Java Swing and Interfaces Practical example

CSCI 2300

What is true about Java Interface?

A. It can be used to define common variables and implement common methods of classes that inherit from it.
B. It cannot be instantiated
C. It defines method signatures that need to be implemented by classes implementing that interface
D. A, B, and C
E. B and C

Pre-defined Interfaces in Java

- Comparable<T>

Modifier and Type	Method and Description
int	compareTo(T o) Compares this object with the specified object for order.

- Collections class has several static methods
  - Example: sort
- We can use Collections.sort to sort Comparable objects
- Example: Comparable Point2D (link on course schedule)

Icon Interface

Modifier and Type	Method and Description
int	getIconHeight() Returns the icon's height.
int	getIconWidth() Returns the icon's width.
void	paintIcon(Component c, Graphics g, int x, int y) Draw the icon at the specified location.

x and y are coordinates of the top left corner of the area to paint

Icon can be used in the JLabel constructor to show an image

Snowman1 example (link on course schedule)

Dynamically Changing the image

- Icon::paintIcon() gets called whenever JLabel::repaint() is called
- We can use this to create animation effects
- Example: when button is clicked, let's change the color of the snowman.
  - Add a list of pre-defined colors in the Snowman class.
  - Pick the 'next' color in paintIcon()
  - Add a button with action listener

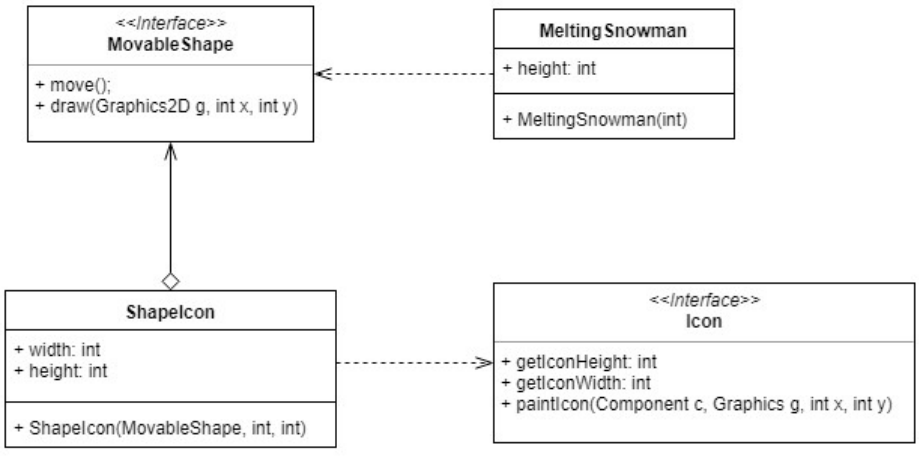
What is the best way to show a completely new image in JLabel when button is clicked?

A. Use JLabel.setIcon() method to change the Icon being dispalyed
B. B. Use paintIcon() method to change the image
C. Create new JLabel with new Icon when button is clicked

Moving Shape

- We want to add some “animation” to our application
- Let’s melt that snowman
- Design changes:
  - Introduce MovableShape interface
  - Introduce “generic” ShapeIcon class that “has-a” MovableShape

UML Diagram



Creating an Animation

- Application sketch:
  - MeltingSnowman snowman = new MeltingSnowman…
  - ShapeIcon icon = newShapeIcon(snowman,…)
  - Jlabel label = new Jlabel(icon)
  - Add the label to the frame
  - Create a "timer" with ActionListener
  - actionPerformed calls snowman.move() and label.repaint()
    - label.repaint() will call icon.paintIcon()
    - icon.paintIcon() will call MovableShape's draw() method
- Snowman 3 (link on course schedule)

Car Animation

- Same design as SnowmanAnimation
- Use Car.java class instead of MeltingSnowman.java
- **`public class Car implements MovableShape`**
- **`{…}`**