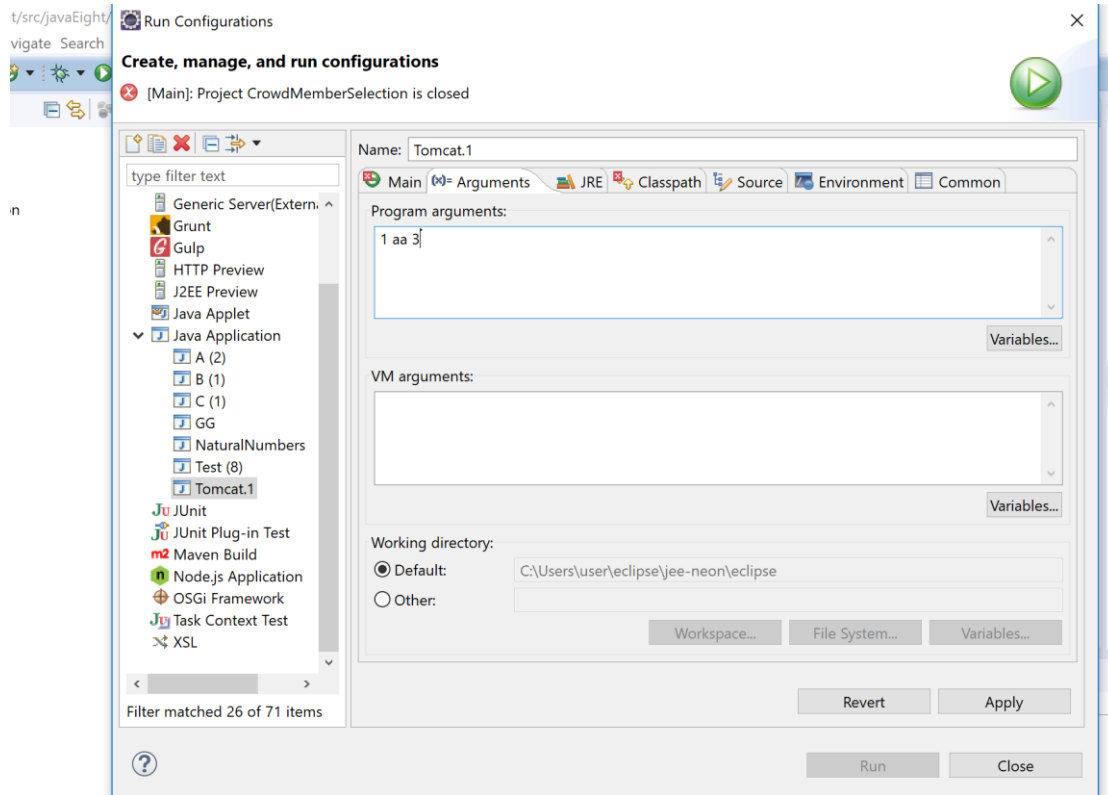
- ראשית צרו פרויקט חדש באקליפס ובו מחלקה בשם Assignment1.
- בפונקציית ה-main של המחלקה, השתמשו במערך המחרוזות args על מנת לקבל את הארגומנטים שנשלחו בשורת הפקודה (המערך args הינו הפרמטר היחיד של פונקציית ה-main).
- ניתן להניח שהתוכנית מקבלת שלושה ארגומנטים, ושלושתם ניתנים להמרה ל int. במצגת התרגול ניתן למצוא דוגמא להמרה ממחרוזת למספר שלם.
- בפלט התוכנית יש רווח בודד בין כל שתי מילים סמוכות, אין להשתמש בטאבים (t).
- בסוף הדפסת השורה אחרונה ניתן לרדת שורה, אך זו לא חובה.

לאחר יצירת הפרוייקט והמחלקה האקליפס נראה כך:



הערה: 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.

תזכורת: נידן בתרגול על העברת ארגומנטים לתוכנית (מוקדש לכך גם שקף במצגת). בנוסף, לשם הבהירות, אנו מצרפים צילום מסך של העלאת שלושת הארגומנטים (1 aa 3) באקליפס.



**בהצלחה !**