# פיתוח תוכנה מונחה עצמים – תרגיל 2

**חלק ראשון**

המחלקה Turtle מגדירה עבורנו צב רגיל. יצירה של צב מציירת דמות של צב במרכזו של מסך גרפי. המסך הגרפי נוצר עם יצירתו של צב בפעם הראשונה. לצב יש יכולת תנועה. הוא יכול להסתובב סביב עצמו ימינה ושמאלה בכל מעלה שלמה ולפנות לכיוונים שונים. צב חדש נוצר כשהוא פונה כלפי מעלה. צב יכול לנוע קדימה אל הכיוון אליו הוא פונה לכל מרחק נתון.

לצב יש זנב. הזנב יכול להיות מורם או מורד. אם הזנב מורד והצב נע קדימה הצב משאיר לאורך מסלולו עקבות בצורה של קו על המסך הגרפי. אם הזנב מורם הצב לא משאיר עקבות. צב גם ניתן להסתרה וגילוי מחדש. בין אם הצב גלוי ובין אם הוא מוסתר הוא מבצע את כל הפעולות באותו האופן. כלומר, הוא יכול להסתובב, להתקדם קדימה, להשאיר עקבות כאשר זנבו מורד, או לא להשאיר סימן כאשר זנבו מורם.

לממשק הציבורי של המחלקה Turtle המאפיינים והיכולות הבאים:

Turtle (); // construct simple turtle

public long show (); // show yourself

public long hide (); // hide yourself

public void tailDown (); // lower the tail

public void tailUp (); // lift the tail

public void turnLeft (int degrees); // turn left in the given degrees

public void turnRight (int degrees); // turn right in the given degrees

public void moveForward (double distance); // advance forward in the given distance

שימו לב! לצב יש מספר שיטות נוספות הקשורות לאופן הפעולה שלו ושל הממשק הגרפי. התעלמו משיטות אלה, הן אינן נחוצות לעבודה ועלולות לפגום בתוכנית שלכם.

כדי להשתמש במחלקה Turtle יש להוריד מאתר הקורס אל המחשב שלכם את החבילה Turtle.jar, שמאגדת בתוכה מספר קבצים הנדרשים לעבודה עם הצב, יש להגדיר פרויקט ולהוסיף לו את החבילה ולייבא (import) את החבילה לקבצים כך שנוכל להשתמש בה בקבצים (ראו הוראות מפורטות לכך בסוף המסמך).

לכתיבת החלק הראשון עקבו אחרי המשימות הבאות.

משימה ראשונה – הכרות עם המחלקה Turtle

כתבו תוכנית (קובץ עם פונקציית main) בשם TwoTurtles.java אשר יוצרת שני צבים בשם Mario ו- Luigi. מריו יכתוב על המסך את האות M.

לויג׳י, **לאחר שמריו יסיים** לכתוב, יכתוב על המסך את האות L לצד האות M.

שימו לב, מטרת המשימה היא לאפשר לכם הכרות עם המחלקה Turtle, לכן אין גודל מסויים שנדרש לאותיות, ואין גודל מוגדר למרחק בין האותיות. בנוסף, סביר מאוד להניח שהאותיות לא יצאו ישרות. אין בעיה עם זה, כל עוד הן יזכירו בצורתן את האותיות (ראה שרטוט):

משימה שניה – הגדרה של צבים עם תכונות שונות ע"י הורשה

לא כל הצבים נוצרו שווים. יש חכמים, יש מוכשרים, ולא עלינו, מוזרים. הוסיפו מחלקות (משמע- תכנתו Classes) שיגדירו את הצבים הבאים:

* צב חכם (SmartTurtle) – צב חכם, מעבר להיותו צב רגיל לכל דבר ועניין, מבין גם משהו בגיאומטריה: הוא יודע לצייר מצולע משוכלל בעל מספר צלעות נתון באורך נתון. כתבו את המחלקה SmartTurtle, שימרו אותה בקובץ בשם SmartTurtle.java במחיצת העבודה, והוסיפו לה את השיטה הבאה:

public void draw (int sides, double size);

// draw a polygon in the given sides and size

שימו לב: כיוון שהזויות העוברת כפרמטר בשיטות turnLeft ו- turnRight היא זווית שלמה עלולה להתעורר בעיה במקרה של פוליגון משוכלל עם זווית לא שלמה. התעלמו מהבעיה. דאגו רק שהצב החכם יצייר נכון פוליגונים בעלי זווית שלמה.

* צב שיכור (DrunkTurtle) – צב שיכור הוא צב רגיל ששתה מעט וכתוצאה מכך קשה לו קצת ללכת. כשהוא מתבקש לנוע קדימה במרחק x הוא מבצע את הפעולות הבאות אחת אחרי השנייה לפי הסדר (הוא מבצע את כולן בכל פעם שהתבקש לנוע קדימה):
  + הוא מתקדם למרחק מקרי בין 0 ל- x.
  + בהסתברות של 30% הוא פונה x מעלות שמאלה.
  + הוא מתקדם מרחק מקרי נוסף בין 0 ל- x0.5 .

**ה-x הוא אותו ה-x בשלושת השלבים.**

בנוסף, בכל פעם שהוא מתבקש לבצע פנייה בזווית y הוא יפנה בזווית אקראית בין 0 ל- 1.5y.

כתבו את המחלקה DrunkTurtle ושימרו אותה בקובץ בשם DrunkTurtle.java במחיצת העבודה.

* צב מקרטע (JumpyTurtle) – צב מקרטע הוא צב חכם שמקרטע: כאשר הוא מצייר הוא מנתר לסירוגין מאושר. התוצאה היא שכאשר זנבו מורד הוא משאיר קו מקווקו, כלומר כאשר הורידו את זנבו למטה וביקשו ממנו ללכת התוצאה תהיה קו מקוטע ולא חלק. הקווקו יהיה בנוי כך שיופיע בכל פעם קו באורך 5 יח' ולאחר מכן 5 יח' מרחק ללא קו. כתבו את המחלקה JumpyTurtle ושימרו אותה בקובץ בשם JumpyTurtle.java במחיצת העבודה שלכם.

שימו לב: כיוון שצב מתקדם על פני סריג של נקודות, הצב לא יכול להתקדם תמיד למרחק הנדרש בדיוק נמרץ. התקדמות למרחק קצר עלולה ליצור אי דיוק גדול יחסית . יוצא אפוא שאם ההתקדמות של צב מקרטע תשבר לרצף ארוך של פסיעות ודילוגם קטנים, אי הדיוק יצטבר והצב עלול להתקדם בפועל למרחק שונה מהנדרש. התעלמו מהבעיה. שיברו את המרחק אליו מתבקש הצב המקרטע להתקדם למספר פסיעות וניתורים קטן, והניחו תמיד שהצב יידרש להתקדם למרחקים גדולים.

* צב שמונה (EightTurtle) - צב שמונה הוא כמו צב חכם, אך יש לו חולשה למספר 8. אם נבקש ממנו לצייר מצולע:
  + בהסתברות של 70% הוא יצייר מצולע עם אורך צלע שקבעתם, אך בעל 8 צלעות (ללא קשר למספר הצלעות שקבעתם).
  + בהסתברות של 30% הוא יצייר מצולע עם מספר הצלעות שקבעתם, אך עם אורך צלע של 18 (ללא קשר לאורך הצלע שקבעתם).

כתבו את המחלקה EightTurtle ושימרו אותה בקובץ בשם EightTurtle.java במחיצת העבודה.

* צב סטלן (SatlanTurtle)- צב סטלן הוא כמו צב שיכור, אך לא תמיד יהיה לו כוח לבצע את הפעולות שהוא נדרש לבצע. בכל פעם שמבקשים ממנו משהו:
  + בהסתברות של 30% הוא יבצע את הפעולה כמו צב שיכור.
  + בהסתברות של 40% הוא יבצע את הפעולה כמו צב פשוט.
  + בהסתברות של 30% הוא לא יבצע כלום.

כתבו את המחלקה SatlanTurtle ושימרו אותה בקובץ בשם SatlanTurtle.java במחיצת העבודה. שנו בה את הדרוש שינוי.

**חשוב ביותר: על כל הצבים מכל הסוגים לא להשאיר מאחוריהם עקבות כאשר הזנב שלהם מורם.**

משימה שלישית – ניהול צבא של צבים באמצעות פולימורפיזם

כתבו תוכנית בשם Army.java שתנהל צבא (מערך) של 5 צבים.

* בשלב ראשון אפשרו למשתמש לבחור את צבא הצבים כרצונו. הציגו לפניו את התפריט שלמטה וקבלו את בחירתו עבור כל אחד מחמשת הצבים בנפרד. אפשרו לו לבחור כל תערובת של צבים. להלן התפריט:

Choose the type of a turtle:

1. Simple
2. Smart
3. Drunk
4. Jumpy
5. Eight
6. Satlan

* בשלב שני צרו את הצבים הנדרשים וקדמו אותם שלב אחר שלב על פני השלבים הבאים:
  1. הסתדרות בשורה (מרחק בין צב לצב 120, כאשר לא נשארים סימנים בהליכה)
  2. הורדת זנב
  3. צעידה קדימה למרחק של 65
  4. פניה של 40 מעלות שמאלה
  5. צעידה קדימה למרחק של 75
  6. כל מי שיודע לצייר צריך לצייר משושה בעל צלע בגודל 40
  7. העלמות הצבים

בסוף הריצה נשארים ציורי הצבים בלבד, ללא הצבים.

שימו לב:

* על הצבים להתקדם כולם יחד. אף צב לא יכול לעבור לשלב גבוה יותר בטרם גמרו כל הצבים האחרים את השלב הקודם. השלבים יחלו באופן אוטומטי לאחר שהמשתמש בחר את הצבים. כאשר הצבים פועלים על פי הסדר שבו הם נבחרו.
* **כל הצבים הולכים בוודאות וישר למיקום הראשוני ורק לאחר ארגונם מתחילים להתנהג לפי אופיים.**

הערות לחלק הראשון:

* יתכן כי הצב השיכור צב סטלן וצב שמונה "ידרסו" חלק משרטוטי הצבים האחרים –זה בסדר.
* יתכן ותדרשו להגדיל את המסך הנוצר ע"מ לראות את הציורים – זה בסדר.
* הצבים יהיו ממוקמים משמאל לימין לפי סדר בחירתם.

# חלק שני

בתרגיל הראשון התבקשתם לממש את המשחק שולה מוקשים. בתרגיל זה תצטרכו לממש את המשחק באמצעות אובייקטים שונים ולהשתמש במימוש המשחק שבניתם בתרגיל הראשון, בצורה מודולרית.

המשחק מכיל לוח בגודל 9X9 של משבצות מסוגים שונים שניתן להפעילן ע"י לחיצה עליהן. יהיו משבצות שהן משבצות ריקות, משבצות של מספרים ומשבצות פעילות. המשבצות הפעילות תופסות את מקומם המוקשים. במקום מוקשים רגילים יהיה ניתן להיתקל בסוגים שונים של מוקשים או עזרה ראשונה. למוקשים קיימת התנהגות מיוחדת של פסילה. חוקי המשחק דומים למשחק שמימשתם בתרגיל הראשון. על הלוח יהיו 10 משבצות פעילות (במקום מוקשים, אופן הגרלת המיקומים מפורט מטה) ועל השחקן לפצח את מיקום המשבצות הפעילות על הלוח.

כאשר השחקן בוחר לחשוף משבצת כלשהי, המשבצת נחשפת. אם חושפים משבצת ריקה שאין עליה מספר או מוקש, כל המשבצות הריקות (לא משבצות פעילות) בסביבתה נחשפות גם כן, עד שנוצר שטח של משבצות ריקות, המוקפות במשבצות ממוספרות. אם חושפים משבצת פעילה, מתבצעת פעולה כלשהי כפי שיפורט בהמשך.

לכל משבצת יש לממש שיטה toString() אשר מציגה את תוכן המשבצת: 0 עבור משבצת ריקה, עבור משבצת של מספר המספר שלה, ועבור משבצת פעילה אות המייצגת אותה (לפי מפתח המחרוזות).

#### סוגי משבצות

* משבצת ריקה – הפעלתה גורמת לפתיחת המשבצות הריקות השכנות לה.
* משבצת מספר – נחשף המספר שעל המשבצת.
* משבצת מוקש קטלני – מסיימת את המשחק.
* משבצת מוקש פוצע – פוצעת את השחקן אם אינו פצוע או פוסלת (מסיימת את המשחק) אם הוא כבר פצוע.
* משבצת עזרה ראשונה – מבריאה את השחקן אם הוא פצוע אחרת לא קורה דבר.

מפתח מחרוזות:

* משבצת ריקה: E
* משבצת מספר: N
* משבצת מוקש קטלני: X
* משבצת מוקש פוצע: IN
* משבצת עזרה ראשונה: FA

#### שלב 1

כתבו את המחלקות השונות של המשבצות כאשר בכל סוג משבצת קיימת שיטה שמפעילה את המשבצת.

#### שלב 2

בשלב זה הפכו את הקוד שלכם מתרגיל 1 למודולארי ע"י שימוש בפולימורפיזם. כלומר צרו לוח מסוג משבצת ריקה, מקמו את המשבצות הפעילות לפי אלגוריתם "מיקום משבצות פעילות" וממשו את המשחק עם החוקים החדשים.

שימו לב! בעבודה זו עליכם להציג את כל הלוח החשוף בסוף המשחק.

#### אלגוריתם "מיקום משבצות פעילות"

בחירת מיקום משבצת פעילה מתבצע לפי האלגוריתם של תרגיל 1. במידה ומשבצת מסוימת נבחרה להיות פעילה מתבצעת בחירה של סוג המשבצת הפעילה כאשר הסיכוי לבחור כל סוג הוא וישנם מקסימום 4 משבצות פעילות מאותו הסוג.

## הערות לשני החלקים

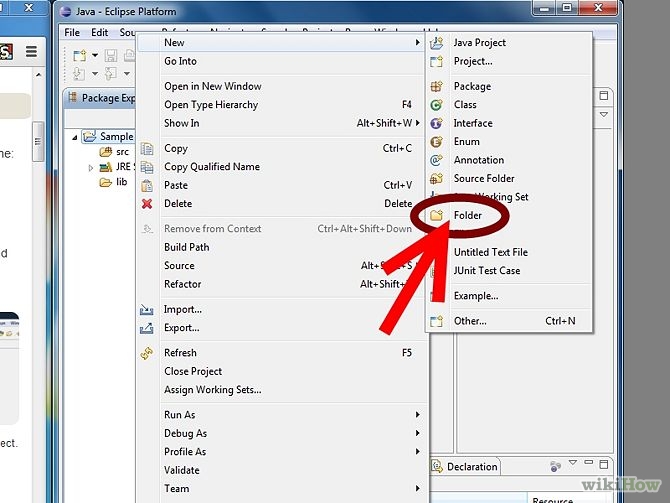
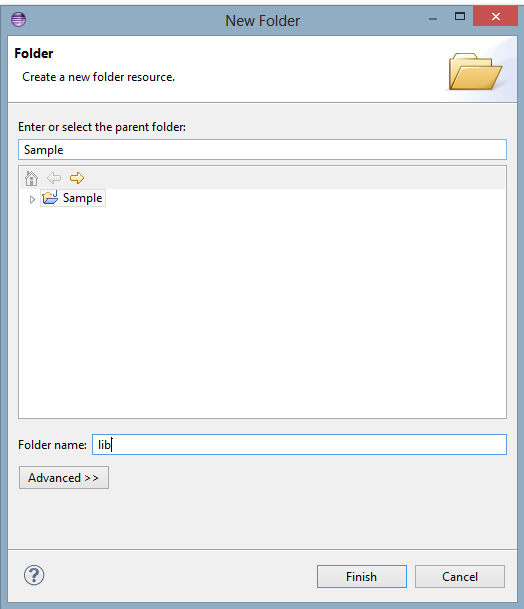
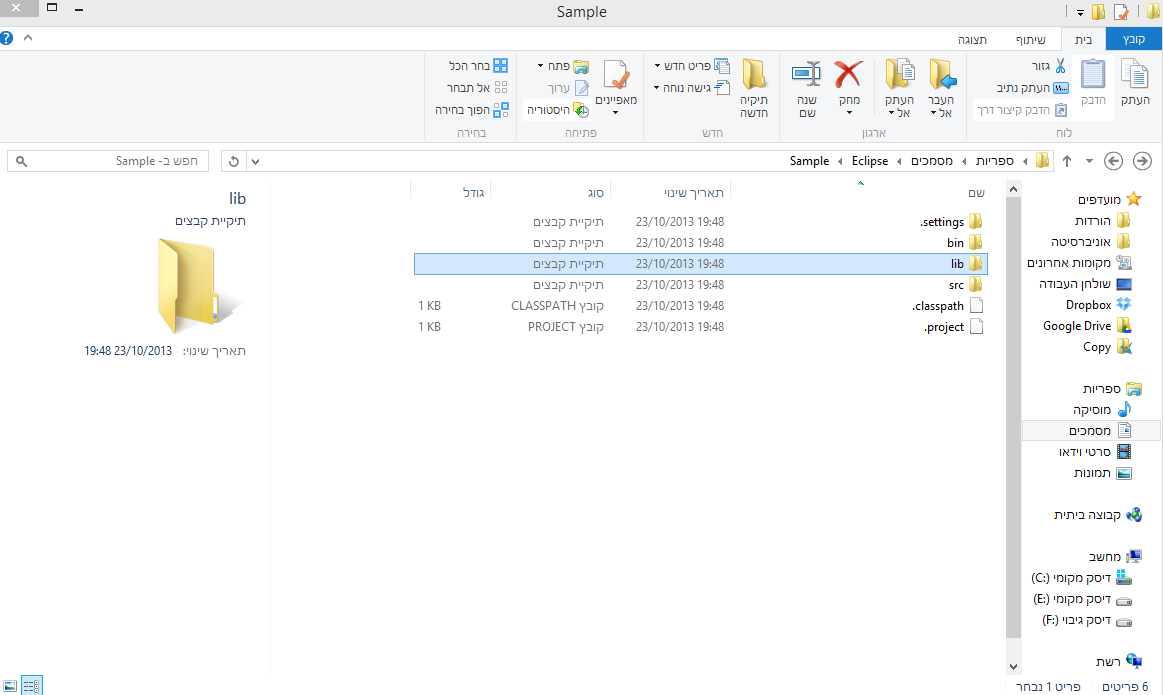
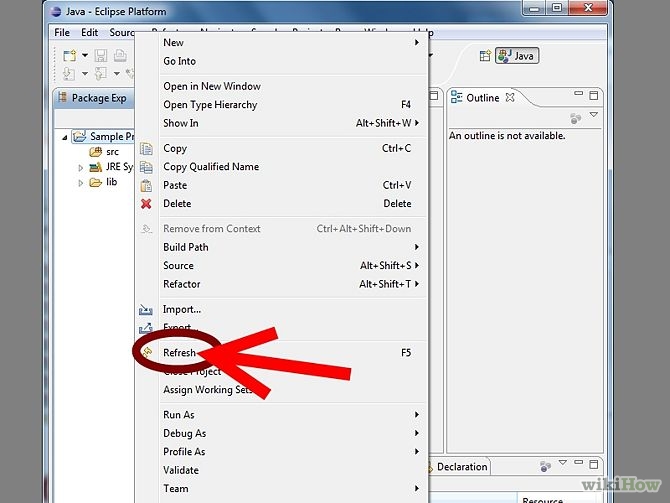
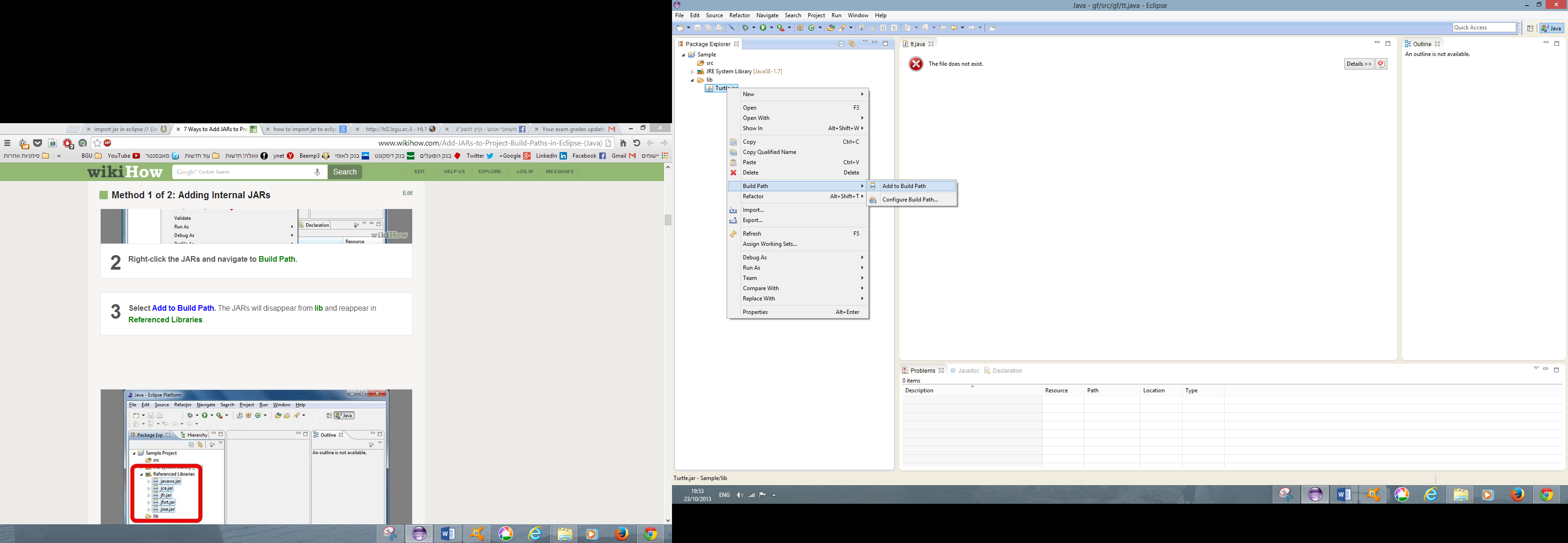
* הדגש בעבודה זו הוא על שימוש נכון בהורשה ופולימורפיזם
* מותר להוסיף מחלקות נוספות כראות עיניכם, בשני החלקים, גם אם לא נרשמו במפורש בעבודה. עם זאת, עליכם לתכנת באותם שמות את כל המחלקות המופיעות בהוראות
* יש להשתמש בעקרונות וכלים של פיתוח תוכנה מונחה עצמים שנלמדו בכיתה. עבודות בהן לא יישומו העקרונות יקבלו ציון נמוך גם אם הפלט יהיה נכון
* הריצו את התוכניות ובדקו אותן. העבודה כוללת סה"כ 3 תכניות (קבצים עם main) - משימת ההיכרות עם הצבים, צבא הצבים, ומשחק שולה המוקשים המשודרג.
* שימו לב, אל תשתמשו ב- casting אלא אם כן הוא הכרחי. יורדו נקודות על שימוש מיותר.
* אין צורך בבדיקת קלט יש להניח כי הקלט תקין.
* הקפידו על כתיבה קריאה וברורה -שמות ברורים למשתנים, אינדנטציות וכו'
* יש להקפיד על רישום הערות, בפרט בתחילת כל פונקציה. חוסר בהערות יגרור הורדת נקודות
* מתרגל אחראי – **מאיה לביא**
* שאלות על התרגיל דרך הפורום בלבד
* שימו לב – התשובות לשאלות הנשאלות בפורום מהוות הבהרות לדברים שלא הובנו בהנחיות, ועל כן מחייבות את כלל הסטודנטים לכל דבר ועניין. אך ורק במידה ויהיה צורך בשינוי נוסח ההנחיות ייצא מייל בנושא. אין לשאול בפורום שאלות הקשורות לאופן מימוש העבודה

## הוראות הגשה

* הגשה בזוגות עד לתאריך 30.11.20 בשעה 23:55
* עליכם להגיש מסמך תכנון למחלקות (בקובץ וורד) עבור החלק השני בלבד – "שולה המוקשים". המסמך יכלול תרשים המתאר את קשרי ההורשה בין המחלקות, שיטות חשובות הנמצאות בכל מחלקה, ואילו מהן נדרסות אצל היורשים
* יש להגיש תיקייה אשר שמה יהיה מספרי תעודת הזהות. לדוגמא: 111111111\_222222222. תיקייה זו תהיה מכווצת (ZIP), ותכלול בתוכה שתי תיקיות – אחת לפרויקט הצבים ותיקייה נוספת הכוללת את פרויקט "שולה המוקשים". שימו לב- צריך לצרף את כל התוכן של תיקיות הפרויקט כפי שהוא נשמר בג'אווה, ולא רק קבצים בודדים
* אם עשיתם שינוי כלשהו בתוכנית לאחר שהגשתם יש להגיש תיקייה חדשה בתוספת : \_2ת.ז , 3\_ת.ז... רק התיקייה האחרונה תיבדק. (לדוג' – 111111111\_222222222\_2 )
* ההגשה היא לאתר הקורס במודל בלבד, ולא לאימייל. הגשות לאימייל לא יקבלו מענה
* דחיות להגשת העבודה יינתנו ע"י אחראי הקורס ד"ר רועי זיוון בלבד
* כל יום איחור גורר הורדה של 5 נקודות

עבודה מהנה!

How To Add the Turtle Package:

1. Download Turtle.zip from the course site via Moodle.
2. Extract Turtle.jar from Turtle.zip.
3. Start Eclipse and create a project
4. Right click on the project -> New -> Folder
5. Name the new folder "lib".
6. From Windows Explorer, copy Turtle.jar to the new library (will be found under %project path%\lib\)
7. In Eclipse, right click on the project -> Refresh
8. Under the project, inside the lib library, the Turtle.jar will be found. Right click -> Build Path -> Add to Build Path

Reminder : Don’t forget to import the package to your files that use it by writing :

"import Turtle.\*;"

As the first line in your file.

Good luck

The course staff