**שאלון Y - תכנות ותיכון מונחה עצמים – 10806, ד"ר יצחק אביב**

הבחינה כוללת תיקייה בשם yyyyyyyyy ובה הקבצים הנדרשים לבחינה. עליכם להחליף את שם התיקייה מ-yyyyyyyyy למספר ת.ז שלכם. לאחר שביצעתם את שינוי שם התיקייה יש לטעון אותה ל- pyCharm כ-project. עם סיום הבחינה יש לכווץ את התיקייה ולשלוח אותה. אי ביצוע שינוי השם, או שליחה בחזרה של קבצים נפרדים בסיום הבחינה ולא את התיקייה כולה מכווצת בשם ת.ז שלכם, יגרום לאי בדיקת הבחינה.

משך הבחינה: 180 דקות (3 שעות).

התוכנית צריכה להיות ללא שגיאות קומפילציה, לעבוד ללא תעופות, ולתת תוצאות נכונות.

פלט התוכנית נכתב על גבי קובץ פלט (קובץ טקסט): results.txt. כל מה שקשור בניהול הקובץ: פתיחתו, סגירתו ופעולות הכתיבה עליו, נתון לכם במלואו. כדי לראות אם התוצאות שלכם נכונות ושלמות, יש לעיין בקובץ זה לאחר הריצה (ניתן לעשות בלי לצאת מה- pyCharm - להקליק על הקובץ results.txt שנמצא ב-project).

יש להשלים את קבצי הפיתון ב-project לפי ההנחיות הבאות. ההשלמות תבואנה במשפטים (מספר המשפטים להשלמה בכל מקרה נתון לשיקולכם), אליהם מוצמדת ההערה # change it (אלו משפטי pass, או משפטי return 0, או משפטי (return “”:

1. המחלקה GeometricObject.

מחלקה עבור צורת גאומטריות. יש בה שתי תכונות: \_\_color : צבע הצורה הגאומטרית - מחרוזת תווים, \_\_filled : האם יש למלא את הצורה הגיאומטרית בצבע – True או False (בפועל הצורה הגאומטרית לא מצוירת במערכת הזו).

מחלקה זו ניתנת במלואה. אין להשלים בה דבר ואין לשנות אותה. יש לקרוא ולהבין אותה.

1. המחלקה Circle (14 נקודות)

מחלקה עבור הצורה הגאומטרית מעגל. מחלקה זו יורשת מהמחלקה GeometricObject. יש לקרוא את המתודות הנתונות במחלקה ולהבין אותן. יש להשלים במחלקה Circle 6 מתודות: \_\_init\_\_, getRadius, setRadius, getDiameter, getPerimter, \_\_str\_\_, לפי ההסבר הבא:

המתודה \_\_init\_\_ קובעת את ערכי התכונות במחלקה בהתאם לפרמטרים שהיא מקבלת. במידה והפרמטר radius קטן או שווה ל-0, המתודה \_\_init\_\_ תעורר חריגה מסוג ValueError. הודעת החריגה תכלול את שם המחלקה, שם התכונה שגרמה לחריגה (radius), ערך הרדיוס הלא תקין והודעה כי הוא חייב להיות גדול מ-0. במקרה הזה של החריגה האובייקט לא ייווצר ומתודת ה-main שתוסבר בהמשך תסיים את פעולתה.

המתודה getRadius מחזירה את ערך הרדיוס.

המתודה setRadius מציבה בתכונה רדיוס המעגל את ערך הפרמטר שקיבלה. במידה והועבר ערך קטן או שווה ל-0 המתודה setRadius לא תשנה את רדיוס המעגל (כלומר לא תעשה כלום).

המתודה getDiameter מחזירה את קוטר המעגל שהוא פעמיים הרדיוס.

המתודה getPerimeter מחזירה את היקף המעגל שהוא קוטר המעגל כפול pi (במתודה אחרת באותה מחלקה ניתן לראות איך משתמשים ב-(pi.

המתודה \_\_str\_\_ מחזירה מחרוזת תווים שהיא המחרוזת אותה תכתוב המתודה writeObject עבור אובייקט זה. את מבנה המחרוזת ותוכנה ניתן לראות בהמשך בהצגת התוצאות של ריצת התוכנית לאחר השלמתה, עבור השורה בקובץ הפלט של אובייקט מסוג Circle, וגם בדוגמא הבאה:

Circle: color: Black filled: False radius: 4 area: 50.3

המתודה writeObject שנתונה לכם כותבת לקובץ outFile את מה ש-\_\_str\_\_ מחזירה.

1. המחלקה Rectangle (14 נקודות)

מחלקה עבור הצורה הגאומטרית מלבן. מחלקה זו יורשת מהמחלקה GeometricObject. יש לקרוא את המתודות הנתונות במחלקה ולהבין אותן. יש להשלים במחלקה Rectangle 6 מתודות: \_\_init\_\_, getWidth, getHeight, setWidth, setHeight \_\_str\_\_, לפי ההסבר הבא:

המתודה \_\_init\_\_ קובעת את ערכי התכונות במחלקה בהתאם לפרמטרים שהיא מקבלת. במידה והפרמטר width או הפרמטר height, קטן או שווה ל-0, המתודה \_\_init\_\_ תעורר חריגה מסוג ValueError. הודעת החריגה תכלול את שם המחלקה, שם התכונה שגרמה לחריגה (width או height), ערך התכונה הלא תקין והודעה כי הוא חייב להיות גדול מ-0. במקרה הזה של החריגה האובייקט לא ייווצר ומתודת ה-main שתוסבר בהמשך תסיים את פעולתה.

המתודה getWidth מחזירה את רוחב המלבן.

המתודה getHeight מחזירה את גובה (אורך) המלבן.

המתודה setWidth מציבה בתכונה רוחב המלבן את הערך שקיבלה כפרמטר. במידה והועבר ערך שהוא קטן או שווה ל-0, המתודה לא תשנה את רוחב המלבן (כלומר לא תעשה כלום).

המתודה setHeight מציבה בתכונה גובה (אורך) המלבן את הערך שקיבלה כפרמטר. במידה והועבר ערך שהוא קטן או שווה ל-0, המתודה לא תשנה את גובה המלבן (כלומר לא תעשה כלום).

המתודה \_\_str\_\_ מחזירה מחרוזת תווים שהיא המחרוזת אותה תכתוב המתודה writeObject עבור אובייקט זה. את מבנה המחרוזת ותוכנה ניתן לראות בהמשך בהצגת התוצאות של ריצת התוכנית לאחר השלמתה, עבור השורה בקובץ הפלט של אובייקט מסוג Rectangle, וגם בדוגמא הבאה:

Rectangle: color: green filled: True width: 1 height: 3 area: 3

המתודה writeObject שנתונה לכם כותבת לקובץ outFile את מה ש-\_\_str\_\_ מחזירה.

שלוש המחלקות הבאות נמצאות באותו קובץ פיתון ששמו regularPolygon.

1. המחלקה RegularPolygon (14 נקודות)

מחלקה עבור פוליגונים (מצולעים) שכול הצלעות שלהם בעלי אותו אורך. המחלקה יורשת מהמחלקה GeometricObject. למחלקה RegularPolygon תכונה אחת משלה \_\_side שהיא אורך הצלע בפוליגון. המתודה \_\_init\_\_ במחלקה RegularPolygon מקבלת 3 פרמטרים: side – אורך הצלע בפוליגון ושני ערכים הנדרשים עבור מחלקת האם שלה - GeometricObject.

יש להשלים במחלקה RegularPolygon 2 מתודות: המתודה \_\_init\_\_ ו-getSide לפי ההסבר הבא:

המתודה \_\_init\_\_ - קובעת את ערכי תכונות המחלקה בהתאם לפרמטרים שהיא מקבלת. יש לשים לב שהתכונה \_\_side היא התכונה היחידה השייכת ישירות למחלקה RegularPolygon ואת שאר התכונות היא יורשת מ-GeometricObject. במידה והפרמטר side קטן או שווה ל-0, המתודה \_\_init\_\_ תעורר חריגה מסוג ValueError. הודעת החריגה תכלול את שם המחלקה, שם התכונה שגרמה לחריגה (side), ערך התכונה הלא תקין והודעה כי הוא חייב להיות גדול מ-0. במקרה הזה של החריגה האובייקט לא ייווצר ומתודת ה-main שתוסבר בהמשך תסיים את פעולתה.

המתודה getSide – מחזירה את ערך התכונה \_\_side.

1. המחלקה Square (14 נקודות)

מחלקה עבור ריבוע (כל הצלעות שוות). המחלקה יורשת מהמחלקה RegularPolygon. למחלקה אין תכונות משלה, כל התכונות שלה באות מההורשה. יש להשלים במחלקה Square את 3 מתודות: \_\_init\_\_, getArea, \_\_str\_\_ , לפי ההסבר הבא:

המתודה \_\_init\_\_: קובעת את ערכי תכונות המחלקה בהתאם לפרמטרים שהיא מקבלת.  
  
המתודה getArea – לא מקבלת פרמטרים. מחשבת ומחזירה את שטח הריבוע שהוא אורך צלע הריבוע כפול אורך צלע הריבוע.  
  
המתודה \_\_str\_\_ מחזירה מחרוזת תווים שהיא המחרוזת אותה תכתוב המתודה writeObject עבור אובייקט זה. את מבנה המחרוזת ותוכנה ניתן לראות בהמשך בהצגת התוצאות של ריצת התוכנית לאחר השלמתה, עבור השורה בקובץ הפלט של אובייקט מסוג Square, וגם בדוגמא הבאה:

Square color: red filled: True side: 10 area: 100  
  
המתודה writeObject שנתונה לכם כותבת לקובץ outFile את מה ש-\_\_str\_\_ מחזירה.

1. המחלקה EquilateralTriangle (14 נקודות)

מחלקה עבור משולש שווה צלעות (כל הצלעות שוות). המחלקה יורשת מהמחלקה RegularPolygon. למחלקה אין תכונות משלה, כל התכונות שלה באות מההורשה. יש להשלים במחלקה EquilateralTriangle 3 מתודות: \_\_init\_\_, getArea , \_\_str\_\_.

המתודה \_\_init\_\_: קובעת את ערכי תכונות המחלקה בהתאם לפרמטרים שהיא מקבלת.

המתודה getArea – לא מקבלת פרמטרים. מחשבת ומחזירה את שטח המשולש שהוא אורך צלע המשולש (שווה הצלעות) כפול אורך צלע המשולש כפול 0.433.

המתודה \_\_str\_\_ מחזירה מחרוזת תווים שהיא המחרוזת אותה תכתוב המתודה writeObject עבור אובייקט זה. את מבנה המחרוזת ותוכנה ניתן לראות בהמשך בהצגת התוצאות של ריצת התוכנית לאחר השלמתה, עבור השורה בקובץ הפלט של אובייקט מסוג EquilateralTriangle, וגם בדוגמא הבאה:

EquilateralTriangle color: blue filled: False side: 5 area: 10.8

המתודה writeObject שנתונה לכם כותבת לקובץ outFile את מה ש-\_\_str\_\_ מחזירה.

1. המחלקה EquilateralPentagon (14 נקודות)

מחלקה עבור מחומש שווה צלעות (כל 5 הצלעות שוות). המחלקה יורשת מהמחלקה RegularPolygon. למחלקה אין תכונות משלה, כל התכונות שלה באות מההורשה. יש להשלים במחלקה EquilateralPentagon 3 מתודות: \_\_init\_\_, getArea ו-\_\_str\_\_.

המתודה \_\_init\_\_: קובעת את ערכי תכונות המחלקה בהתאם לפרמטרים שהיא מקבלת.

המתודה getArea – לא מקבלת פרמטרים. מחשבת ומחזירה את שטח המחומש שהוא אורך צלע המחומש (שווה הצלעות) כפול אורך צלע המחומש כפול1.72048 .

המתודה \_\_str\_\_ מחזירה מחרוזת תווים שהיא המחרוזת אותה תכתוב המתודה writeObject עבור אובייקט זה. את מבנה המחרוזת ותוכנה ניתן לראות בהמשך בהצגת התוצאות של ריצת התוכנית לאחר השלמתה, עבור השורה בקובץ הפלט של אובייקט מסוג EquilateralPentagon, וגם בדוגמא הבאה:

EquilateralPentagon color: yellow filled: True side: 5 area: 43.0

1. מתודת main בקובץ polymorphism (16 נקודות)

יש לקרוא ולהבין אותה.  
יש להחליף בה את משפט ה-pass האחד שמופיע בה במשפט שממיין את הליסט l לפי שטח האוביקט (שהוא צורה גאומטרית) בסדר יורד. מעבר לכך אין לשנות בה דבר.

הבהרות נוספות:

את שטח הצורה הגאומטרית שנכתב כחלק מהאובייקט על גבי קובץ הפלט יש לעגל במתודה \_\_str\_\_, בכול אחת ממחלקות הצורות, לספרה אחת אחרי הנקודה. עיגול מספר x מתבצע כך: round(x, 1) (אין צורך ב-import כלשהו עבור פעולת העיגול).

המתודה writeObject שכותבת כל אחד מהאובייקטים על קובץ הפלט ניתנת בכול המחלקות בשלמותה.

לא ניתן להוסיף תכונות נוספות לאף אחת מהמחלקות מעבר לתכונות שהוגדרו ששייכות להן. למשל, התכונה side אינה תכונה של המחלקות square ו- EquilateralTriangle, הן מכירות אותה בזכות ההורשה.

כל התכונות בכול המחלקות הן מסוג private. כלומר כולן חייבות להתחיל ב- \_\_.

אין להוסיף מחלקות נוספות למערכת.

אין לשנות את ההגדרות של המתודות (שם המתודה והפרמטרים שהיא מקבלת) שניתנות בשלמותן ואת אלו שיש להשלים.

**יש לנצל, היכן שניתן, את ההורשה. זה יבוא לידי ביטוי במתודות \_\_init\_\_ ו-\_\_str\_\_.**

מצורף הפלט של ריצת התוכנית לאחר השלמתה כנדרש - תוכן הקובץ הפלט results.txt:

list of geometric objects before sort:  
Circle: color: Black filled: False radius: 4 area: 50.3  
Rectangle: color: green filled: True width: 1 height: 3 area: 3  
Square color: red filled: True side: 10 area: 100  
EquilateralTriangle color: blue filled: False side: 5 area: 10.8  
EquilateralPentagon color: yellow filled: True side: 5 area: 43.0  
  
list of geometric objects after sort (descending):  
Square color: red filled: True side: 10 area: 100  
Circle: color: Black filled: False radius: 4 area: 50.3  
EquilateralPentagon color: yellow filled: True side: 5 area: 43.0  
EquilateralTriangle color: blue filled: False side: 5 area: 10.8  
Rectangle: color: green filled: True width: 1 height: 3 area: 3  
  
from RegularPolygon:  
side = 0 must be > 0.

ב ה צ ל ח ה