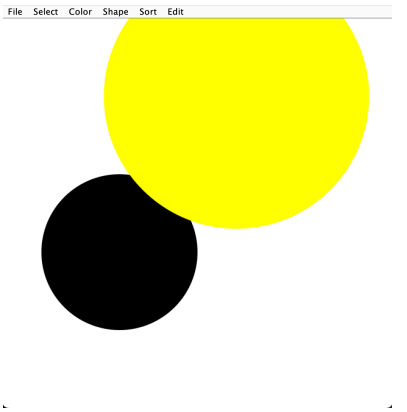
Ariel University, School of Computer Science, 2024

Introduction to Computer Science

Ex2: Main OOP Assignment

Abstract:

This assignment includes a relatively complex set of classes and interfaces. You are asked to implement a set of **geometric shapes**, a shape container, a basic GUI, and basic save/load capabilities. Moreover, in this assignment, you are required to implement a JUnit class - to test all your functions (Geo, Collections, save & load. GUI classes are **not** required to have JUNIT tests).



To Do

1. Download Ex2_run.zip, and uncompress it.
2. Run the Ex2_sol-out.jar file (java -jar Ex2_sol.V0.1-out.jar). This is a complete solution to Ex2 - you should implement your solution accordingly. Try loading the a0 file - you should see an image as shown in Figure 1 (Left and Right). The a1 file is shown in Figure 2.
3. Create a new project (named Ex2), and download all the classes and interfaces of Ex2.zip. These files contain most of the required classes (and all the interfaces). The supplied “skeleton” is runnable (run the Ex2Main t2() function). You should get a simple GUI that will allow you to draw circles.
4. Implement and update your solution to all the GeoShape classes: (Point2D), Circle2D**,** Rect2D, Segment2D, Triangle2D**,** and **Polygon2D**. Note: the polygon is a relatively complex class - make sure to go over the guidance of area, and contains. Concerning the **area** and the **contains** methods - one may assume that the polygon is **simple** (no self-intersection).
5. Implement and update the related needed classes, in particular: Ex2, GUIShape, and ShapeCollection - as defined in GUI_Shape_Collection (you can add additional classes if needed).
6. Implement detailed JUnit classes (this time there are no skeleton classes given to you - make sure you implement a complete testing

suite in classes: Point2DTest, Rect2DTest, Segment2DTest, Triangle2DTest, Polygon2DTest, GUIShapeTest, ShapeCollectionTest, Ex2Test

7. **Add detailed documentation (in English) to the Ex2.java. Moreover, each class should have a** related documentation for each function.
8. Make sure to submit two files: Ex2.zip, Ex2.jar. Ex2.zip should contains ALL the needed classes, as a single zip file containing alll the classes (in the same format as in Ex2.zip). Ex2.jar should be an “executable” jar that can be run by double-clicking. See this link for an explanation regarding how to create a jar file with IJ.

Notes:

1. Work in alone! - you can talk about this assignment with anyone in class - but when writing your solution DIY!. Please go over this document which covers the School’s honesty policy. Make sure to read the remarks regarding chatGPT at the end of this document.
2. **Make sure you write your ID in the files Ex2.java**
3. The implementation of the function should be as efficient and elegant as possible.
4. **Keep in mind: we have “planted” few minor “bugs” in the suggested jar file ⇒ in order to force you to use proper testing! (and not just compare with our solution).**
5. Your solution should be submitted to Moodle according to the instructions - as presented to you in the TA sessions.

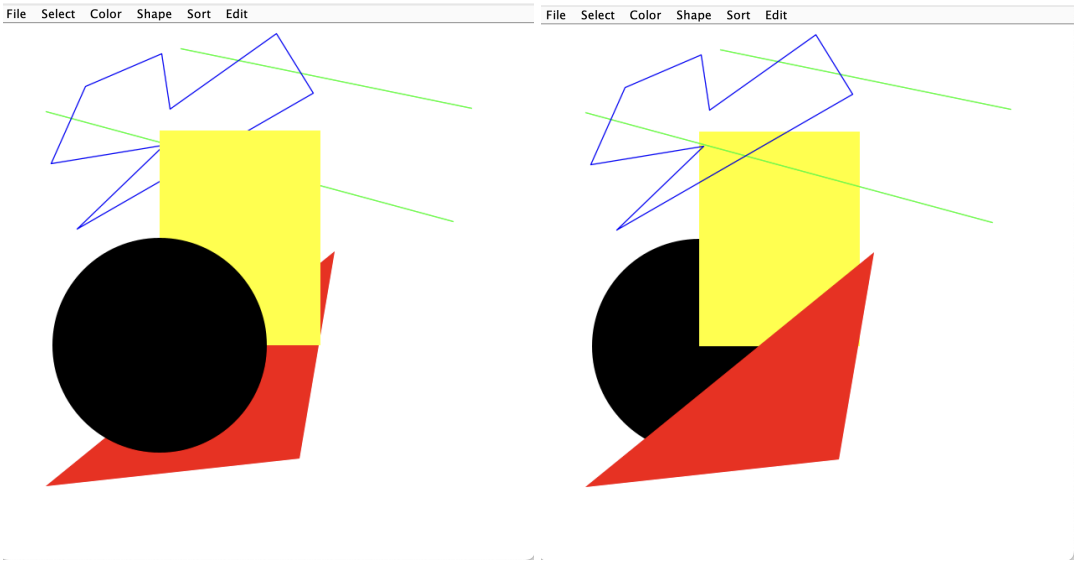


Figure 1: The “a0” file, as opened in the runnable jar. Left: ordered by “area” (black circle on top, i.e., largest). Right: sorted by “perimeter” (red triangle on top i.e., largest). Note: the green segment is “larger” than the yellow rectangle - as its parameter is twice as long as its length. Make sure you play with all the options - including the sorting, scaling, rotation, save & load.

Figure 2: file “a1” - rotation and scaling.

Links:

- Polygon: https://en.wikipedia.org/wiki/Polygon
- Testing if a polygon contains a point: https://en.wikipedia.org/wiki/Point_in_polygon

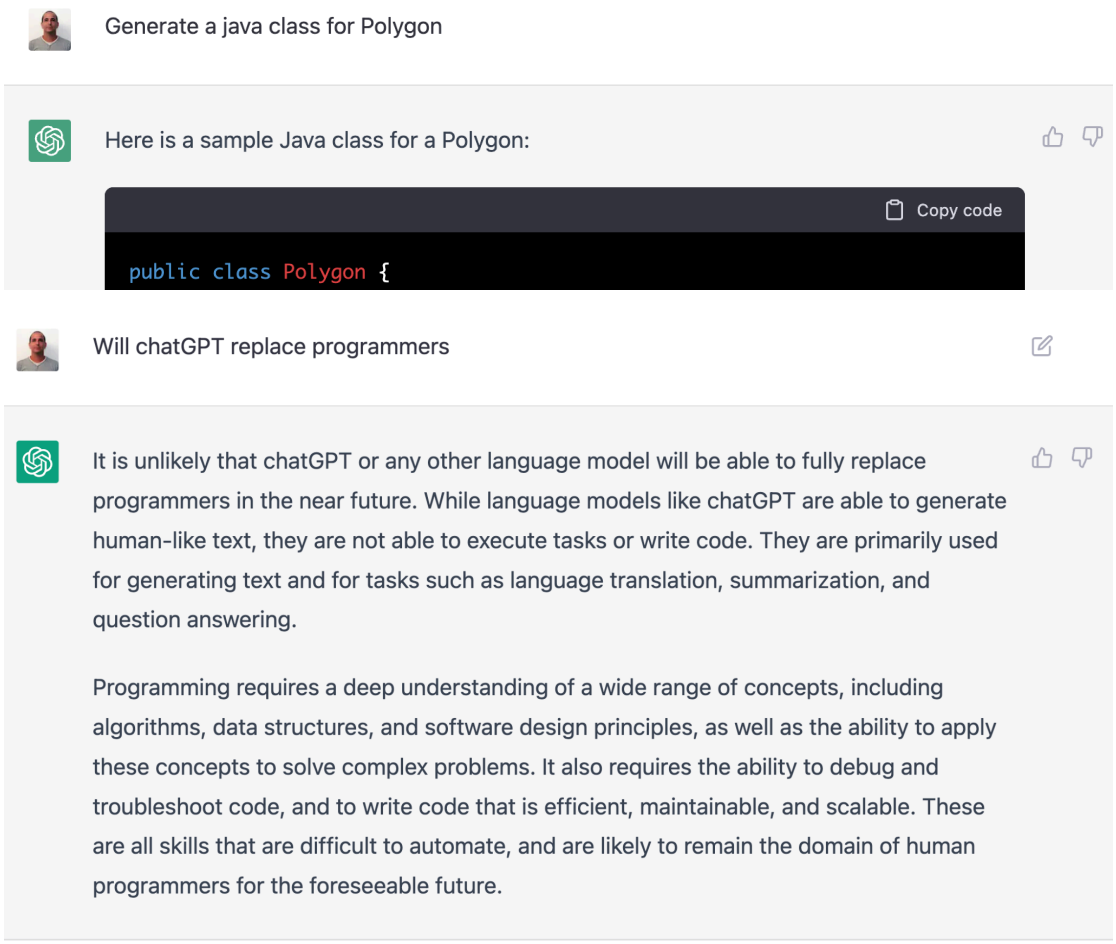


Figure 3: chatGPT is a source like any other - make sure to report on any source you are using! → Write your own code!

Q & A

1. Comparator:

```
if(_flag == Ex2_Const.Sort_By_toString) {
        ans = o1.toString().compareTo(o2.toString());
}
if(_flag == Ex2_Const.Sort_By_Area) {
        double d1 = o1.area();
        double d2 = o2.area();
        if(d1<d2) {ans  = -1;}
        if(d2<d1) {ans = 1;}
}
```

2. Rotate & Scale (center point)
   https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html#atan2-double-double-
3. Triangle area??
   A:
   https://he.wikipedia.org/wiki/%D7%A0%D7%95%D7%A1%D7%97%D7%AA_%D7%94%D7%A8%D7%95%D7%9F

4. I have an issue with the color encoding, can you elaborate on it:
   A: Sure, here is a short function that converts from a Color to an int with the same encoding as in the suggested solution:

```
public static int color encoding(Color c) {
   int r = c.getRed();
   int b = c.getBlue();
   int g = c.getGreen();
   int ce = r*256*256 + g*256 + b;
   return ce;
}
```

5. Do I need to implement (@Overide) the toString and equals methods of the classes I'm implementing?
   A: In general, Yes! this is a good practice and will allow you to use assertEquals. In short, make sure to implement the equals and toString methods both for the classes in the geo package and the containers (as well as the GUIShape).