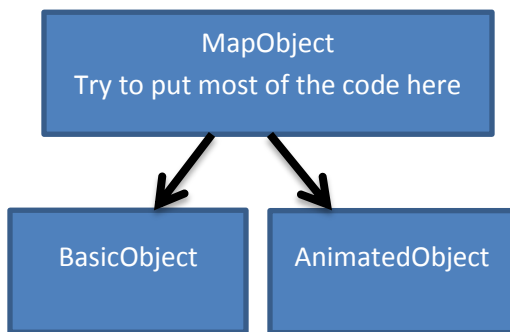


Programming:

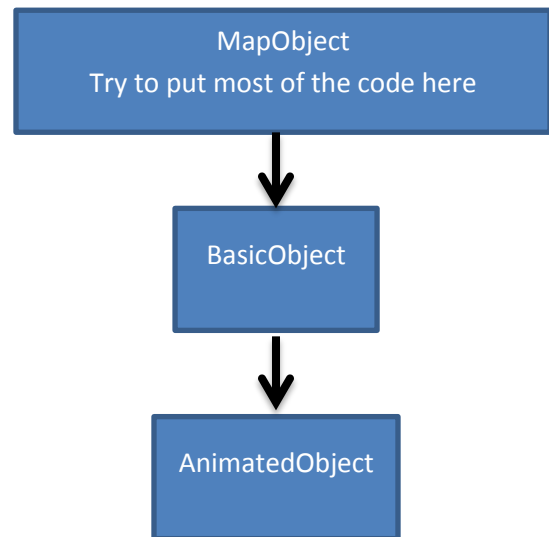
For this assignment you will be creating 3 java classes: MapObject, BasicObject, and AnimatedObject. The first is a top level class called MapObject and the other 2 will be extensions of the MapObject class using either Option 1 or Option 2 below.

Your goal is to try to avoid duplicating code using Inheritance.

Option 1



Option 2



MapObject

Although the only real method a MapObject needs to have is a draw method, your goal should be to move as much of the methods as you can from the BasicObject and AnimatedObject class into the MapObject class to avoid duplicating code

- An abstract method `void draw(Graphics2D g);`
 - Graphics2D is imported from `java.awt`.

BasicObject

A basic object needs to remember the following information:

- An x and a y coordinate
- A boolean that determines if the object is solid or can be walked through
- A BufferedImage that is imported from java.awt.image.

A basic object can be constructed as follows: Remember, with inheritance you need to make a call to the super classes constructor using the word super (*parameters here*) if you want it to setup your variables.

- A 3 parameter constructor
 - (int x, int y, boolean b)
 - You set the other variables
- A 4 parameter constructor
 - (int x, int y, boolean b, BufferedImage i)

A basic object has the following methods

- public int getX()
- public void setX(int x)
- public int getY()
- public void setY(int y)
- public boolean getSolidState()
- public void setSolidState(boolean b)
- public void moveTo(int x, int y)
 - which is basically a setter for 2 variables
- public BufferedImage getCurrentImage()
- public void setCurrentImage(BufferedImage i)

You also need to override the public void draw method from the parent class and make it work

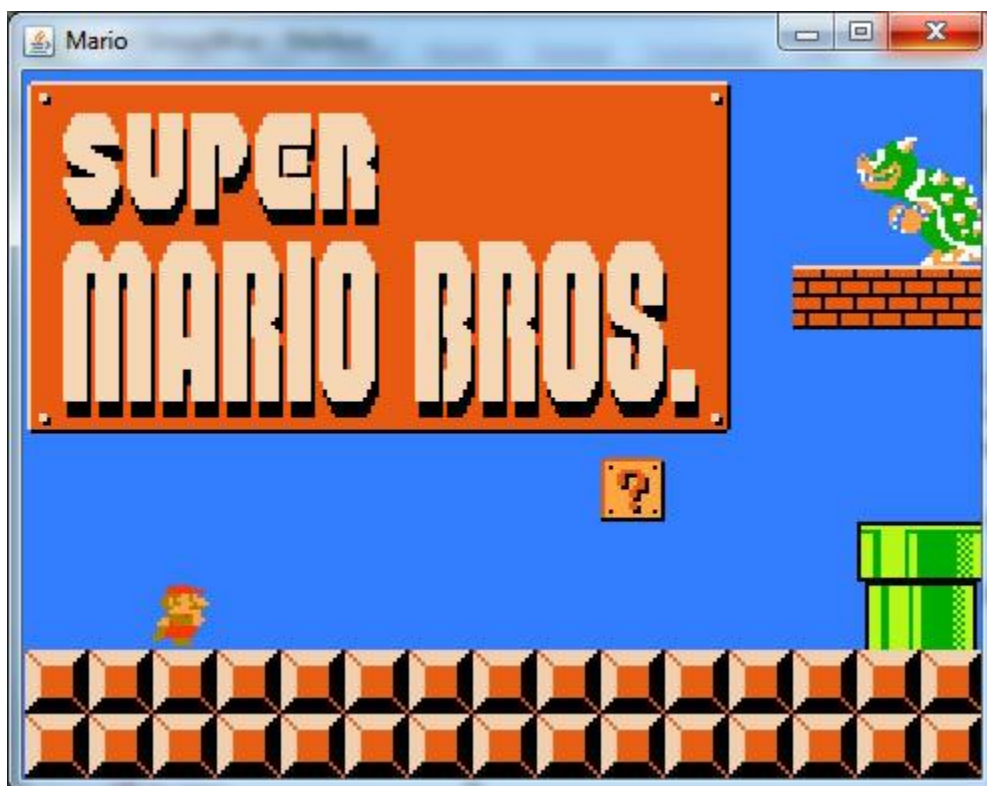
- public void draw(Graphics2D g)
 - use the following method call, replacing the parameters as needed
 - g.drawImage(BufferedImage, int, int, null)
 - where g is your Graphics2D parameter name
 - BufferedImage is the image you are storing in your instance variables
 - the ints are the location, you may need to use a super class method to get them
 - just leave the word null where it is.

AnimatedObject

This class needs to do EVERYTHING that a BasicObject does except

- Instead of remembering a single buffered image, it should remember an array of buffered images.
- The 4 parameter constructor should be changed to
 - (int x, int y, boolean b, BufferedImage [] a)
- You may need to look at the following 2 methods, and they may need to do something slightly different than the BasicObject methods depending on your code
 - public BufferedImage getCurrentImage()
 - public void setCurrentImage(BufferedImage i)
- make the public void draw(Graphics2D g) method work.
 - Each time the draw method is called, it draws the next image in the array. So, you need to somehow remember (between method calls) which array spot you just drew and then increment it.
 - It should start over at the beginning of the array once every image in the array has been drawn.
 - Use the same method call as you do in the BasicObject method, but use a location of your array, for example arraySpot[currentSpot] in the BufferedImage place

If you place your files into the InheritanceFiles project, remove the comments that comment out most of the CreatePicture file, and remove the comments in the main you should get a picture similar to the following



Submissions:

Upload and submit the following java files you created for this assignment

- MapObject.java
- BasicObject.java
- AnimatedObject.java