

תרגיל 2

תאריך פרסום: 03.04.23

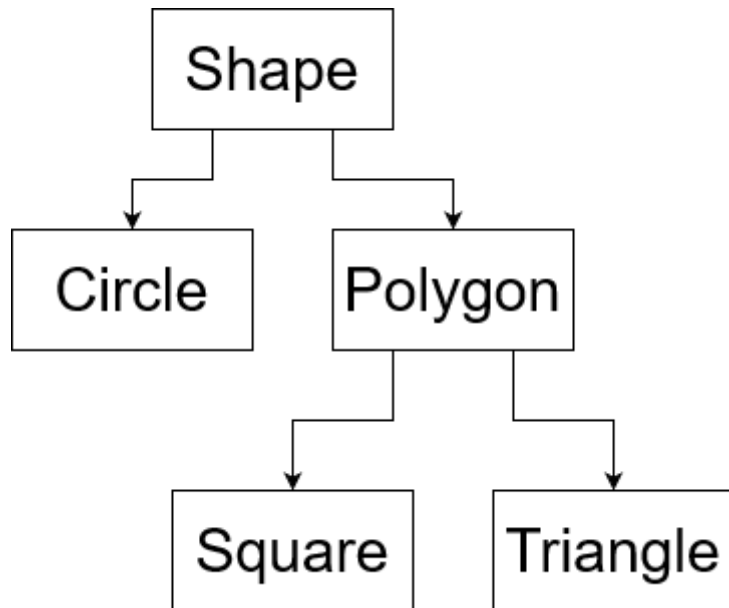
תאריך הגשה: 22.04.23

דגשים:

- קראו בקפידה את הנחיות ההגשה. אל תגישו קבצים או קוד מיותר ווודאו שאתם עומדים בכל הנחיות לפני ההגשה.
- ניתן להשתמש רק במה שנלמד עד הרצאה 4 (כולל) וב-Math (ובמתודות ואופרטורים שמצויינים לאורך העבודה).
- **אין לשנות חתימה של מתודות או קוד קיים בשום צורה.**
- ניתן להשתמש ב-break ו-continue בלולאות.
- מצורף לכם קובץ בדיקה שמאפשר לכם לבדוק את הקוד שלכם, אולם הוא לא מהווה את הבדיקה הכוללת שעליה ניתן לכם הציון.
- יש להקפיד לרשום הערות לאורך הקוד.
- ההנחות שניתן להניח לגבי הפרמטרים רשומות בתיאור העבודה.
- שאלות לגבי העבודה יש לשאול בפורום המיועד.
- במקרה שבו הקוד עובר את מצב הריצה בהצלחה מלאה, אולם לא מתקפל או קורס לחלוטין במצב בדיקה, יש ליצור קשר עם ליאור במייל בצירוף הקבצים ותיאור הבעיה.

הערות לביצוע העבודה:

- בכל אובייקט עליכם לממש בנאי שמקבל את הערכים של כל השדות שיש לאותו אובייקט. על הבנאים האלה להעתיק באופן עמוק אובייקטים.
- בכל אובייקט עליכם לממש את המתודה equals.
- בכל אובייקט, חישבו האם ניתן לממש בו את המתודה הנדרשת, או שצריך לעשותה אבסטרקטית.
- ניתן ואף רצוי להשתמש בתגית @Override:
<https://docs.oracle.com/javase/17/docs/api/java/lang/Override.html>
- שימו לב למחלקת Point שכבר נתונה לך ולמתודות שכבר ממומשות בה.



עליכם לממש את עץ ההורשה שמתואר
באיור הבא:

מחלקת Shape

מחלקה שמייצגת צורה כלשהי במרחב.

שדות:

- שם הצורה (יכול להיות null).

מתודות:

- חישוב שטח הצורה.
- חישוב היקף הצורה.
- הזזת הצורה לאורך ציר ה-x וציר ה-y במספר שניתן (הסימן מסמן את הכיוון).

הערות:

- מחלקה זו מממשת את הממשק EdgePointer.

מחלקת Circle

מחלקה שמייצגת עיגול.

שדות:

- רדיוס (ניתן להניח שלא שלילי).
- נקודה מרכזית.

מתודות:

- בנאי דיפולטיבי שמאתחל את העיגול להיות עם רדיוס 0 ונקודה דיפולטיבית.
- בנאי העתקה שמעתיק את העיגול השני באופן עמוק.
- compareTo שמשווה ערכים לפי הרדיוס (עיגול קטן מעיגול אחר אם הרדיוס שלו קטן יותר).

הערות:

- מחלקה זו מממשת את הממשק Comparable.
- במחלקה זו, במקום להשתמש ב- π , יש להשתמש בקירוב שניתן במחלקה.

Polygon מחלקת

מחלקה שמייצגת צורה סגורה (ניתן להניח שחוקית – ללא null והנקודות מייצרות צורה סגורה) שבנויה מאוסף נקודות וקווים המחברים בינם.

שדות:

- מערך נקודות.

הערות:

- אין לממש מתודה שנותנת גישה למערך הנקודות עצמו אל מחוץ למחלקה (getter מלא), יש להשתמש אך ורק ב-getPointAt().

Square מחלקת

מתודות:

- חישוב היקף בצורה יעילה יותר על ידי בדיקת שתי נקודות לכל היותר ממערך הנקודות.

Triangle מחלקת

הערות:

- ניתן לחשב שטח של משולש לפי שלוש נקודות ע"י הנוסחה הבאה (כאשר הנקודות הן $((x_1, y_1), (x_2, y_2), (x_3, y_3))$:

$$\left| \frac{1}{2} \cdot (x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)) \right|$$

EdgePointer ממשק

ממשק שנותן פונקציונליות לגישה לנקודות קצה של האובייקט הממש.

מתודות:

- החזרת ערך ה-x השמאלי ביותר שמופיע לאורך ההיקף של האובייקט הממש.
- החזרת ערך ה-y הגבוה ביותר שמופיע לאורך ההיקף של האובייקט הממש.

בנוסף, עליכם לממש מערך שמכיל את כל הצורות הנ"ל.

מחלקת ShapeContainer

מערך שמכיל צורות שונות ומספק מתודות שפועלות עליהם.

שדות:

- מערך של צורות.

מתודות:

- בנאי דיפולטיבי שמאתחל מערך בגודל 1.
- הוספת צורה (לא null) למערך (העתקה שטחית) במקום הפנוי הבא.
- החזרת האינדקס הראשון בו צורה מופיעה. אם היא לא מופיעה, המתודה תחזיר -1.
- חישוב סכום היקף כל הצורות.
- חישוב סכום השטח של העיגולים.
- חישוב סכום השטח של הפוליגונים.
- סידור הצורות במערך לפי comparator כלשהו. ניתן לעשות זאת על ידי קריאה למתודה `Arrays.sort(arr, Comp)`.

הערות:

- בבנאי של מחלקה זו ניתן להעתיק באופן שטוח את המערך המתקבל.
- הגדלת המערך תיעשה על ידי הכפלת גודלו ב-2.
- על ערכי null להימצא ברצף בסוף המערך (ניתן להניח שמערכים שמתקבלים בבנאי עונים על תנאי זה).
- מחלקה זו ממשלת את הממשק Iterable (ועל כן יש לממש את Iterator שמופיע בה). ה-iterator לא יחזיר ערכי null שנמצאים בסוף המערך.
- למחלקה זו שלוש מחלקות פנימיות שמממשות את Comparator:
 - משווה לפי שם הצורה (לפי ההשוואה של מחרוזות).
 - משווה לפי הנקודה השמאלית ביותר בצורה (אובייקט יהיה קטן יותר אם לנקודה השמאלית ביותר שלו ערך קטן יותר). שבירת-שיוויון נעשית לפי הנקודה הגבוהה ביותר בצורה (אובייקט יהיה קטן יותר אם לנקודה הכי גבוהה שלו ערך קטן יותר).
 - משווה לפי היקף הצורה (אובייקט יהיה קטן יותר אם ההיקף שלו קטן יותר).
 - ערכי null תמיד צריכים להופיע אחרונים.