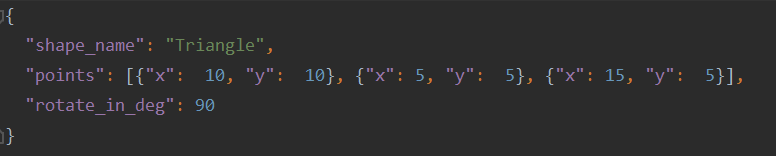
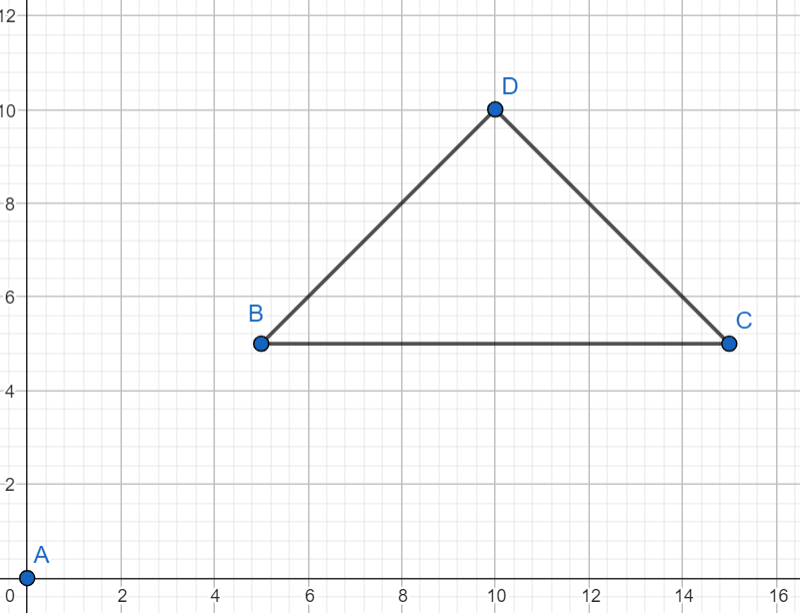
דגשים לבדיקת תרגיל ה-OOP:

* המטרה העיקרית בתרגיל זה היא לתת לנחפפת להתנסות בפיתוח מודולרי ולתכנן את הארכיטקטורה מראש תוך שמת דגש על עקרונות בסיסיים מתכנות מונחה עצמים ו-design patterns.
* אחרי קריאת התרגיל, יש להנחות את הנחפפת לאפיין קודם כל את הקלט והפלט של התכנית, ובעיקר להגדיר את הסכמה של ה-JSON (הקלט). **מומלץ לזרוק למים ולתת לנחפפת להעלות שאלות בעצמה.** חשוב להתעכב בשלב הזה עד שנסגרים באופן הרמטי על הסכמה. מומלץ לתפוס עמדה של לקוח ותוך דיון איטרטיבי של שאלות ותשובות עם הנחפפת להגדיר את הדרישות:
  + על ה-JSON לאפשר את שלושת הפעולות המוגדרות בתרגיל: translation, rotation, scale לכל אחת מהצורות, הן הבסיסיות והן ה-composite.
  + מצופה מהנחפפת להגדיר ולסגור מולך איך הנקודות ב-JSON מתורגמות לצורה. לדוגמא, אם נתון הקלט הבא:



הצורה מצוירת כך:



האם הצורה מוגדרת רק ע"י נקודות המשולש? סביב איזו נקודה יש לעשות את ה-rotate? מרכז המסה של המשולש או ראשית הצירים סביבה הוגדר המשולש ב-JSON? כל אלה הן שאלות שחשוב שיצופו בשלב האפיון כי הן בלתי תלויות בקוד. **מומלץ להכווין את הנחפפת לכך שהסיבוב יהיה סביב ה-(0,0) סביבו הוגדרה הצורה כי זה מקל על הפיתוח בהמשך ויותר אלגנטי.**

* + ה-JSON צריך לאפשר הפנייה לצורה שהוגדרה בקובץ JSON אחר.
  + חשוב לבקש מהנחפפת ליצור קבצי JSON לדוגמא שממחישים את כל הדרישות ולתאר איך הפלט אמור להראות בהינתן הקלטים לדוגמא. (בשלב הזה סביר שיצופו שאלות חשובות כמו – סביב איזו נקודה לסובב).
* לאחר שלב האפיון יש להנחות את הנחפפת לבנות את הארכיטקטורה של הפרויקט.
  + חשוב לשים דגש בשלב זה על ה-design patterns.
  + ארכיטקטורה לדוגמא: (אפשר לסטות ממנה אך לשמור על עקרונות דומים).

**להוסיף פה את הארכיטקטורה**

* + חשוב לשים לב שיחסי הירושה מוגדרים באופן הגיוני. שהמחלקה Shape תהיה אבסטרקטית**,** שיהיה Factory שיבנה את הצורות מה- JSONושהצורות עצמן יהיו מנותקות מאופן כתיבת ה-json. שתהיה מחלקת צייר שמציירת את הצורות (עדיף שתהיה singleton).
* לאחר שלב הגדרת הארכיטקטורה יש לעבור למימוש. חשוב להנחות את הנחפפת לכתוב טסטים תוך כדי הפיתוח כדי שטעויות לא ייגררו. בבדיקת התרגיל יש לבדוק שהמימוש תואם לארכיטקטורה, שאין שכפול קוד, שיש תיעוד.
  + בונוס: מימוש **memoization**.מכיוון ש-CompositeShapes יכולים לחזור על עצמם (למשל, house ב-street), אין צורך לבנות את הבית כל פעם מחדש. ניתן לשמור מילון של כל הצורות שנבנו עם האובייקט המתאים של הצורה ולהעתיק את האובייקט בפעם הבאה שנתקלים שוב בצורה.