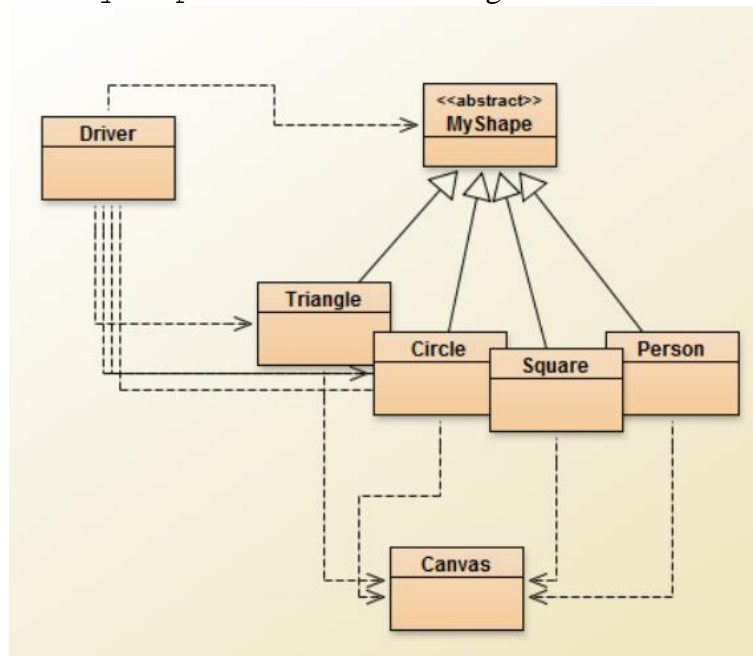


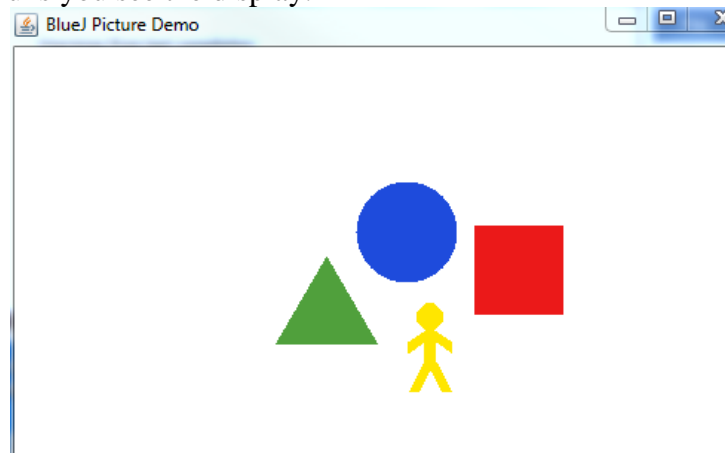
Hierarchies

1. Do exercise 1 on page 92:
“Develop another subclass of the Practitioner hierarchy (Dentist).
Demonstrate your new subclass with a class that instantiates Dentist Objects.”
2. Consider the code at <http://www.acs.uwinnipeg.ca/rmcfadyen/CreativeCommons/LabStart.zip>.

In this code you will see the Shapes example has been modified to incorporate a hierarchy where the root class is MyShape. The BlueJ class diagram:



When the Driver runs you see the display:



Continued on next page ...

Some points you must observe:

- The field `isVisible` has been moved from `Triangle`, `Square`, `Circle`, and `Person` into a superclass. Note that the field `isVisible` has the private modifier in the original code, but there is no modifier present in `MyShape`.
- The declarations for `isVisible` in the subclasses has been commented-out.
- The method `makeVisible()` has been moved from the subclasses into the superclass.
- The code for `makeVisible()` in the subclasses has been commented-out. The code was the same in all the subclasses.
- The superclass has an abstract method `draw()`. It is abstract because the definition of this method is defined in each subclass ... the subclasses have their own implementations.

Modify this shapes example a little bit further, as mentioned below:

- Move the `color` field from the subclasses to the superclass... similar to the way `isVisible` was moved.
- Move the `changeColor(...)` method from the subclasses to the superclass ... similar to the way `makeVisible()` was moved.

Run the Driver program to verify it works as before.

Submit **all** your `.java` files to your lab demonstrator.