LAB 3

 Feb 19, 2021

Ayush Shresth ( Scriber)

[Nachiketh Mamidi](mailto:nachiketh.mamidi@sjsu.edu)( Driver)

PART A

1. No-step

/\*\*

     \* Moves the robot to right

     \*

     \*/

    public void moveRight()

    {

    }

1. No-step
2. No-step
3. No-step

/\*\*

     \* Constructor for objects of class Robot

     \*

     \* @param theX displays the x-coordinate of the upper-left corner of the robot

     \* @param theY displays the y-coordinate of the upper-left corner of the robot

     \* @param pic displays the image of the robot

     \*/

    public Robot(int theX, int theY)

    {

        x = theX;

        y = theY;

        pic = new Picture("myrobot.jpeg");

    }

1. We initialize the values for X and Y, we get an error and the picture is not displayed
2. pic.draw();
3. pic.translate(pic.getWidth(), 0);
4. When we click on moveRight on the red blob the Robot moves right.

PART B

1.

    /\*\*

     \* Moves the robot to right

     \*

     \*/

    public void moveDown()

    {

        pic.translate(0, pic.getHeight());

    }

/\*\*

     \* Moves the robot to right

     \*

     \*

     \*/

    public void moveDown()

    {

        pic.translate(0,-pic.getHeight());

    }

2. We compile a test run

3.

    /\*\*

     \* Moves robot horizontally by a multiple of its width

     \*

     \* @param steps Movement distance by the times of its width

     \*

     \*/

4.

   public void moveHorizontally(int steps)

     {

        pic.translate(steps\*pic.getWidth(), 0);

     }

5. We compile our Robot class and in the red blob we put the move horizontal to 5; The robot moves right by 5 units of its width. Then we put the move horizontal to -3; The robot moves 3 units of its width left.. So it is now at 2 units of its width from right.

6. We put the move horizontal to -3 units. So the Robot disappears.

7.

/\*\*

     \* Moves robot vertically by a multiple of its height

     \*

     \* @param steps Movement distance by the times of its height

     \*

     \*/

    public void moveVertically(int steps)

    {

        pic.translate(0, steps\*pic.getHeight());

    }

8. We first move horizontally, the robot 5 units of its width size to its right and then it goes 5 units of its height downwards from that point.

9. To bring back the robot to its original position, we move the robot horizontally by 5  units of its width and then we move the robot -5 units vertically.

PART C

1. This method is an accessor. Because it does not change the value of the object.
2. We expected the X coordinate of the top left corner of the robot image.
3. We get the original position of the robot (1) when we run the getX method.
4. I added x = steps\*pic.getWidth();
5. We get the value -306
6. It is mutator method, We change the method to x = x + steps;

    public int getY()

     {

        return y;

     }

1. We compile it and run it

PART D

1.

/\*\*

 \* Write a description of class MoveTester here.

 \*To test the methods and accessor of class Robot

 \* @author Nachiketh Mamidi

 \* @version Feb 19, 2021

 \*/

public class MoveTester

{

    public static void main()

    {

       Robot one = new Robot (0,0);

       one.moveHorizontally(5);

       one.moveHorizontally(-3);

       System.out.println(one.getX());

       System.out.println("Expected: 2");

    }

}

2. When we run the program we get 2 and in the next line it is expected: 2

3.

       one.moveVertically(6);

       one.moveVertically(-4);

       System.out.println(one.getY());

       System.out.println("Expected: 2");

4. We get 2 and expected: 2 , again

5. No, they aren’t increasing by the same amount

6. We can get the X value of the robot by dividing it by pic.getWidth()

7. We get the same result as return x;

8.  return pic.getY() / pic.getHeight();

9. We are using getX and getY in our return statement so X and Y are useful.

PART E

1. We use move horizontally method to moveRight
2. We need to call the moveHorizontally method on pic object.
3. No-Step
4. We need to moveRight to the moveTester class to check if it works

           one.moveRight();

           System.out.println(one.getX());

           System.out.println("Expected: 3");

1. The other way is pic.translate(steps\*pic.getWidth(), y);
2. Yes
3. I like moveVertically(1); more because we are just using the method we already made.

public void moveDown()

    {

        moveVertically(1);

    }