Interface Exercise

Mohammad Faraz Idris Siddiqui

This is the GitHub link for all the codes:
https://github.com/feedme007/Software-Engineering-UPC-Codas/tree/main/Assignment-2(Interface)

The files **Person.java**, **Sorter.java**, **Program.java**, and **Rectangle.java** are the codes before implementing the interface and represent the solution for **Q.1–Q.4**.

The files **ComparableItem.java**, **Person2.java**, **Sorter2.java**, **Program2.java**, and **Rectangle2.java** are the updated codes after introducing the interface. These files (with the suffix "2" in their names) represent the solution for **Q.6**.

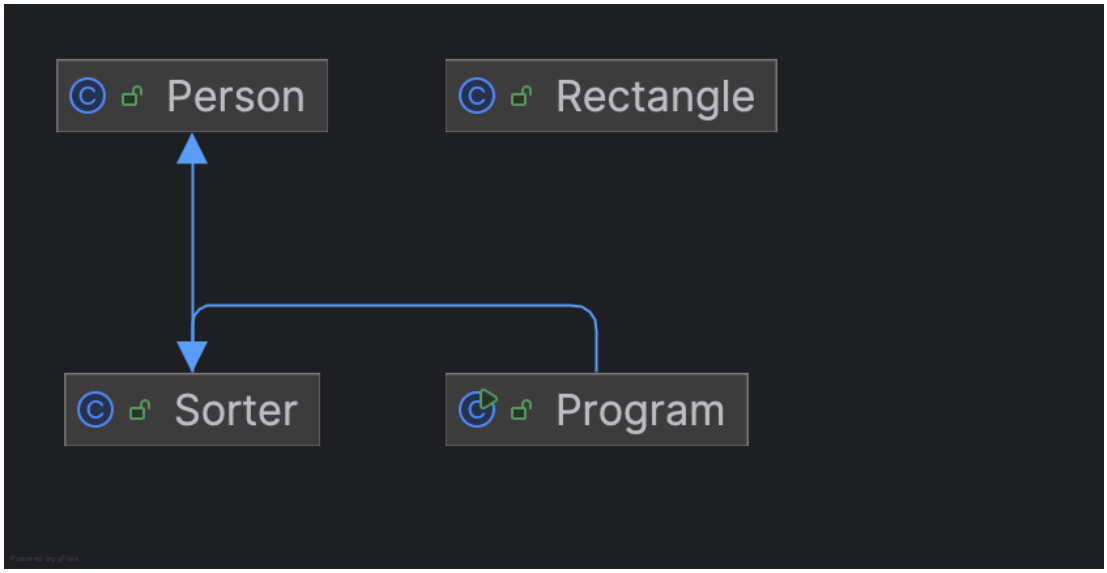
The diagrams for **Q.5** and **Q.6.c** are added below and were made using IntelliJ.

Q1–Q4.

Ans. For these questions (Person.java, Sorter.java, Program.java, Rectangle.java), please refer to the GitHub repository. The classes are implemented there and can be checked directly.

Q5. Draw a diagram of the current classes of the system

Ans. The class diagram of the current design: (made with IntelliJ)



Q6. Which modifications you have to do to make the Sorter class capable of sorting both Rectangles and Persons?
Ans. The modification in the **Sorter class** is that the method signature was changed from sort(Person[] people) to sort(ComparableItem[] arr). Inside the method, instead of directly comparing surnames and names, the comparison now uses the compareTo method of the interface. The bubble sort structure itself remains the same, only the type and comparison logic were generalized through the interface.

(ComparableItem.java, Person2.java, Sorter2.java, Program2.java, Rectangle2.java), please refer to the GitHub repository, to check these codes.

a) Rectangles must be sorted by its area
Ans. To extend sorting to rectangles, the comparison logic for rectangles was defined in terms of their area(). This way, when sorting, rectangles are ordered from the smallest to the largest area.

b) Which element of your design enables the class Sorter and its sort method to accept both Persons and Rectangles?
Ans. An interface called ComparableItem was introduced. Both Person2 and Rectangle2 implement this interface. Each class provides its own compareTo method:

- Person2 compares by surname, and if equal, by name.
- Rectangle2 compares by area.

This allows Sorter2 to work with any array of ComparableItem, meaning the same sort method can handle both persons and rectangles.

c) Draw a new diagram with the proposed changes before trying to implement it
Ans. The updated diagram (after interface) (made using IntelliJ)