

## פיתוח תוכנה מונחה עצמים – תרגיל 2

**הגשה עד 17.08.2021 בשעה 23:55**

### חלק א- צבים

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 מורס או מורד. אם הזנב מורד והצב נע קדימה הצב משאיר לאורך מסלולו עקבות בצורה של קו על המסך הגרפי. אם הזנב מורס 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 מורס.

לממשק הציבורי של המחלקה Turtle המאפיינים והיכולות הבאים:

```
Turtle (); // construct a turtle  
show (); // show yourself  
hide (); // hide yourself  
tailDown (); // lower the tail  
tailUp (); // lift the tail  
turnLeft (int degrees); // turn left in the given degrees  
turnRight (int degrees); // turn right in the given degrees  
moveForward (double distance); // advance forward in the given distance
```

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 בשם TwoTurtles.java אשר יוצרת שני צבים בשם ארטיום ולינוי. הצבים מאד אוהבים את נבחרת האולימפיאדה הישראלית בה מתמודדים ארטיום דולגופיאט ולינוי אשרם. מכיוון שארטיום זכה ראשון במדליה הוא הצב הראשון, ולכן יכתוב את האות הגדולה באנגלית A ולינוי אחריו תכתוב את האות L.

שימו לב, מטרת המשימה היא לאפשר לכם היכרות עם המחלקה Turtle, לכן אין גודל מסויים שנדרש לאותיות. 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 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 שהזוויות 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. כמו כן, אין צורך שהפוליגון יהיה עם אותה אוריינטציה של האופק.

- צב שיכור (DrunkTurtle) – צב שיכור הוא צב רגיל ששתה מעט וכתוצאה מכך קשה לו קצת ללכת. כשהוא מתבקש לנוע קדימה למרחק x הוא מבצע את הפעולות הבאות:

- הוא מתקדם קדימה למרחק מקרי בין 0 ל-  $x$ .
- בהסתברות של 30% הוא פונה  $x$  מעלות שמאלה.
- הוא מתקדם מרחק מקרי נוסף בין 0 ל-  $x$ .

בנוסף, בכל פעם שהוא מתבקש לבצע פנייה בזווית  $y$  הוא יפנה לכיוון שהוגדר לו בזווית אקראית בין 0 ל-  $2y$ .

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

- צב מקרטע (JumpyTurtle) – צב מקרטע הוא צב חכם שמקרטע: הוא מנתר לסירוגין מאושר. התוצאה היא שכאשר זנבו מורד הוא משאיר קו מקווקו, כלומר כאשר הורידו את זנבו למטה ומתקדם במרחב, 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. שנו בה את הדרוש שינוי.

**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 ודילוגים 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. 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

- בשלב שני צרו את הצבים הנדרשים וקדמו אותם שלב אחר שלב על פני השלבים הבאים:

1. הסתדרות בשורה ופנייה לחזית – 90 מעלות שמאלה.  
(מרחק בין צב לצב 120, כאשר לא נשארים סימנים בהליכה- כולם מתקדמים כמו צב רגיל- רק בשלב זה)
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 (להזכירכם- צב שמונה יודע לצייר, ולכן, אם הוא יצייר צורה אחרת או גודל אחר מהדרוש זה תקין).
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.
- צב שיכור מתמקם כמו כל הצבים, כלומר הוא הולך ישר למיקום ההתחלתי.

## הערות לחלק הראשון:

- ♦ יתכן כי הצב השיכור 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.

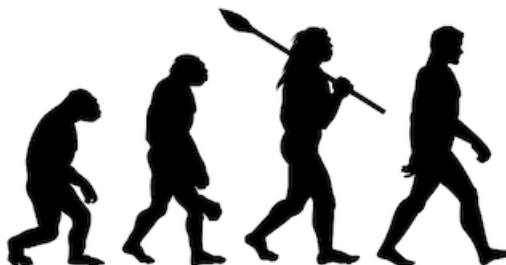
## חלק שני- 2048 אבולוציה

בתרגיל בית 1 התבקשתם לממש את המשחק 2048. בתרגיל זה אתם מתבקשים לממש את המשחק, כאשר האריחים המופיעים על הלוח מיוצגים על ידי אובייקטים השונים זה מזה, כפי שתראו בהמשך. יש לציין כי חוקי המשחק משתנים בתצורת איחוד האריחים ואופן חישוב התוצאה.

### סוגי האריחים במשחק

ישנם 4 סוגים שונים של אריחים: קוף, שימפנזה, אדם קדמון ואדם. להלן שמות סוגי האריחים באנגלית על פי סדר ההתפתחות שלהם עם המשחק (משמאל לימין- כלומר חיה ראשונה שמופיעה הינה הקוף):

Monkey, Chimpanzee, Flintstone, Human



בשונה מהמשחק המקורי, איחוד האריחים מתבצע על בסיס סוג האריח וגילו בשנים (מספר שלם). טווח הגילאים האפשרי משתנה בין משחק למשחק. בתחילת המשחק יוגרל מספר אקראי בין 5 ל 9, אשר יגדיר את ערך הגיל המקסימלי של סוגי האריחים השונים גיל זה זהה לכלל הסוגים (ראו תחילת משחק). כאשר שני אריחים מתאחדים (לאור סוגם וגילם הזהה) גילו של האריח המאוחד יגדל. אופן ההתקדמות בגיל האריח משתנה לפי סוג האריח המאוחד (ראה חוקיות בהמשך). כאשר נאחד שני אריחים אשר נמצאים בגילם המקסימלי, ייווצר אריח חדש מהסוג הבא באבולוציה (קוף) ← שימפנזה ← אדם קדמון ← אדם) שגילו יהיה 1 עם ייווצרו. כאשר הבן אדם מגיע לגילו המקסימלי התוכנית תגיב באופן דומה למשחק המקורי כאשר אחד האריחים הגיע לראשונה ל 2048 (תופיע הודעה למשתמש: (You won! Human is at max age! Would you like to continue? Press y if yes, n if not

כל אריח על הלוח יהיה מיוצג על בסיס סוגו וגילו. לכל סוג אריח יש אות אשר מייצגת אותו :

• **Monkey = M**

• **Chimpanzee = C**

• **Flinston = F**

• **Human = H**

לדוגמא : בן אדם בגיל 4 יופיע על הלוח כך : H4. במצב בו אין שיבוץ של אריח יופיע התו '#':

### אופן התקדמות האריחים בגיל

- **Monkey** – כאשר שני אריחים של קופים בעלי גיל זהה יתאחדו, גיל הקוף החדש שיווצר במיקום האיחוד יהיה גילם הקודם של הקופים בתוספת יחידה. לדוגמא : כאשר נאחד שני קופים בגיל 2 גיל הקוף שנוצר יהיה 3.
- **Chimpanzee** – כאשר שני אריחים של שימפנזות בעלי גיל זהה יתאחדו, גיל השימפנזה החדש שיווצר במיקום האיחוד יהיה גילם הקודם של השימפנזות בתוספת 2 יחידות. שימו לב תוספת 2 היחידות אינה יכולה לאפשר לעבור את הגיל המקסימלי שהוגדר עבור המשחק – בעת הגעה לגיל המקסימלי יוצר אריח חדש מסוג **Flintstone**.  
לדוגמא עבור גיל מקסימלי 6 ו-2 קופים עם גיל 5 (כלומר איחוד של  $C5 + C5 = F1$ )
- **Flintstone** – כאשר שני אריחים של איש קדמון בעלי גיל זהה יתאחדו, גיל האיש הקדמון החדש שיווצר במיקום האיחוד יהיה גילם הקודם של שני האריחים בתוספת יחידה ( $F2 + F2 = F3$ ). בנוסף לכך, גילם של כל הקופים על הלוח יתקדם באחד גם כן. באם יוצר איחוד של מספר אריחי איש קדמון בתור יחיד, גילם של כלל הקופים בלוח יגדל כמספר האיחודים בתור זה (שימו לב לא לעבור את הגיל המקסימלי גם כאן).
- **Human** – כאשר שני אריחים של בני אדם בעלי גיל זהה יתאחדו, גיל הבן אדם החדש שיווצר במיקום האיחוד יהיה גילם הקודם של בני האדם בתוספת יחידה. לדוגמא : כאשר נאחד שני בני אדם בגיל 2 גיל הבן אדם שנוצר יהיה 3. בנוסף לכל אדם יש סכום כסף. בתהליך איחוד האריחים של בני האדם, גם סכום הכסף שברשותם נכלל באיחוד. באיחוד של שני בני אדם סכום הכסף של האדם החדש הינו החיבור של סכומי הכסף של שני בני האדם שהתאחדו. לדוגמא כאשר נאחד בן אדם בגיל 3 שברשותו 8 יחידות כסף עם אדם נוסף בגיל 3 שברשותו 10 יחידות כסף גם כן, האדם לאחר האיחוד יהיה בגיל 4 שברשותו 18 יחידות של כסף. כאשר אדם נוצר לראשונה (לאחר המעבר מאיש קדמון) סכום הכסף שברשותו הוא 2.

### **חידודים :**

- אף אריח אינו יכול לעבור את גילו המקסימלי
- כאשר נוצר אריח מסוג חדש בעקבות אבולוציה גיל האריח תמיד יותחל ל 15 ויסתיים התור,
- ללא קשר למספר השנים שהאיחוד מוסיף לגיל.
- כל שינוי של אריח משפיע על הניקוד עם התוספת הרלוונטית לשינוי

### אופן חישוב הניקוד

בכל פעם שנוצר אריח חדש או גיל האריח משתנה יתווסף ניקוד לתוצאה הסופית. לכל סוג אריח קיים ערך פקטור, המיוצג על ידי מספר שלם. ערכי הפקטור הינם דיפולטיבים על פי המצוין בהמשך. תוספת הניקוד תחושב על פי תבנית של גיל האריח כפול ערך הפקטור.

• **Monkey default factor = 1**

• **Chimpanzee default factor = 2**

• **Flinston default factor = 3**

• **Human default factor = 4**

### תחילת המשחק:

בתחילת המשחק תוצג למשתמש ההודעה: "Welcome, would you like to start the game?" אם תשובת המשתמש היא n התוכנית תסתיים. אם תשובת המשתמש היא y המשחק מתחיל בהודעה:

"The max age is XX" כאשר XX הינו הגיל המקסימלי שהוגרל עבור העצמים השונים.

לאחר מכן, יוצג על המסך לוח בגודל  $4 \times 4$  ובמיקום כל אריח יופיע התו #. התו # מייצג את המשבצות הריקות. בשלב שני יודפסו על הלוח בשני מיקומים אקראיים קופים בגילאים 1 או 2. משמעות המיקום האקראי בא לידי ביטוי בכך שקיימת הסתברות שווה לקוף להופיע באחת המשבצות. בתחילת המשחק כל המשבצות ריקות לכן קיים סיכוי של  $1/16$  שהקוף יופיע בכל אחת מהמשבצות. בעבור הקוף השני שיופיע, קיימת הסתברות של  $1/15$  מכיוון שמשבצת אחת נתפסה על ידי הקוף הראשון. באשר לגילם של הקופים קיים סיכוי של 0.7 שגיל הקוף יהיה 1 וסיכוי של 0.3 לגיל 2.

### מהלך המשחק:

בכל תור יודפסו הפריטים המוצגים:

- תוצאת המשחק הנוכחי (score) - ראו "אופן חישוב הניקוד"
- תוצאת שיא (best) - בסיום המשחק המשתמש יכול לבחור להתחיל משחק חדש ולכן ירצה לדעת את שיא הניקוד אותו קיבל עד כה.
- סכום הכסף של האריחים על הלוח (money) - תזכורת, לבן אדם יש גם סכום כסף שבידו
- לוח המשחק הנוכחי.
- מקשים אפשריים (המקשים בהם המשתמש ייעזר בכדי לשחק את המשחק). ניתן להניח קלט תקין ואותיות קטנות. (המקשים האפשריים דומים לאילו שבעבודה מספר 1)

המקשים הם: w = up, s = down, a = left, d = right, r = return, e = exit, n = new game

בכל פעם שהמשתמש יזיז את האריחים יתווסף אריח חדש של קוף בגיל 1 או 2 במיקום אקראי על פי החוקים של המשחק המקורי, בדומה להוספת הספרות 2 או 4.

### סיום המשחק-

- בהקבלה למשחק המקורי, שבו מופיעה הודעה למשתמש שניצח במשחק כאשר הגיע לאריח של 2048, משתמש אשר הצליח להגיע לבן אדם בגיל המקסימלי, משמעות הדבר שניצח במשחק. תוצג ההודעה:

You won! Old Human! would you like to continue? Press y if yes, n if not

- עבור בחית האופציה y: המשחק ימשיך **וגיל בן האדם ימשיך לגדול** עד לפסילה או יציאה מהתוכנית (כלומר לבן אדם אין הגבלה של גיל מקסימלי)
- עבור בחירת האופציה n: המשתמש יישאל אם הוא רוצה להתחיל משחק חדש או לסיים את התכנית:

For a new game press y, n if not

במידה ונבחר משחק חדש – יוגרל גיל מקסימלי חדש.

- יציאה על ידי הלחצן e
- המשתמש נפסל כאשר לשחקן אין יותר מהלכים אפשריים. כלומר, שאין יותר אריחים ריקים עם הערך אפס ואין אריחים זהים הצמודים זה לזה. עבור מצב זה תופיע ההודעה:

.Game over your score is xxxxxx. For a new game press y, n if not  
o xxxxxx הינה התוצאה במשחק הנוכחי. כעת המשתמש יכול להתחיל משחק חדש או  
לסיים את התוכנית.

### הערות לשני החלקים

- עבור חלק 2 אבולוציה – לחיצה פעמיים על r, בחירת כיוון בלתי אפשרי ... ועוד מקרים כמו אלו שלא פורטו בעבודה זו – יש להתייחס לאותם חוקיות כפי שפורטו בעבודה 1.
- מקרי קצה שלא פורטו – יש לטפל כרעות עיניכם כל עוד לא סתרתם אף הנחיה של העבודה.
- 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 ה"2048 אבולוציה"
- שימו לב, אל תשתמשו ב-casting ו-InstanceOf אלא אם כן הוא הכרחי. יורדו נקודות על שימוש מיותר
- הקפידו על כתיבה קריאה וברורה - שמות ברורים למשתנים, אינדנטציות וכו'
- יש להקפיד על רישום הערות, בפרט בתחילת כל פונקציה. חוסר בהערות יגרור הורדת נקודות
- מתרגלת אחראית – נעם גאון
- שאלות על התרגיל דרך הפורום בלבד – ואך ורק על הנחיות העבודה ולא על דרכי מימוש

### הוראות הגשה

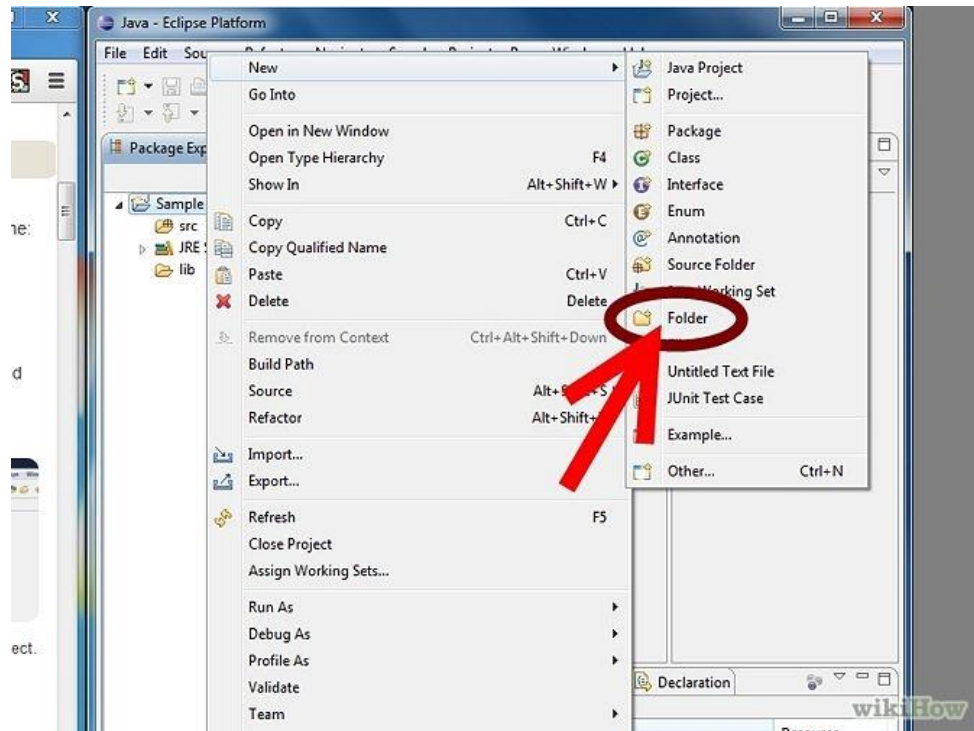
- הגשה בזוגות עד לתאריך 17.08.2021 בשעה 23:55
- עליכם להגיש מסמך תכנון למחלקות (בקובץ וורד/פאורפוינט) עבור החלק השני בלבד – "2048 אבולוציה". המסמך יכלול תרשים המתאר את קשרי ההורשה בין המחלקות, שיטות חשובות הנמצאות בכל מחלקה, ואילו מהן נדרסות אצל היורשים
- יש להגיש תיקייה אשר שמה יהיה מספרי תעודת הזהות. לדוגמא: 22222222\_11111111.
- תיקייה זו תהיה מכווצת (ZIP), ותכלול בתוכה שתי תיקיות – אחת לפרויקט הצבים ותיקייה נוספת הכוללת את פרויקט "evoultion 2048". שימו לב- צריך לצרף את כל התוכן של תיקיות הפרויקט כפי שהוא נשמר בג'אווה, ולא רק קבצים בודדים
- שאלות על התרגיל דרך הפורום בלבד, שאלות שישאלו במייל לא יזכו למענה
- 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
- ההגשה היא לאתר הקורס במודל בלבד, ולא לאימייל. הגשות לאימייל לא יקבלו מענה
- זחיות להגשת העבודה יינתנו ע"י אחראי הקורס רועי בלבד
- כל יום איחור גורר הורדה של 5 נקודות



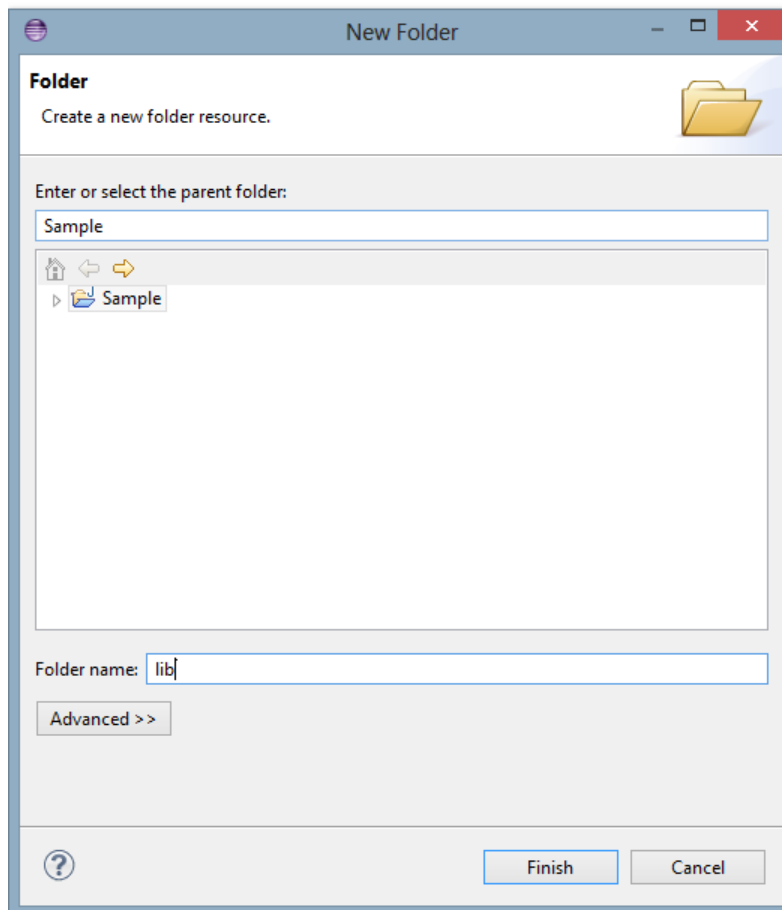
בהצלחה!

## How To Add the Turtle Package:

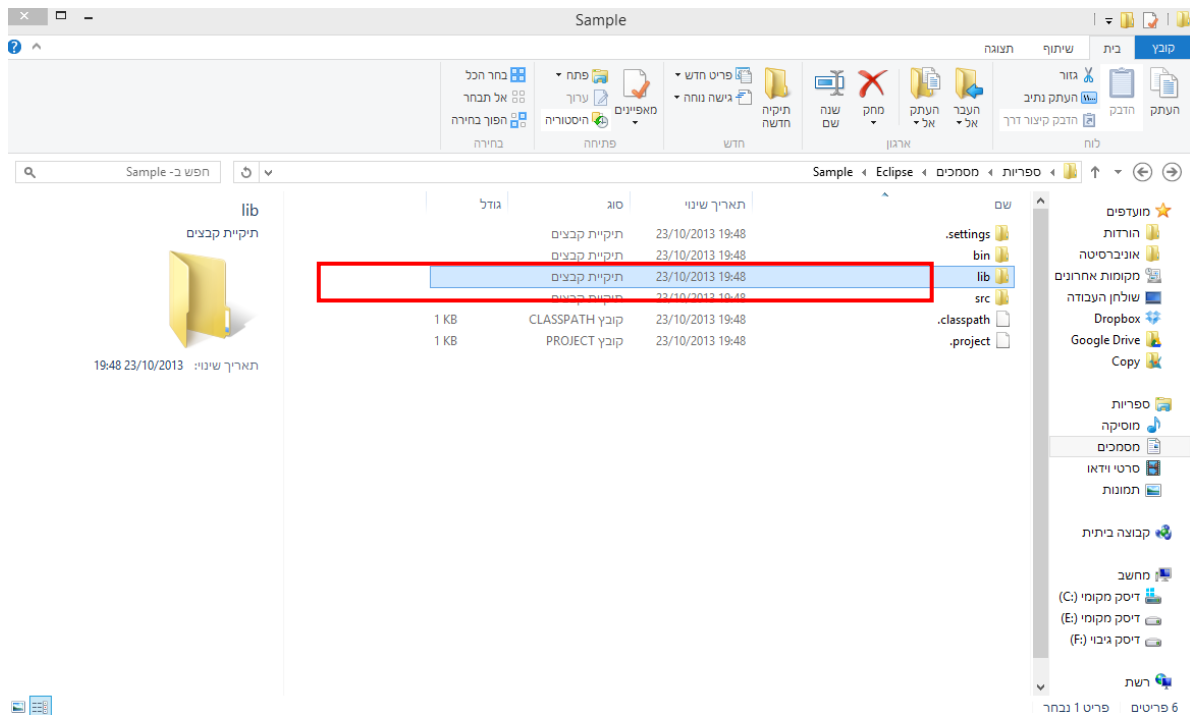
1. Download Turtle.zip.
2. Extract Turtle.jar from Turtle.zip.
3. Start Eclipse and create a project
4. Right click on the project -> *New*-> *Folder*



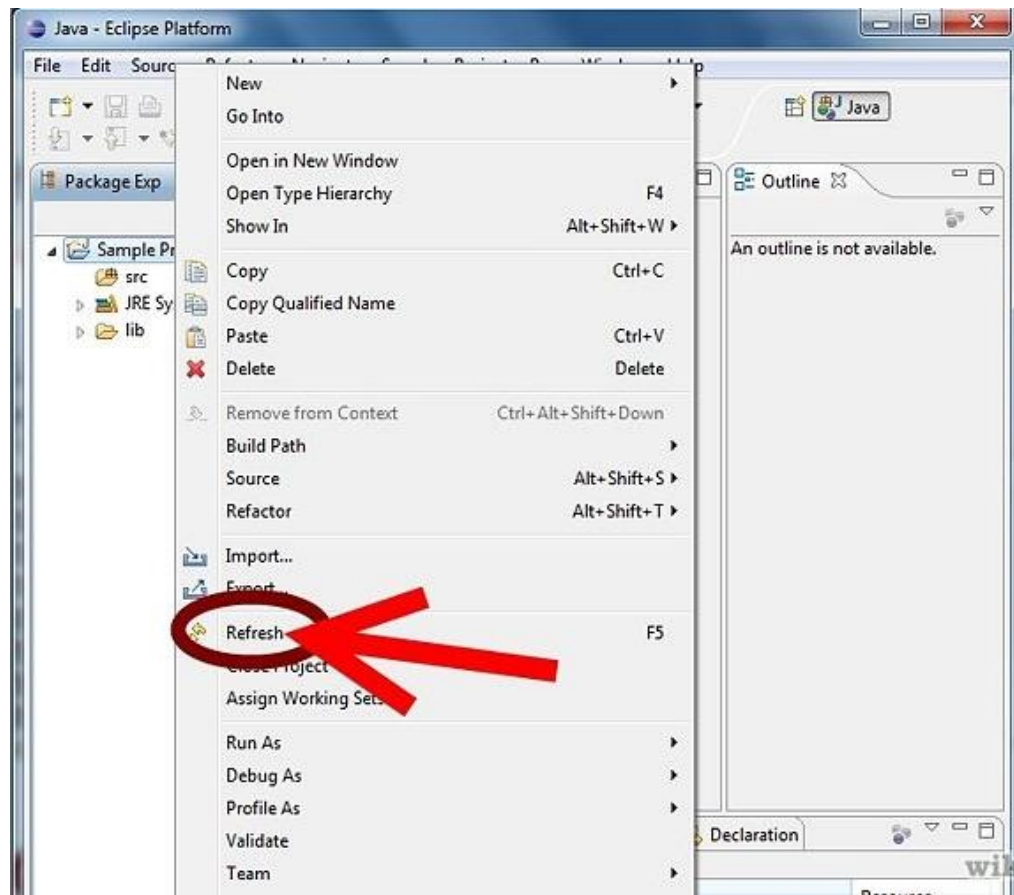
5. Name the new folder "*lib*".



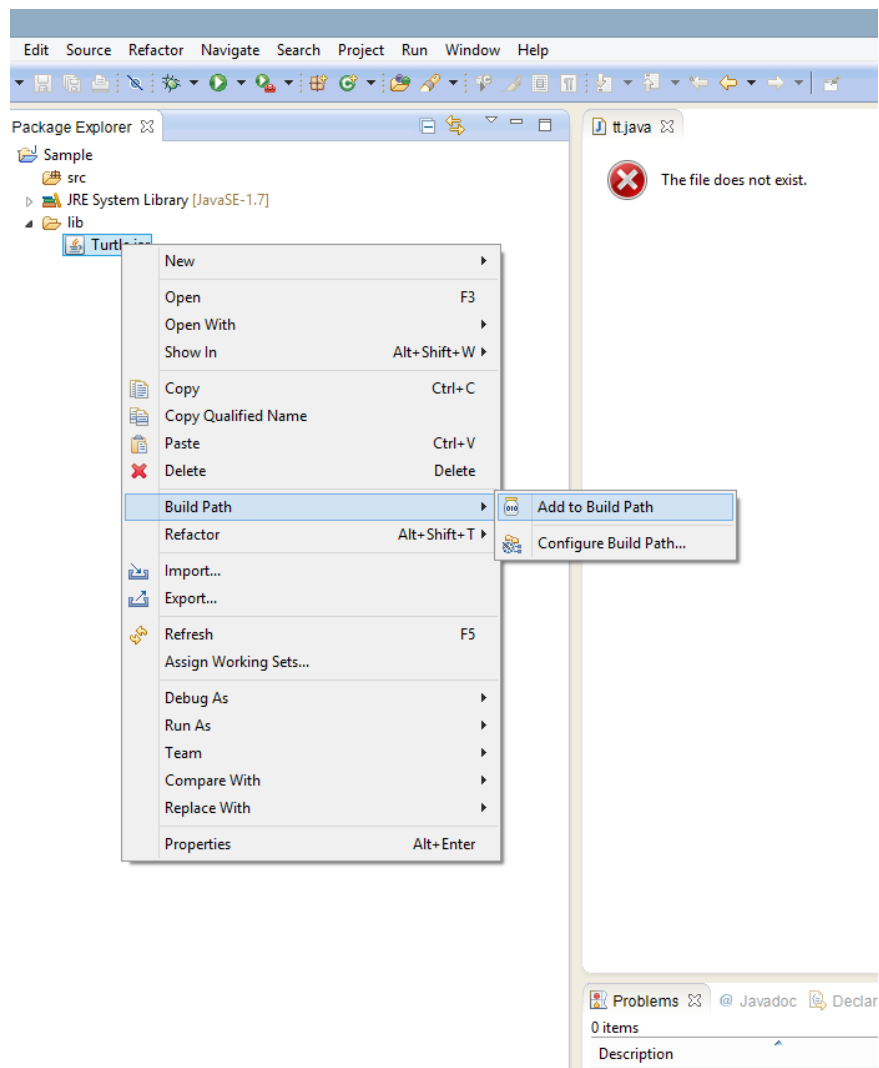
6. From Windows Explorer, copy Turtle.jar to the new library (will be found under %project path%\lib)



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