

# תוכנה 1 – חורף תשע"ז

## 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 stav יקרא הקובץ stav\_hw1.zip). קובץ ה-zip יכיל:
  - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
  - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש.
  - ג. קובץ טקסט יחיד המכיל את התשובות לשאלות 2-3.

---

### 1. הכרות עם Eclipse

- ✓ התקינו Java ואת סביבת העבודה של Eclipse על פי ההוראות בקישור הבא:  
<http://www.cs.tau.ac.il/courses/software1/1314b/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.

- א. ענו במשפט אחד על השאלות הבאות (תשובות ארוכות יותר לא תיבדקנה):
1. מהו תפקידו של המהדר בג'אווה (Java Compiler)?
  2. מהו תפקידו של המפרש בג'אווה (Java Interpreter)?
  3. מה היתרון בשימוש בפורמט Byte-code?
  4. מהי ה-Java Virtual Machine (JVM)?
  5. מהם ארגומנטי שורת פקודה (Command-line arguments)?

ב. עבור כל אחד מהמשפטים הבאים ציינו אם הוא נכון או לא. במידה והמשפט אינו נכון – נמקו.

1. המפרש של Java מייצר קבצי exe מקבצי ה byte-code.
2. קובץ class הוא למעשה קובץ טקסט אשר מכיל פקודות בשפת java וניתן לפתוח אותו בכך עורך טקסט.
3. המהדר של Java קורא קבצים בפורמט ה byte-code.
4. ניתן להריץ קובץ byte-code שנוצר במערכת הפעלה אחת על מערכות הפעלה אחרות, ובלבד שיהיה מותקן JRE בגרסה המתאימה.
5. המהדר של Java בודק את תחביר הקוד.

### 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. במידה ולא, ציינו מהי הודעת השגיאה המתקבלת ומדוע היא נגרמת. הקפידו על תשובות קצרות וענייניות.

1.

```
public class MyClass {
    public static void main(String[] args) {
        System.out.println("Hello java " + '!');
    }
}
```
2.

```
public class MyClass {
    public static void main2() {
        System.out.println("Hello" + "Java!");
    }
}
```
3.

```
public class MyClass {
    public static void main(String args) {
        System.out.println();
    }
}
```
4.

```
public class MyClass {
    public static void main(String[] myArgs) {
        System.out.println("Java is number " + 2-1);
    }
}
```
5.

```
public class MyClass {
    private static void main(String[] args) {
        System.out.println("10/0 doesn't equal " + 10/0.0);
    }
}
```

}

## 4. תוכנית לדוגמא

כתבו תוכנית ג'אווה המקבלת בשורת הפקודה שני מספרים. במידה והמספר הראשון מתחלק בשני ללא שארית, תודפס תוצאת החלוקה ביניהם. אחרת (כלומר, יש שארית), תודפס תוצאת החלוקה והשארית על פי הפורמט שבדוגמא. שם התוכנית הוא: Assignment1

במידה והמספר השני הוא 0, תודפס הודעה על כך שלא ניתן לבצע חלוקה ב 0.

דוגמאות לקלטים אפשריים ולפלטים הנדרשים:

12 3

יודפס הפלט הבא למסך:

The division result is: 4

עבור הקלט לתוכנית

12 5

יודפס הפלט הבא למסך:

The division result is: 2, remainder is: 2

עבור הקלט לתוכנית

12 0

יודפס הפלט הבא למסך:

Cannot perform zero division!

הדרכה:

- ראשית צרו פרויקט חדש באקליפס ובו מחלקה בשם Assignment1.
- 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).
- ניתן להניח שהתוכנית מקבלת שני ארגומנטים, ושניהם ניתנים להמרה ל int. במצגת התרגול ניתן למצוא דוגמא להמרה ממחרוזת למספר שלם.
- ניתן להעזר באופרטור % (מודולו) על מנת לקבל את שארית החלוקה של שני מספרים (למשל,  $x\%y$  יחזיר את שארית החלוקה של  $x$  ב  $y$ ).
- בפלט התוכנית יש רווחי בודד בין כל שתי מילים סמוכות, אין להשתמש בטאבים (t).
- בסוף הדפסת השורה אחרונה ניתן לרדת שורה, אך זו לא חובה.

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.

# בהצלחה !