קורס פיתוח תכנה מונחה עצמים

סמסטר א'

# תרגיל בית מס' 2- הגשה ב-29.11.21 23:59

**מטרת התרגיל :** שימוש נכון במבנה ירושה ופולימורפיזם

**דגש:** שמות המחלקות, שמות השיטות והחתימות שלהן שמופיעות בהנחיות **חייבות** להיות **זהות** בקוד!

**חלק א'**

המחלקהTurtle מגדירה עבורנו צב רגיל. ביצירה של צב מופיעה 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 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 given distance

public long fillColor(String fillColor) //draw the turtle. input is “Blue”, “Red” etc.

**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.

1. **משימה ראשונה – הכרות עם המחלקה Turtle**

כתבו תוכנית בשם TwoTurtles.java אשר יוצרת שני צבים בשם Romeo ו- Juliet אשר מציירים יחד לב. Romeo מצייר את צד ימין ו- Juliet מציירת את צד שמאל. שימו לב, מטרת המשימה היא לאפשר לכם הכרות עם המחלקה **Turtle**, לכן אין גודל מסויים שנדרש ללב. בנוסף, סביר מאוד להניח שהלב יצא ייצא עקום. אין בעיה עם זה, כל עוד הוא יהיה סימטרי ויזכיר בצורה את השרטוט הבא:

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

הוסיפו את המחלקות הבאות – **יש להקפיד על יצירת המחלקות עם השמות האלו במדוייק!**

* **צב חכם (SmartTurtle) –** צב חכם, מעבר להיותו צב רגיל לכל דבר ועניין, מבין גם משהו בגיאומטריה: הוא יודע לצייר מצולע משוכלל, פוליגון, בעל מספר צלעות נתון באורך נתון (כל הצלעות שוות). 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 למרחק x הוא מבצע את הפעולות הבאות **(לפי הסדר, אין תלות בין הפעולות):**
  1. הוא מתקדם למרחק מקרי בין 0 ל- x.
  2. בהסתברות של 25% הוא פונה 50 מעלות שמאלה, בהסתברות המשלימה הוא לא יפנה וימשיך באלגוריתם.
  3. בהסתברות של 50% הוא ילך אחורה מרחק מקרי בין 0 ל-x, **אחרת** הוא מתקדם מרחק מקרי נוסף בין 0 ל- x.

**בנוסף,** בכל פעם שהוא מתבקש לבצע פנייה בזווית y הוא יפנה בזווית אקראית בין 0 ל- 2y.

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

* **צב מדלג (JumpyTurtle) –** צב מדלג הוא צב חכם שמדלג: 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. 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 שאם ההתקדמות של צב מדלג 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 המדלג להתקדם למספר פסיעות וניתורים קטן, והניחו תמיד שהצב יידרש להתקדם למרחקים גדולים.

* **צב ממורמר (GrumpyTurtle) –** צב ממורמר הוא כמו צב חכם, אך בשונה מהצב החכם, חסרה לו מוטיבציה והוא לא טוב במיוחד בשליטה בכעסים, ולכן אינו מצייר את המצולע כראוי:
  1. בהסתברות של 60% הוא יתחיל לצייר את המצולע, אך יתייאש וייעצר צלע אחת לפני הסוף (לא יסיים את המצולע). במקרה זה, בהסתברות של 30% (בלתי תלוי) הוא יתעצבן ויקשקש משהו נוסף כדי להרוס את הצורה שהוא התחיל. (הקישקוש לבחירתכם-כל עוד זה אינו קו שמסיים בהצלחה את הפוליגון)
  2. בהסתברות המשלימה (40%) הוא יחליט לעשות דווקא ויצייר את המצולע לכיוון ההפוך (בהתאם לכיוון שהגדרתם. אם הגדרתם את כיוון המצולע של הצב החכם ימינה, הוא יצייר אותו שמאלה, ולהיפך). לאחר מכן, בהסתברות של 20% הצב יפחד פתאום שתכעסו עליו ויברח מהמסך (ילך מרחק גדול מאוד לכיוון רנדומלי, עד שיצא מהחלון). שימו לב שעל הצב לא לצייר בזמן שהוא בורח (כדי שלא תוכלו לעלות על עקבותיו ולתפוס אותו כמובן).
* **צב בהנגאובר (HangoverTurtle)-** צב בהנגאובר הוא כמו צב שיכור, אך לא תמיד יצליח להיות מרוכז מספיק לביצוע המשימה. בכל פעם שמבקשים ממנו משהו הוא יבצע אחד מהבאים:
  1. בהסתברות של 25% הוא יבצע את הפעולה כמו צב שיכור.
  2. בהסתברות של 20% הוא יבצע את הפעולה כמו צב פשוט.
  3. בהסתברות של 55% הוא לא יבצע כלום.

שימו לב שההסתברויות משלימות- הצב יבצע רק את אחד מהנ"ל.

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

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

1. **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 שיודפס:

Choose the type of the turtle:

1. Turtle
2. Smart
3. Drunk
4. Jumpy
5. Grumpy
6. Hangover

* 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 35
  4. פניה של 20 מעלות שמאלה
  5. צעידה קדימה למרחק של 55
  6. כל מי שיודע לצייר צריך לצייר משושה בעל צלע בגודל 50
  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 וצב בהנגאובר צריכים להצליח להסתדר בשורה עם כל הצבים האחרים בדיוק מופתי, עד אז הם מתמקמים כמו כל הצבים ורק לאחר מכן חוזרים לעצמם.

**הערות לחלק הראשון:**

* יתכן כי הצבים "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 ב'**

בעבודה 1 נדרשתם לממש את משחק "חידת ה-15". בתרגיל זה תצטרכו לממש את המשחק לפי **עקרונות תכנות מונחה עצמים**, עם שינויים קלים. המשחק יורכב מלוח בגודל 4x4, עליו ממוקמים 15 יצורים קסומים מארבעה סוגים שונים.

**מטרת המשחק**: היא לסדר את היצורים כך שבכל טור ובכל שורה לא יחזור על עצמו אותו יצור פעמיים. אין חשיבות למיקום התא הריק בסוף המשחק. כאשר מגיעים למצב זה מנצחים והמשחק מסתיים.

**סוגי היצורים הקסומים:**

|  |  |  |
| --- | --- | --- |
|  | **חד-קרן (Unicorn)** | יכול לנוע לתא הסמוך אליו – גם אם הוא באלכסון, ובנוסף, יכול לדלג מעל יצורים שמפריעים לו בדרך אל התא הריק הנכסף, ובלבד שהוא מדלג מעל יצור אחד בכל פעם. כאשר חד הקרן מדלג, הדילוג לא יכול להתבצע באלכסון. |
| Yeti Coloring Pages - Coloring Pages Kids 2019 | **ביגפוט (Bigfoot)** | יכול לנוע רק לתאים הסמוכים אליו, ולא באלכסון. |
| **סודו של הפניקס - למות ולהיוולד מחדש מתוך המשברים הגדולים של חיינו/ אלעד  פרוטר** | **פיניקס (Fenix)** | יכול לעוף מעל כמה יצורים שהוא רוצה, ולנוע לכל תא – אך במגבלה של תנועה בקו ישר בלבד (אנכי או אופקי). |
|  | **דרקון (Dragon)** | יכול לעוף מעל כמה יצורים שהוא רוצה בקו ישר בדרכו לתא הריק (כמו הפיניקס, אנכי או אופקי). יש מקרה אחד יוצא דופן, בו תנועה אלכסונית מתאפשרת (כולל דילוג מעל חיות אחרות) - רק כאשר בסמוך למשבצת הריקה נמצא ביגפוט. במידה ותנאי זה לא מתקיים, הדרקון יכול לעוף רק בקווים ישרים. |

**אופן המשחק**

* + 1. אתחול -המשחק יחל בהגרלת היצור שיופיע רק 3 פעמים על הלוח, ובמיקום אקראי של כלל היצורים על הלוח. גם מיקום המשבצת הריקה יהיה אקראי.

נבקש מהמשתמש לפני כל משחק מספר הזזות מקסימלי אותו נבצע (יש לוודא קלט גדול שווה ל-1 (, דוגמא:

Number Of Moves: 100

אם לאחר 100 הזזות לא נגיע לפיתרון -נכריז:

The mystical creatures did not find their place!

ונסיים את ההרצה.

* + 1. בכל שלב, יתבקש המשתמש לבחור איזה תא הוא מבקש להזיז. תחילה יבחר את השורה ולאחר מכן את העמודה, במספר רציף. לדוג' – 23 משמעו שורה שלישית ועמודה רביעית. שימו לב שהמספור מתחיל מ-0, כך ש – 0 מייצג את השורה הראשונה, ו-3 את השורה הרביעית.
    2. אם בחר המשתמש להזיז תא שלא ניתן להזיז אותו (לדוג', חד-קרן שצריך לדלג מעל יותר מיצור אחד) – תוצג על כך הודעה והמשתמש יתבקש לבחור תא אחר.
    3. בסוף כל תור יוצג למשתמש הלוח מחדש, וכן תתבצע בדיקה האם הוא ניצח.

**ריצה לדוגמא:**

|  |  |  |  |
| --- | --- | --- | --- |
| U | D | B | F |
| D | B | F | U |
| F | U | U | B |
| B | F |  | D |

Please insert the cell that contains the creature you want to move (row, then column):

30

This creature cannot move to the empty cell. Please try again:

22

|  |  |  |  |
| --- | --- | --- | --- |
| U | D | B | F |
| D | B | F | U |
| F | U |  | B |
| B | F | U | D |

Congratulations! the mystical creatures found their place!

ונסיים את ההרצה.

**דגשים לחלק זה**

* שימו לב לכללי הורשה ופולימורפיזם.
* אתם רשאים להוסיף מחלקות נוספות מעבר למתואר בסיפור. עם זאת, חובה עליכם ליצור את המחלקות **Unicorn**, **Bigfoot**, **Fenix**, **Dragon**. (יש להגיש את כל המחלקות)
* יש להציג את הפלט בדומה לדוגמא שלעיל באופן מסודר וברור.
* שימו לב שמיקום היצורים בלוח ישתנה בין ריצה לריצה.
* יש לשים לב לסדר הכנסת הקלט למשחק. תכניות שיקרסו בשל סדר הכנסה שונה יאבדו נקודות. אין צורך בבדיקות קלט שאינן קשורות לחוקיות המשחק (למשל, אם המשתמש הכניס String במקום מספר).
* **בנוסף לתכנית, יש להגיש קובץ PDF ובו מסמך המתעד את תכנון התוכנית לפי עקרונות ה- Top Down, ותרשים המתאר את ההורשות האובייקטים בחלק זה.**

#### **הערות לשני החלקים**

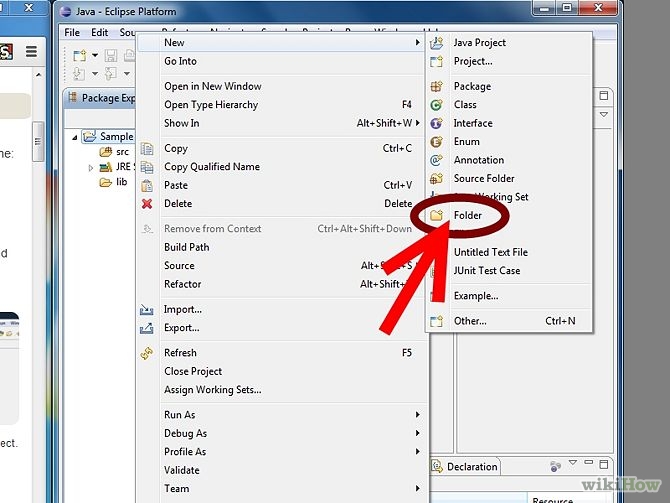
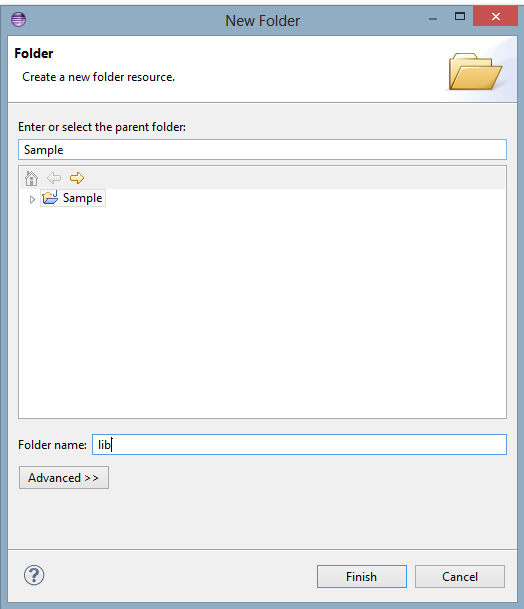
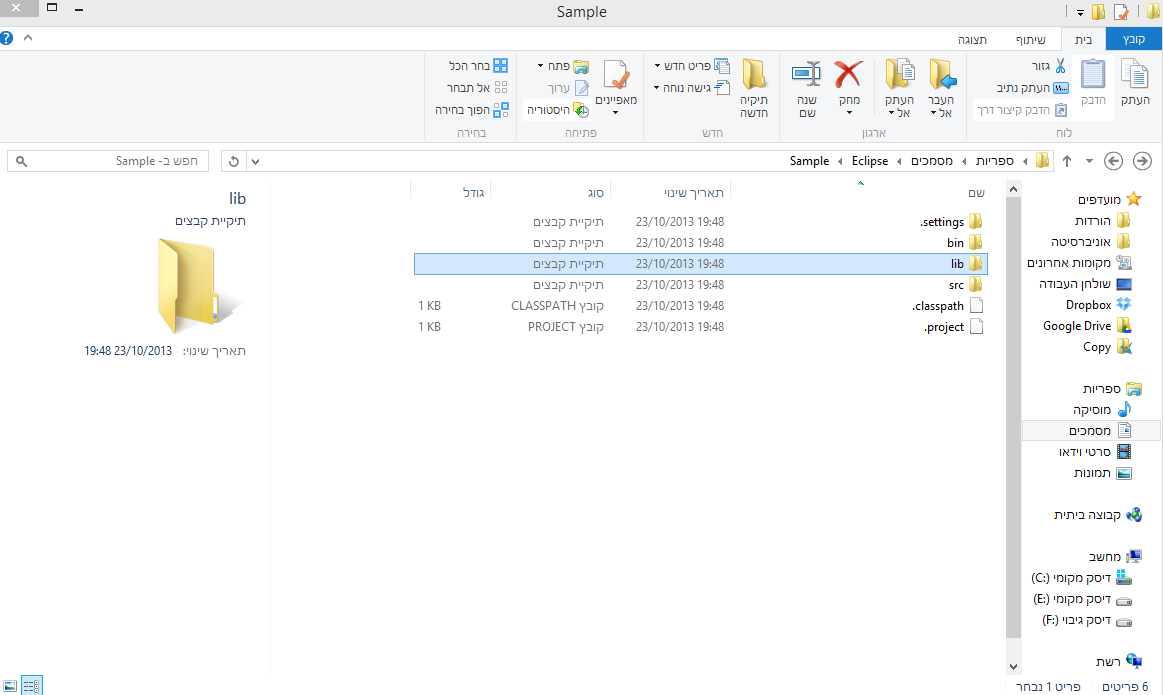
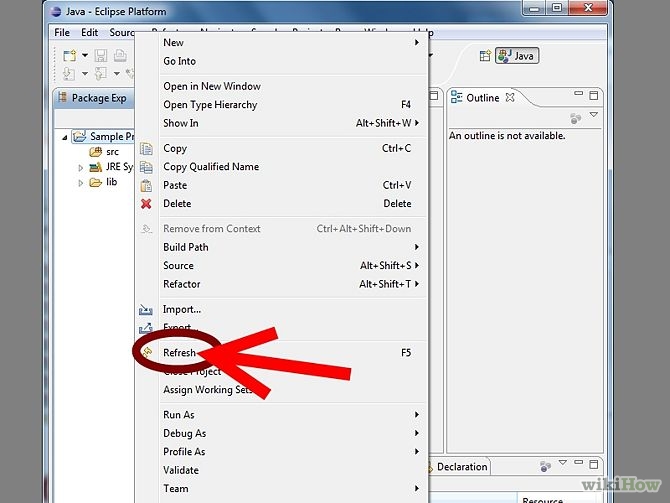
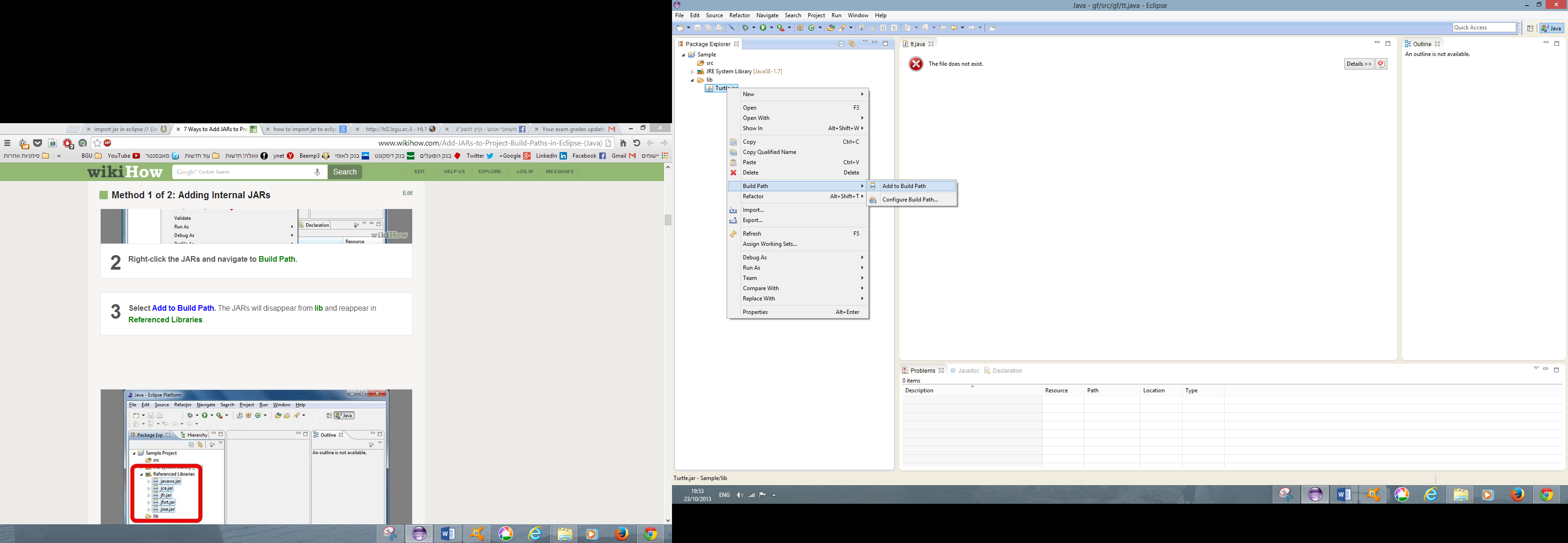
* **הדגש בעבודה זו הוא על שימוש נכון במבנה ירושה ופולימורפיזם**.
* יש להשתמש בעקרונות וכלים של פיתוח תוכנה מונחה עצמים שנלמדו בכיתה. עבודות בהן לא יישומו העקרונות יקבלו ציון נמוך גם אם הפלט יהיה נכון.
* מקרי קצה שלא פורטו – יש לטפל כרעות עיניכם כל עוד לא סתרתם אף הנחיה של העבודה.
* **שימו לב**, אל תשתמשו 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.
* מתרגלת אחראית על התרגיל: ליאור בלכר. בודקי תרגילים: הדר מרמור, עפרי סרוסי.
* שאלות על התרגיל דרך הפורום בלבד.

#### **הוראות הגשה**

* הגשה בזוגות עד לתאריך **29.11.21** בשעה **23:59**.
* הגשה בזוגות.
* יש להגיש את **כל תיקיית הפרויקט** + קובץ של PDF שהתבקשתם להגיש עבור החלק השני **בקובץ zipאחד.**
* יש להגיש את תיקיית הפרויקט כאשר שם התיקייה יהיה מספרי תעודת הזהות. לדוגמא: 111111111\_222222222 בתוכה יש לשים את תיקיית הפרויקט.
* שאלות על התרגיל דרך הפורום או בשעת הקבלה בלבד, שאלות שיישאלו במייל לא יזכו למענה.
* ניתן לשאול בפורום אך ורק שאלות והבהרות על הנחיות התרגיל ולא על דרכי מימוש אפשריות!
* **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.
* דחיות להגשת העבודה יינתנו ע"י אחראי הקורס, 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 HighLearn.
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 -> *Buid 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 stuff