תוכנה 1

תרגיל מספר 8

הנחיות כלליות:

- קראו בעיון את קובץ נוהלי הגשת התרגילים אשר נמצא באתר הקורס.
- הגשת התרגיל תעשה במערכת ה VirtualTAU בלבד (http://virtual2002.tau.ac.il/).

1. יש להגיש קובץ zip יחיד הנושא את שם המשתמש (לדוגמא, עבור המשתמש zvainer יקרא הקובץ zvainer.zip) קובץ ה zip יכיל:
   א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז. הזהות שלכם.
   ב. קבצי ה java של התוכניות אותם התבקשתם לממש.
   ג. קובץ טקסט עם העתק של כל קבצי ה java
   ד. קובץ טקסט בשם answers עם התשובות לשאלות

בתרגיל זה נממש תוכנת ציור פשוטה, jPaint. יש להוריד את קבצי הקוד מאתר הקורס ולייבא את הפרוייקט לתוך Eclipse. הסבר מפורט כיצד יש לעשות כן ניתן למצוא בתרגיל מספר 6.

חשוב!

1. בפרויקט המסופק ישנה חבילה בשם tests. עליכם לוודא, בסיום כל חלק בתרגיל, שהמחלקה המתאימה לו בחבילה זו רצה ללא שגיאות. תרגיל אשר יוגש עם שגיאות קומפילציה ו/או זמן ריצה לא ייבדק החלק המתאים ולא ינוקד. הקוד המסופק מיועד כדי לבדוק את חתימות הפונקציות ולא את המימוש עצמו.
2. שתיים מהמחלקות המסופקות (io.StandardImageSavers ו-ui.ShapesPanel) מראות שגיאת קומפילציה ב-eclipse. שגיאה זו תיפתר כאשר תשלימו את חלק ג'.

חלק א' – המודל (20 נק')

בחלק זה נעסוק במבנה הנתונים אשר יאפשר לנו לבנות ציור. נשים לב שמטרתנו בחלק זה היא לבנות תיאור של הציור בזכרון ולאו דוקא לצייר אותו בפועל. כל המחלקות המתוארות בחלק זה הן חלק מהחבילה model.

נגדיר ציור כאוסף של צורות. איננו יודעים אילו צורות נרצה לצייר ולכן נגדיר מנשק עבור צורות באפליקציה שלנו. צורה תוגדר על ידי הריבוע העוטף אותה, צבע המסגרת שלה, וצבע הרקע שלה:

```
public interface Shape {
	public Rectangle getBounds();
	public Color getFillColor();
	public Stroke getStroke();

	public void setFillColor(Color fill);
	public void setStroke(Stroke stroke);
}
```

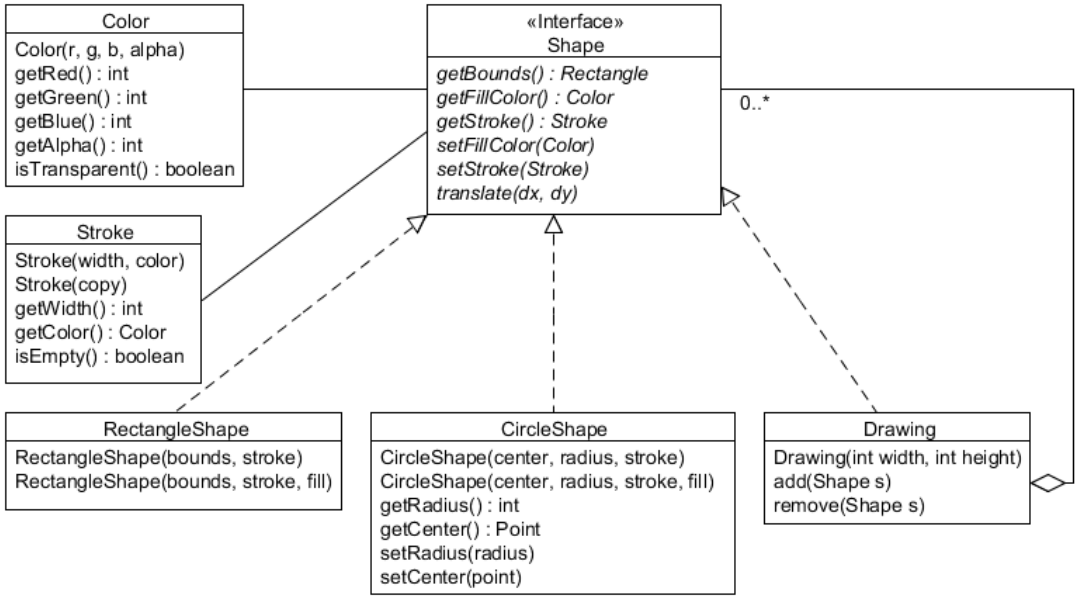
המחלקות Stroke ו- Color כבר הוגדרו עבורכם בפרוייקט.

במחלקה Stroke מוגדרים מאפייני המסגרת (עובי וצבע).

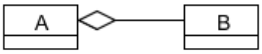
המחלקה Color מגדירה צבעים. צבע מוגדר ע"י הרכיבים אדום, ירוק וכחול (RGB), וכן על ידי רכיב אלפא, המבטא את שקיפותו. ערכי כל אחד מהרכיבים נעים בין 0 ל-255. לדוגמא, הצבע האדום מוגדר ע"י ערך אדום של 255, ערכי 0 עבור כחול וירוק וערך 255 עבור אלפא. במודל צבעים זה ניתן לצייר צורות ריקות ע"י בחירה של צבע שקוף כרקע (ניתן להשתמש בשדה Transparent של המחלקה Color).

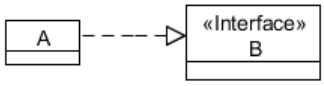
מערכת הצירים: ציר x גדל משמאל המסך לימינו. ציר y גדל מלמעלה למטה.

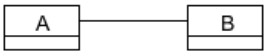
להלן דיאגרמת המחלקות בחבילה model, יש לקרוא את ההערות והחוזים בקוד על מנת להכיר את המתודות השונות לעומק.



תזכורת לגבי הסימונים בתרשים:

יחס הכלה, A מכיל את B 

מימוש מנשק. A ממש את המנשק B 

יש קשר/תלות בין A ל-B, לדוגמא המחלקה A מחזירה ערך מסוג B. 

פרטים נוספים ניתן למצוא ברשת. לדוגמא: http://en.wikipedia.org/wiki/Class_diagram

כעת נממש את המחלקות: Drawing, CircleShape ו-RectangleShape.

1. תחילה נממש את המחלקה Drawing. זוהי מחלקה המתארת ציור שלם. כלומר, משטח ריבועי עליו ניתן למקם צורות נוספות.

   הפעולות הנדרשות:

   א. בנאי המקבל את גודל הציור הנדרש.

```
public Drawing(int width, int height) {

}
```

   ב. מתודות להוספה והסרה של צורות מהציור.

```
public void add(Shape s) {

}

public void remove(Shape s) {

}
```

   ג. מימוש המתודות המוגדרות במנשק Shape תחת הכללים הבאים:

   למחלקה Drawing אין מסגרת. לכן, תמיד יוחזר Stroke.None בקריאה למתודה getStroke(). המתודה setStroke() לא תבצע דבר.

   במחלקה Drawing הגבולות תמיד יתחילות בנקודה (0,0). כלומר: getBounds().getLeft() וכן getBounds().getTop() תמיד יחזירו אפס.

<u>הערות:</u>

- מכיוון שאיננו יודעים מראש כמה צורות יתווספו לציור, נשתמש במבנה רשימה גנרי המסופק ע"י הספרייה הסטנדרטית ב-Java. רשימת הצורות, תשמר בשדה shapes. בקוד המסופק, שדה זה כבר מאותחל. לפרטים יש הסתכל בתיעוד של Java עבור המנשק הגנרי List.
- החבילה core מספקת את המחלקות Point ו-Rectangle.

2. נממש את המחלקה RectangleShape. זוהי מחלקה המתארת מלבן.

א. יש לספק שני בנאים, הראשון מייצג ריבוע ריק, והשני מייצג ריבוע עם מילוי:

```
public RectangleShape(Rectangle bounds, Stroke frame) {

}

public RectangleShape(Rectangle bounds, Stroke frame, Color fill) {

}
```

ב. יש לממש את המתודות של המנשק Shape.

3. נממש את המחלקה CircleShape המתארת עיגול.

א. יש לספק בנאים מתאימים עבור עיגול ריק ועיגול מלא:

```
public CircleShape(Point center, int radius, Stroke frame) {

}

public CircleShape(Point center, int radius, Stroke frame, Color
fill) {

}
```

ב. יש להשלים את החוזים עבור הבנאים וכן את משתמר המחלקה.

ג. יש לממש את המתודות של המנשק Shape.

ד. יש לממש את השאילתות של המחלקה (getCenter, getRadius).

תזכורת: יש לוודא שסיימתם חלק זה בהצלחה ע"י הרצת test.PartA

חלק ב' – חברים (20 נק')

נשים לב שבמודל הנוכחי יש לקבוע את ההגדרות הגרפיות לכל צורה בנפרד. אולם, נרצה אפשרות לבצע פעולה על מספר רב של צורות בבת אחרת. לשם כך, נוסיף צורה אשר תכיל מספר (קבוע) של צורות. (התנהגות דומה לפעולה group בתוכנות ציור ובתוכנות עריכת מצגות). המחלקה תמומש אף היא בחבילה model.

ההתנהגות הנדרשת:

1. יש להגדיר בנאי המקבל את הצורות אשר בקבוצה. הבנאי אינו משנה את תכונות חברי הקבוצה.

```
public Group(Shape[] shapes) {

}
```

2. יש לממש את המנשק Shape. שימו לב שכאשר משנים את הרקע ו/או המסגרת יש לשנות את התכונות של כל חברי הקבוצה. בנוסף, אין לקבוצה מסגרת או מילוי ולכן יש להחזיר Stroke.None ו-Color.Transparent בתגובה לשאילתות אלו.
3. יש לממש את השאילתה size() המחזירה את מספר הצורות בקבוצה.

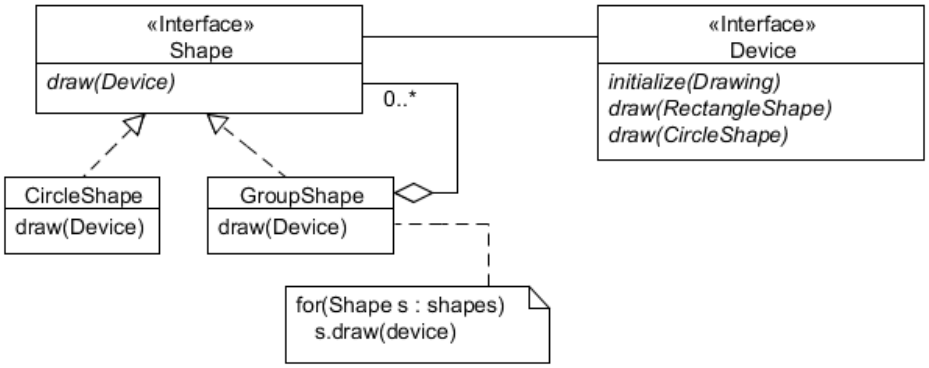
חלק ג' (20 נק')

בנינו בהצלחה את המודל וכעת נרצה לצייר אותו. עולות מספר שאלות:

- איך יודעים אילו צורות יש במודל? וכיצד עוברים עליהן? בהינתן מימוש הקבוצה, יצרנו מבנה מסובך של עץ בזכרון.
- איך יודעים כיצד לצייר? האם מציירים למסך? אולי יוצרים קובץ של תמונה? אם כן, באיזה פורמט?

נתחיל מהסוף. אם איננו יודעים כיצד לצייר, נגדיר מנשק המאפשר לנו לצייר על משטח ציור דמיוני, ומישהו כבר יממש אותו כך שיבצע את הרצוי.

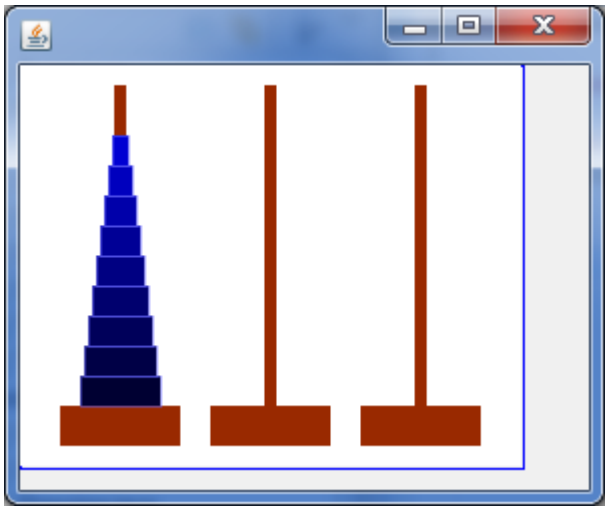
איך נעבור על המודל ונצייר אותו עפ"י ההגדרות? נוסיף מתודה draw() למנשק Shape. צורות "פשוטות", כדוגמת ריבוע או עיגול, יציירו את עצמן ישירות. צורות מורכבות, יעבירו את הבקשה לציור לאוסף הצורות אותו הן מייצגות. כך, הידע על מבנה העץ מוכמס לעצמים המודעים למבנה ואופן ביצוע הפעולה (הציור) נשמר במחלקות המתאימות. למעשה, השתמשנו כאן בתבנית עיצוב המכונה Composite. מומלץ להכיר תבנית זו ולקרוא אודותיה באינטרנט או בספרות.



המנשק Device כבר הוגדר בחבילה render. נותר אם כן להוסיף את המתודה draw למנשק Shape ולממש אותה בכל המחלקות אשר ממשות את המנשק.

```
public void draw(Device d);
```

אם מימשתם נכון אז הרצת test.PartC תציג את החלון הבא:



<u>חלק ד' – קבצים (40 נק')</u>

נרצה לשמור (ולטעון) את הציורים שאנו בונים. נשים לב, שייתכנו מספר צורות לשמור תמונה (קבצי גרפיקה בפורמטים שונים). בנוסף, ייתכן כי פורמטים מסויימים מאבדים מידע ולכן לא ניתן לקרוא מהם. נצטרך לבנות מערכת כללית לשמירה וטעינה של קבצים בפורמטים שונים.

שימו לב! חלק זה מורכב יותר מחלקים הקודמים. וודאו שהבנתם את הנעשה עד כה!

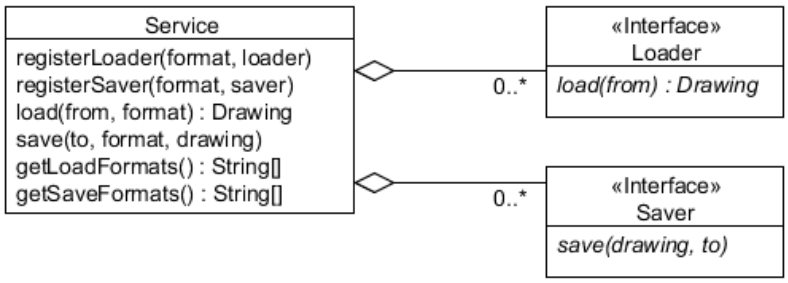
נתחיל מהגדרת מנשקים לשמירת ציור לטעינתו:

```
public interface Saver {
	public void save(Drawing image, String to);
}
public interface Loader {
	public Drawing load(String from);
}
```

כיצד נדע לטפל במספר פורמטים? כיצד נעשה זאת בדרך שתקל עלינו פיתוח של פורמטים נוספים אם יהיה בכך צורך? התשובה: דרושה מזכירות. כל מי שיממש את אחד הממשקים, ירשם במזכירות ואז נדע על קיומו.

בחבילה io מוגדרים הממשקים הנ"ל והמזכירות (המחלקה Service) שלהם. המזכירות גם מציעה שרותי "תווך". פשוט מבקשים לשמור ציור בפורמט נתון והיא מאתרת את המימוש (אם קיים) ומפעילה אותו.

מה הרווחנו? מי שרק רוצה לשמור או לטעון, אינו מכיר כלל את הממשקים Loader/Saver אלא רק את נותן השירות (המזכירות). רק מי שמכיר את פרטי הפורמט המבוקש ומעוניין להציע שירותים, צריך להכיר את הממשקים הנ"ל, בפרט, הוא צריך להכיר רק את החבילה model, ואינו מעוניין בשאר חלקי הקוד.



להדגמה, מסופקת המחלקה StandardImageSavers בחבילה io. מחלקה זו מספקת (באמצעות Java) אפשרות לשמור את התמונות שייצרנו למספר פורמטים פופולריים (jpg, bmp, png).

לדאבוננו, שמירה בפורמטים אלו גורמת לאיבוד מידע רב, מכיוון שהתמונה הנוצרת איננה מכילה את מבנה העץ שלנו (את המודל האובייקטלי).

נרצה להגדיר פורמט שמירה של ציורים לקובץ טקסט, כך שישמור על המודל. ננהג עפ"י הכללים הבאים:

1. צבעים ישמרו על ידי המילה "color" ולאחריה סדרה של 4 ספרות המתאימות (לפי הסדר) לרכיב האדום, הירוק, הכחול והאלפא. כל פרט מידע יופרד ברווח מקודמו.
2. מסגרת (Stroke) תשמר על ידי המילה "stroke" ולאחריה עובי המסגרת. בשורה נפרדת יישמר צבעה עפ"י האמור ב-(1).
3. ריבוע יישמר ע"י המילה "rect", לאחריה הקואורדינטות getLeft() ו-getTop() ולאחר מכן הרוחב (נתון ע"י getWidth()) והאורך – getHeight(). בשורה חדשה תישמר המסגרת לפי (2) ובשורה נוספת ישמר צבע הרקע עפ"י האמור ב-(1)
4. עיגול יישמר ע"י המילה "circle" ולאחריה הקואורדינטות של מרכזו (קואורדינטת x ולאחריה קואורדינטת y) ולבסוף הרדיוס. בשורה חדשה תישמר המסגרת לפי (2) ובשורה נוספת ישמר צבע הרקע עפ"י האמור ב-(1)
5. קבוצה תשמר על ידי המילה group ולאחריה מספר הצורות שהיא מכילה. לאחר מכן ישמרו הצורות אחת אחרי השנייה. כל צורה תתחיל בשורה חדשה.
6. ציור,יישמר ע"י המילה "drawing" ולאחריה נשמור את מימדי הציור (רוחב ואח"כ אורך). בשורה נפרדת יש לשמור את צבע הרקע עפ"י האמור ב-(1).

<u>דוגמא פשוטה:</u>

כאשר נשמור את הציור הנבנה בקטע הקוד הבא:

```
Drawing d = new Drawing(100,100);
d.setFillColor(Color.Red);

Stroke border = new Stroke(5, Color.Blue);
d.add(new CircleShape(new Point(50, 50), 40, border, Color.Green));
```

נקבל את הקובץ:

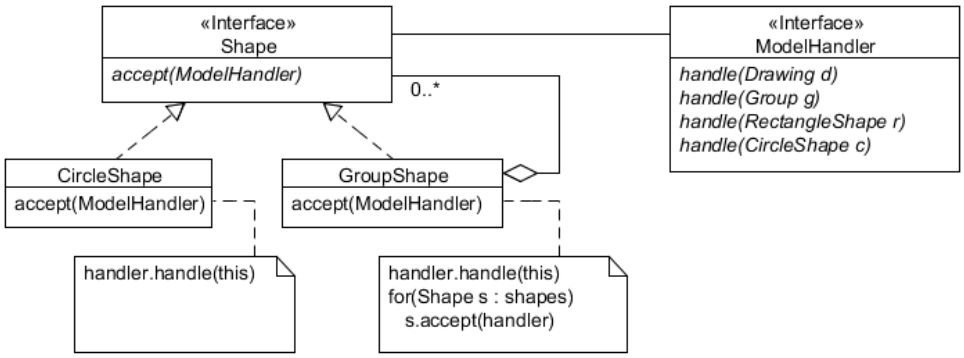
```
drawing 100 100
color 255 0 0 255
circle 50 50 40
stroke 5
color 0 0 255 255
color 0 255 0 255
```

לנוחיותכם, סימנו את השורות בקוד והשורות המתאימות בקובץ הפלט.

דוגמאות מורכבות יותר מסופקות עם התרגיל (קבצי txt. בספרייה resources). הקובץ hanoi.txt מתקבל כתוצאה מהרצת תוכנית הבדיקה test.PartD1 לאחר מימוש אלגוריתם השמירה.

שוב אנו נתקלים בבעיית המעבר על עץ המודל, אולם כעת נדרש גם זיהוי של המחלקות הממשות את המנשק Shape. כאשר ממישנו את ()draw היה מספיק להגדיר Device המספק שירותי ציור כללים (כגון ציור ריבוע או עיגול) והאובייקטים תפעלו אותו כרצונם. כעת אלגוריתם צריך לזהות את האובייקטים ולפעול בהתאם. נציע את הפתרון הבא: האובייקטים יהיו אחראים על המעבר בעץ (כמו ב-draw) דרך המתודה accept, אולם כעת כל אובייקט ידווח כאשר עוברים דרכו. למי לדווח? למישהו חיצוני (האלגוריתם) אשר יידע מה לעשות. כיצד נדווח? ע"י קריאה לפונקציה handle של האלגוריתם עם האובייקט הנוכחי (this) כפרמטר.

נסכם הדברים בדיאגרמה הבאה:



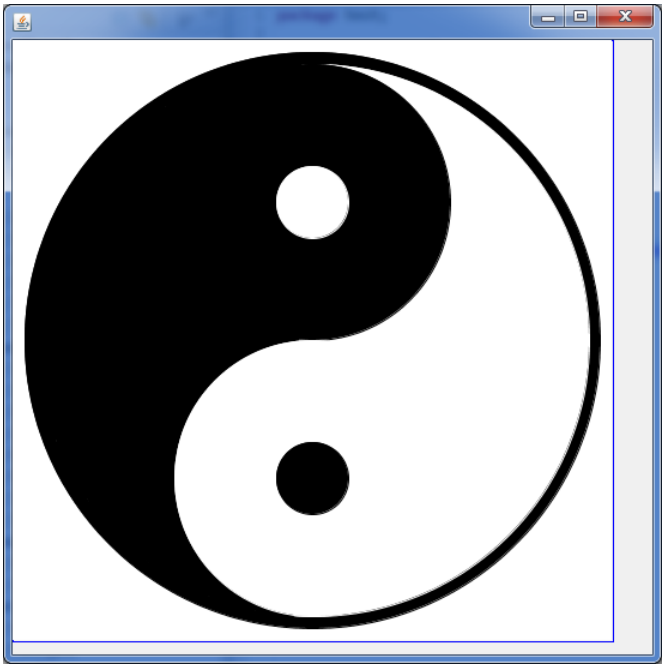
א. המנשק ModelHandler כבר מוגדר במחלקה model. יש להוסיף את המתודה ()accept למנשק Shape ולממש בהתאם.

ב. יש להשלים את מימוש המחלקה TextSaver הממשת את המנשקים ModelHandler ו-Saver ושומרת את הציור עפ"י ההגדרות לעיל.

ג. יש להשלים את מימוש המחלקה TextLoader הטוענת את ציור הנתון בפורמט הנ"ל.

הערות:

1. המחלקה TextSaver כבר כוללת קוד המאפשר כתיבה לקובץ בדומה לכתיבה למסך. ניתן להשתמש בשדה out של המחלקה TextSaver כמו ב-System.out.
2. עבור חלק זה של התרגיל מסופקות שתי מחלקות בדיקה: PartD1 הבודקת את השמירה בלבד, ו-PartD2 הבודקת גם את הטעינה. בנוסף, בתיקייה resources תוכלו למצוא את הקובץ Hanoi.txt המראה מהו הפלט בצריך לקבל מהרצת PartD1.
3. ניתן לשנות את תוכנית הבדיקה כך שתטען את קובץ הבדיקה yinyang.txt. התוצאה הרצויה:



4. יש להשתמש ב- string.compareTo(other)==0 עמ"נ לבדוק אם שתי מחרוזות הן שוות.

ב ה צ ל ח ה !