Homework 3

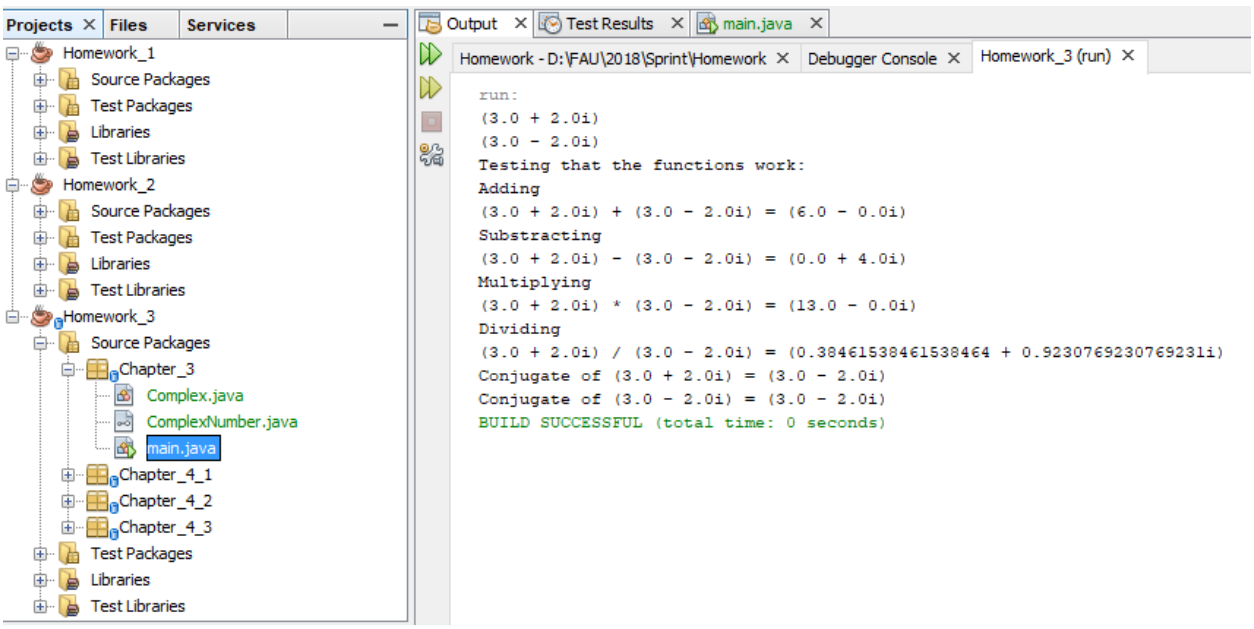
3.1

A. Encapsulation is an important principle of OODP because it hides the data implementation by restricting access to accessors and mutators. This allows for code that is reusable and maintainable for both people using the code and designers.

B. It is okay to throw exceptions as part of the contract whenever the precondition fails or is not fulfilled. One example can be removing an elements at the head of a linked list when the list is empty or set to null.

C. Side effects should be avoided because they can cause unnecessary changes to an object and or its state that you may or may not be aware of.

3.2



```java
/**
 *
 * @author Carlos Guisao
 */
public class main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Complex variable_1 = new Complex(3, 2),
                variable_2 = new Complex(3, -2);

        System.out.println(variable_1.toString());
        System.out.println(variable_2.toString());

        Complex sum         = variable_1.add(variable_2),
                difference  = variable_1.sub(variable_2),
                multiplying = variable_1.mult(variable_2),
                Dividing    = variable_1.div(variable_2),
                conjugate1  = variable_1.conj(),
                conjugate2  = variable_1.conj();

        System.out.println("Testing that the functions work:");
        System.out.println("Adding");
        System.out.println(variable_1.toString() + " + " + variable_2.toString() + " = " +  sum.toString());

        System.out.println("Substracting");
        System.out.println(variable_1.toString() + " - " + variable_2.toString() + " = " +  difference.toString());

        System.out.println("Multiplying");
        System.out.println(variable_1.toString() + " * " + variable_2.toString() + " = " +  multiplying.toString());

        System.out.println("Dividing");
        System.out.println(variable_1.toString() + " / " + variable_2.toString() + " = " +  Dividing.toString());

        System.out.println("Conjugate of " + variable_1.toString() + " = " + conjugate1.toString());
        System.out.println("Conjugate of " + variable_2.toString() + " = " + conjugate2.toString());

    }
}
```

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Chapter_3;

/**
 *
 * @author Carlos Guisao
 */
public interface ComplexNumber {
    double r();
    double i();
    Complex add(Complex c);
    Complex sub(Complex c);
    Complex conj();
    Complex mult(Complex c);
    Complex div(Complex c);
    boolean equals(Complex c);
}
```

Complex Class

Projects | Files | Services

- Homework_1
- Homework_2
- Homework_3
  - Source Packages
    - Chapter_3
      - Complex.java
      - ComplexNumber.java
      - main.java
    - Chapter_4_1
    - Chapter_4_2
    - Chapter_4_3
  - Test Packages
  - Libraries
  - Test Libraries

Complex - Navigator

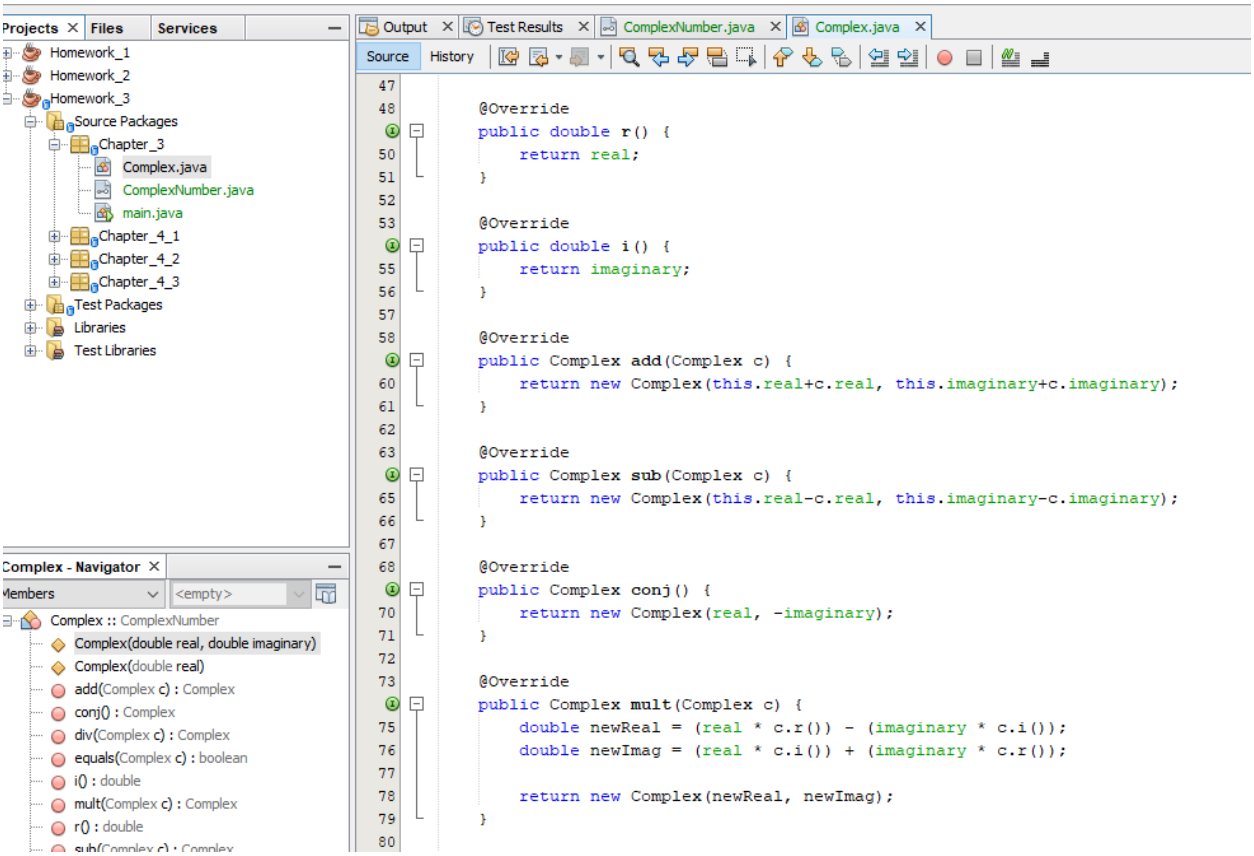
Members | <empty>

- Complex :: ComplexNumber
  - Complex(double real, double imaginary)
  - Complex(double real)
  - add(Complex c) : Complex
  - conj() : Complex
  - div(Complex c) : Complex
  - equals(Complex c) : boolean
  - i() : double
  - mult(Complex c) : Complex
  - r() : double
  - sub(Complex c) : Complex
  - toString() : String ↑ Object
  - imaginary : double
  - real : double

Output | Test Results | ComplexNumber.java | Complex.java

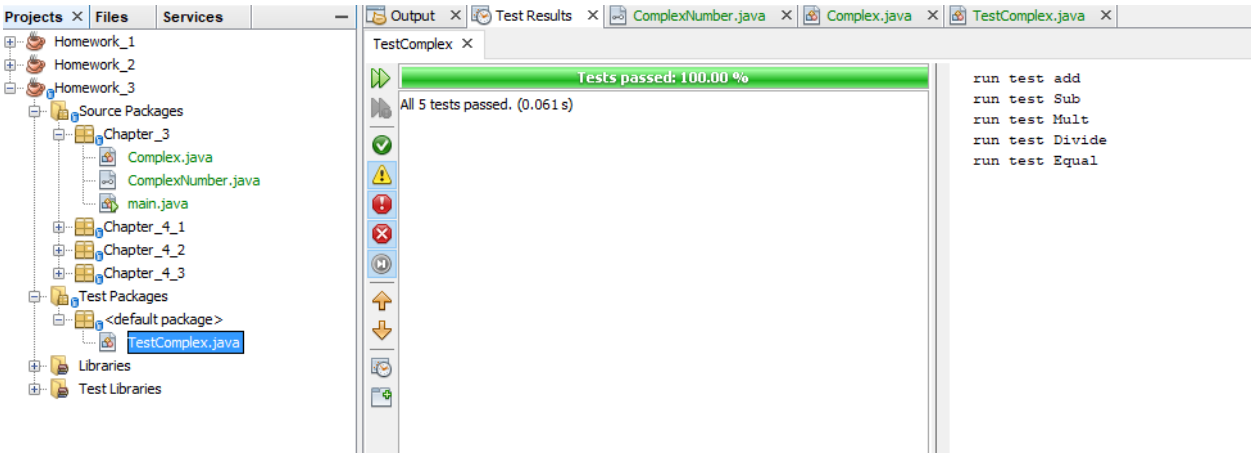
Source | History

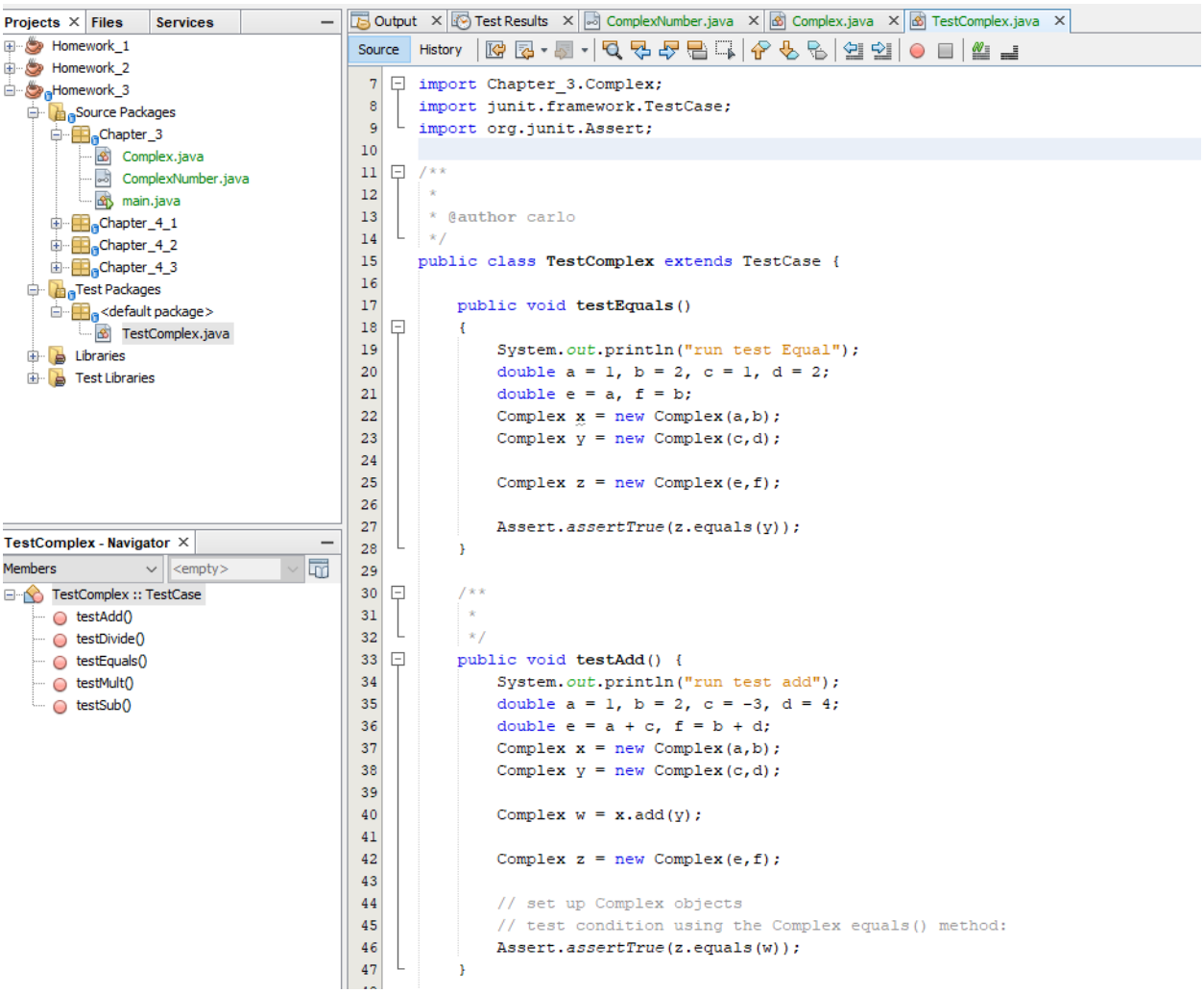
```java
/**
 *
 * @author Carlos Guisao
 */
public class Complex implements ComplexNumber {

    private final double real, imaginary;

    public Complex(double real, double imaginary)
    {
        this.real = real;
        this.imaginary = imaginary;
    }

    public Complex(double real)
    {
        this.real = real;
        this.imaginary = 0.0;
    }

    /**
     *
     * @return a complex number using the real and imaginary number
     */
    @Override
    public String toString()
    {
        double currentReal = this.real;
        double currentImaginary = this.imaginary;
        String sign;

        if(currentImaginary > 0.0)
            sign = " + ";
        else
            sign = " - ";

        return "(" + Double.toString(currentReal) + sign +
                Double.toString(Math.abs(currentImaginary)) + "i" + ")";
    }
```

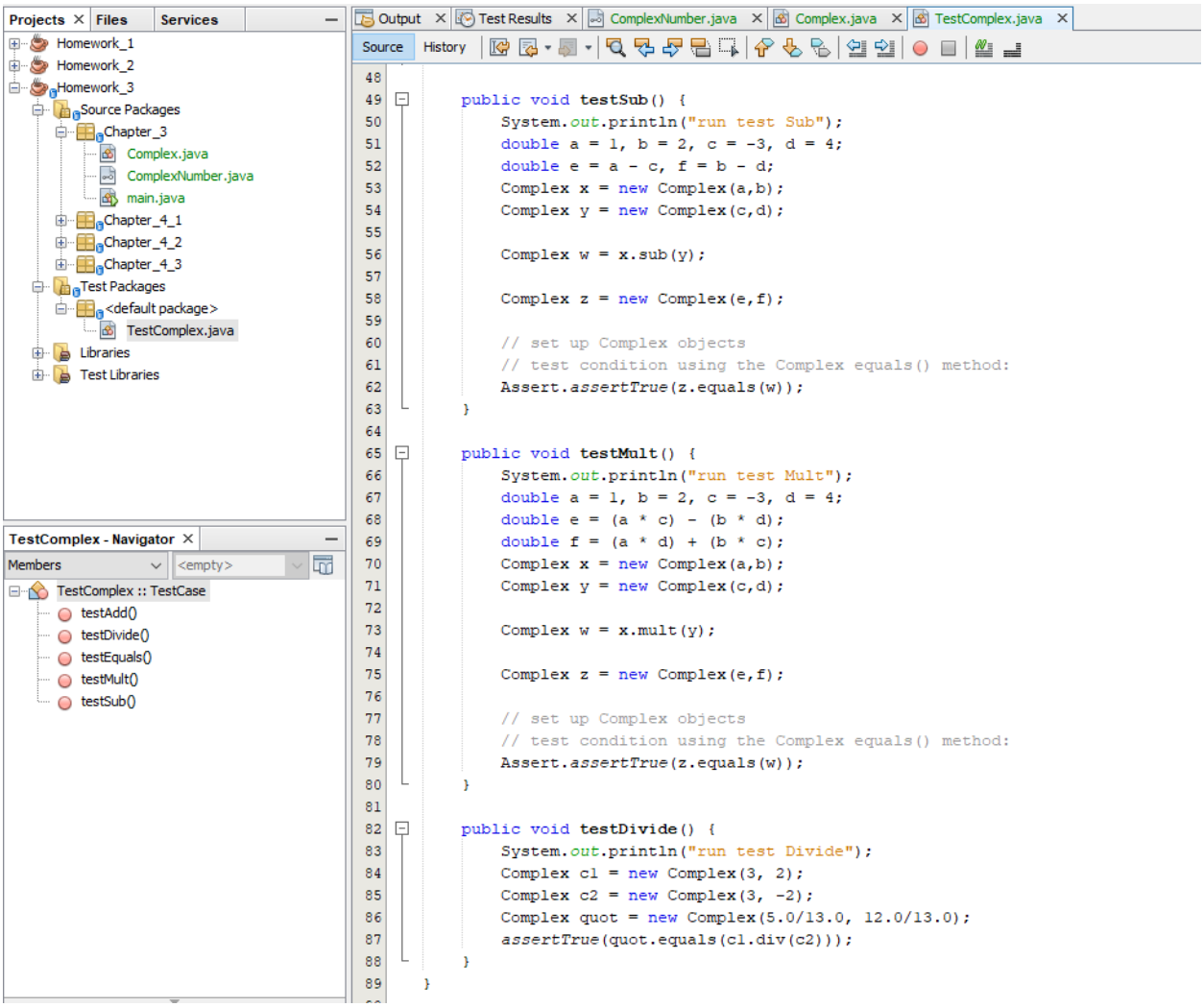



```java
    @Override
    public Complex div(Complex c) {
        if (c.r() == 0.0 && c.i() == 0.0) {
            throw new IllegalArgumentException("Quit dividing by zero.");
        }
        Complex denominatorConjugate = c.conj();
        Complex numerator = mult(denominatorConjugate);
        Complex denominator = c.mult(denominatorConjugate);

        double newReal = numerator.r() / denominator.r();
        double newImag = numerator.i() / denominator.r();

        return new Complex(newReal, newImag);
    }

    @Override
    public boolean equals(Complex c) {
        return real == c.real && imaginary == c.imaginary;
    }

}
```

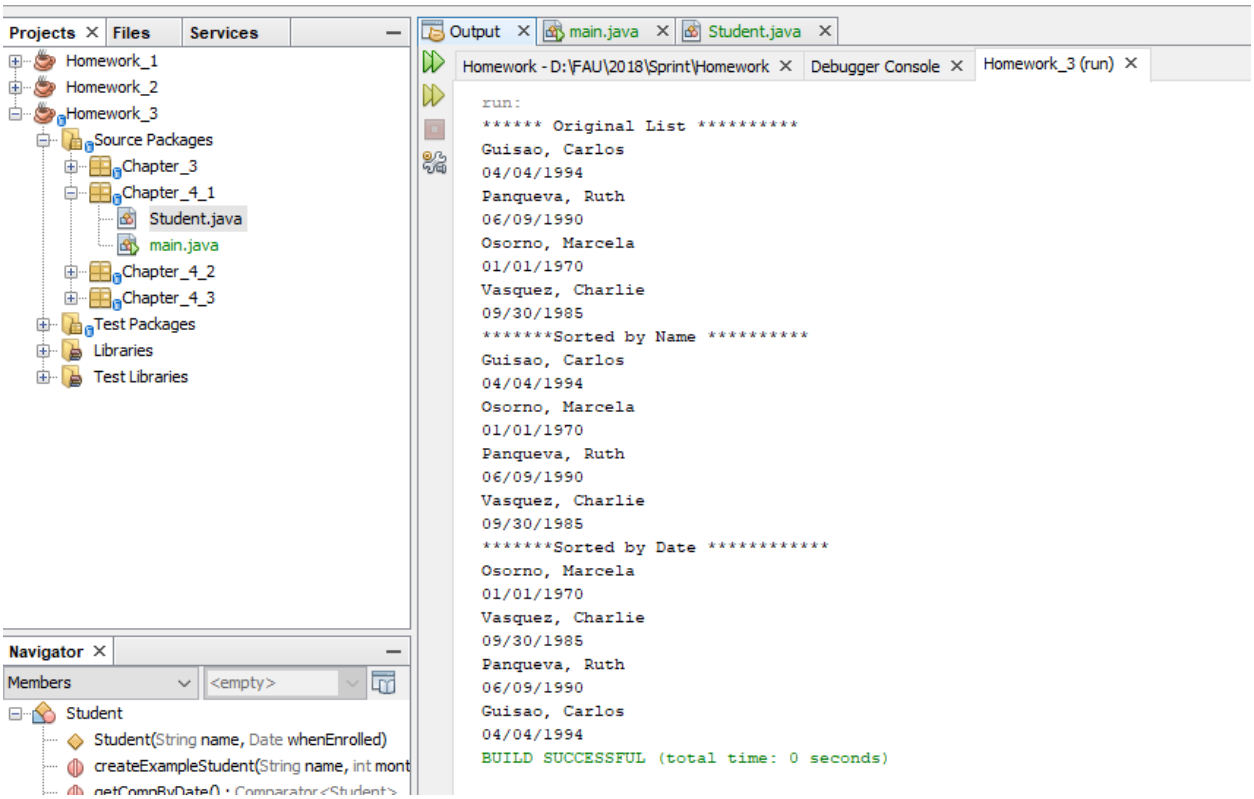
Unite test


Projects
Files
Services
Homework_1
Homework_2
Homework_3
Source Packages
Chapter_3
Complex.java
ComplexNumber.java
main.java
Chapter_4_1
Chapter_4_2
Chapter_4_3
Test Packages
<default package>
TestComplex.java
Libraries
Test Libraries
Output
Test Results
ComplexNumber.java
Complex.java
TestComplex.java
TestComplex
Tests passed: 100.00 %
All 5 tests passed. (0.061 s)
run test add
run test Sub
run test Mult
run test Divide
run test Equal

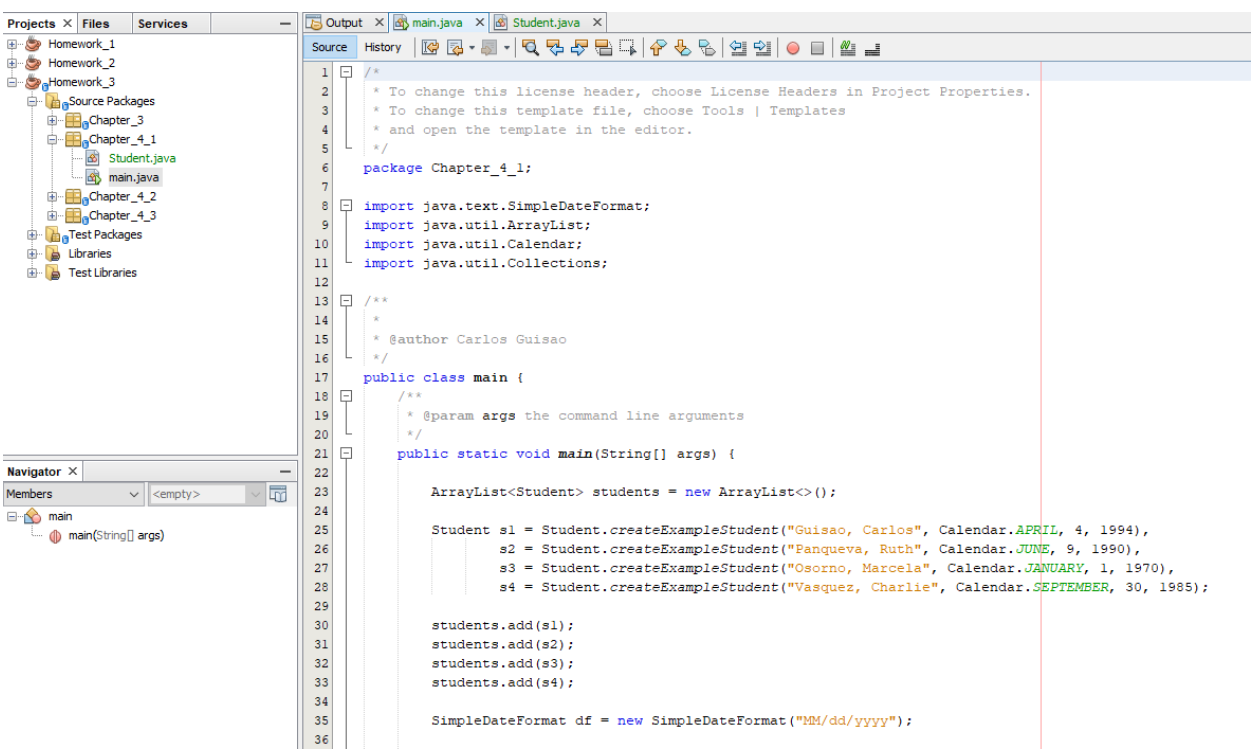

Projects
Files
Services
Homework_1
Homework_2
Homework_3
Source Packages
Chapter_3
Complex.java
ComplexNumber.java
main.java
Chapter_4_1
Chapter_4_2
Chapter_4_3
Test Packages
<default package>
TestComplex.java
Libraries
Test Libraries
TestComplex - Navigator
Members
<empty>
TestComplex :: TestCase
testAdd()
testDivide()
testEquals()
testMult()
testSub()
Output
Test Results
ComplexNumber.java
Complex.java
TestComplex.java
Source
History
7 import Chapter_3.Complex;
8 import junit.framework.TestCase;
9 import org.junit.Assert;
10
11 /**
12 *
13 * @author carlo
14 */
15 public class TestComplex extends TestCase {
16
17 public void testEquals()
18 {
19 System.out.println("run test Equal");
20 double a = 1, b = 2, c = 1, d = 2;
21 double e = a, f = b;
22 Complex x = new Complex(a,b);
23 Complex y = new Complex(c,d);
24
25 Complex z = new Complex(e,f);
26
27 Assert.assertTrue(z.equals(y));
28 }
29
30 /**
31 *
32 */
33 public void testAdd() {
34 System.out.println("run test add");
35 double a = 1, b = 2, c = -3, d = 4;
36 double e = a + c, f = b + d;
37 Complex x = new Complex(a,b);
38 Complex y = new Complex(c,d);
39
40 Complex w = x.add(y);
41
42 Complex z = new Complex(e,f);
43
44 // set up Complex objects
45 // test condition using the Complex equals() method:
46 Assert.assertTrue(z.equals(w));
47 }



4.1 Student Class

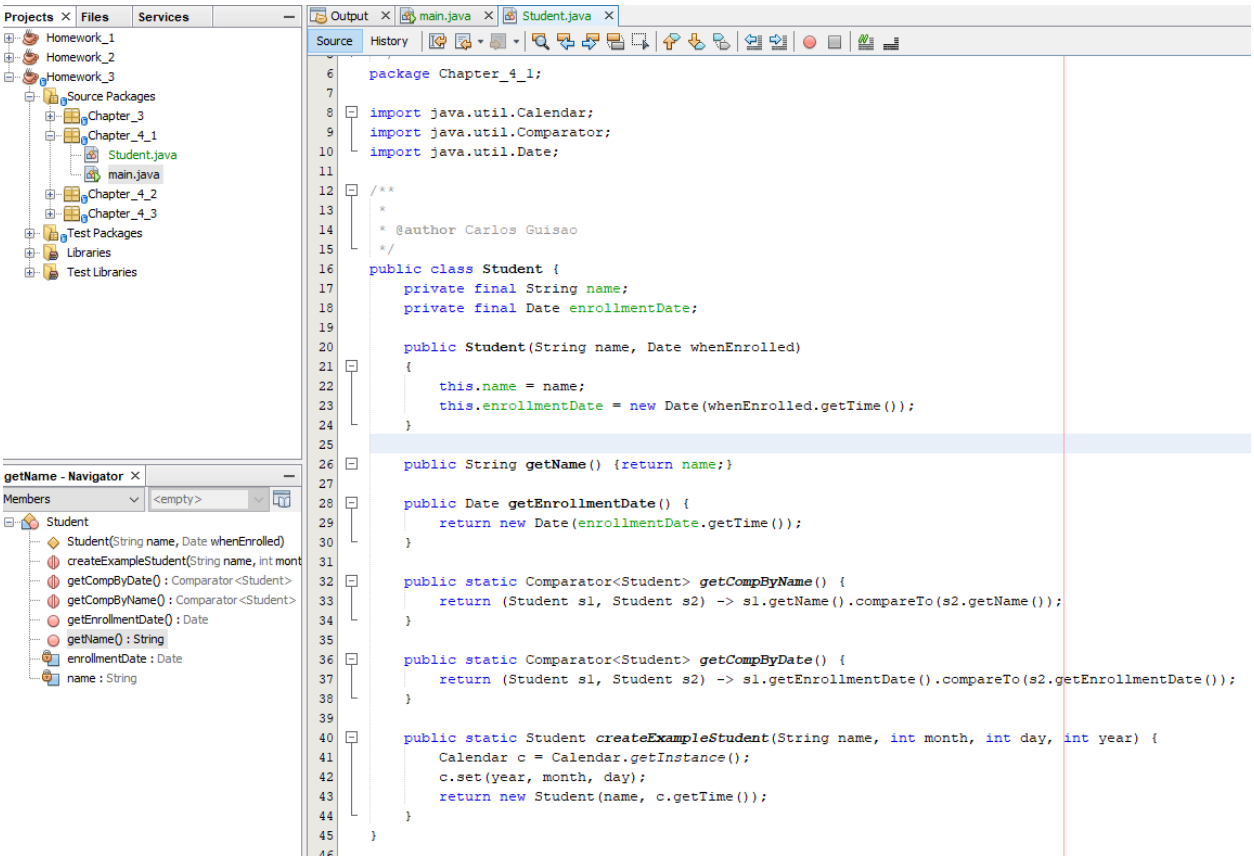

run:
****** Original List **********
Guisao, Carlos
04/04/1994
Panqueva, Ruth
06/09/1990
Osorno, Marcela
01/01/1970
Vasquez, Charlie
09/30/1985
*******Sorted by Name **********
Guisao, Carlos
04/04/1994
Osorno, Marcela
01/01/1970
Panqueva, Ruth
06/09/1990
Vasquez, Charlie
09/30/1985
*******Sorted by Date ************
Osorno, Marcela
01/01/1970
Vasquez, Charlie
09/30/1985
Panqueva, Ruth
06/09/1990
Guisao, Carlos
04/04/1994
BUILD SUCCESSFUL (total time: 0 seconds)

Main Class

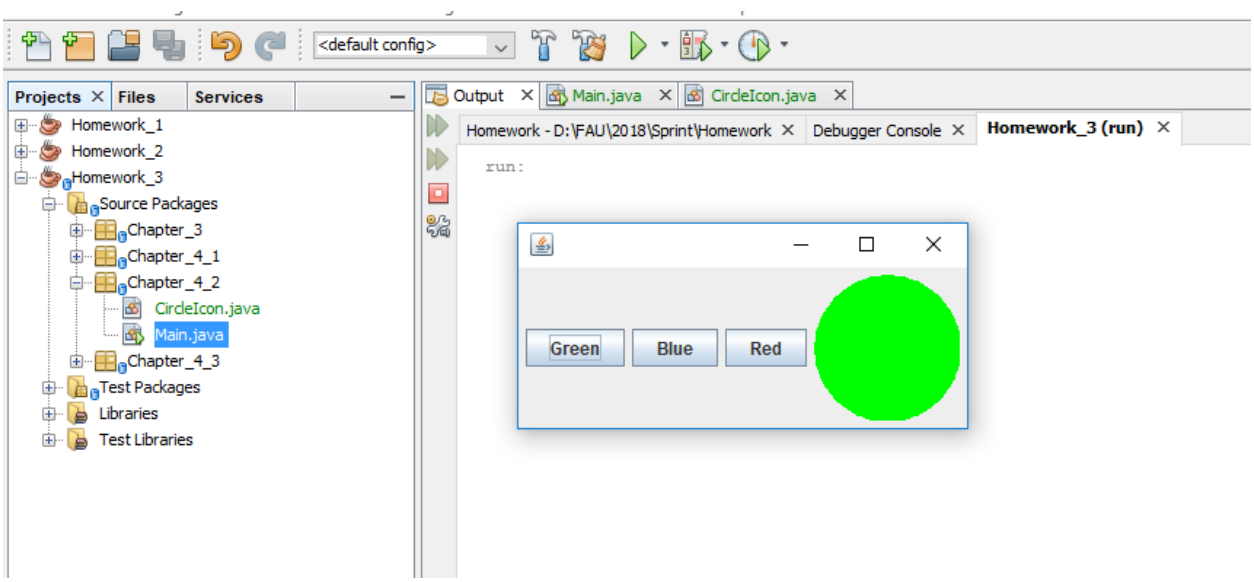

/*
* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
package Chapter_4_1;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collections;
/**
*
* @author Carlos Guisao
*/
public class main {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
ArrayList<Student> students = new ArrayList<>();
Student s1 = Student.createExampleStudent("Guisao, Carlos", Calendar.APRIL, 4, 1994),
s2 = Student.createExampleStudent("Panqueva, Ruth", Calendar.JUNE, 9, 1990),
s3 = Student.createExampleStudent("Osorno, Marcela", Calendar.JANUARY, 1, 1970),
s4 = Student.createExampleStudent("Vasquez, Charlie", Calendar.SEPTEMBER, 30, 1985);
students.add(s1);
students.add(s2);
students.add(s3);
students.add(s4);
SimpleDateFormat df = new SimpleDateFormat("MM/dd/yyyy");

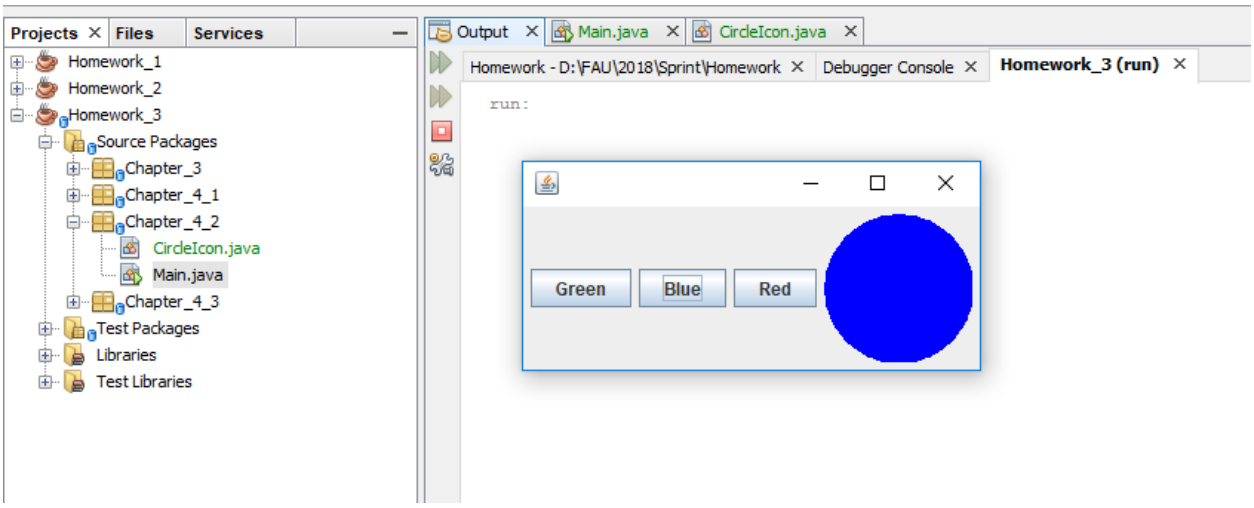
```java
        SimpleDateFormat df = new SimpleDateFormat("MM/dd/yyyy");

        System.out.println("****** Original List **********");
        students.stream().map((s) -> {
            System.out.println(s.getName());
            return s;
        }).forEachOrdered((s) -> {
            System.out.println(df.format(s.getEnrollmentDate()));
        });

        System.out.println("*******Sorted by Name **********");
        Collections.sort(students, Student.getCompByName());
        students.stream().map((s) -> {
            System.out.println(s.getName());
            return s;
        }).forEachOrdered((s) -> {
            System.out.println(df.format(s.getEnrollmentDate()));
        });

        System.out.println("*******Sorted by Date ************");
        Collections.sort(students, Student.getCompByDate());
        students.stream().map((s) -> {
            System.out.println(s.getName());
            return s;
        }).forEachOrdered((s) -> {
            System.out.println(df.format(s.getEnrollmentDate()));
        });
    }
}
```

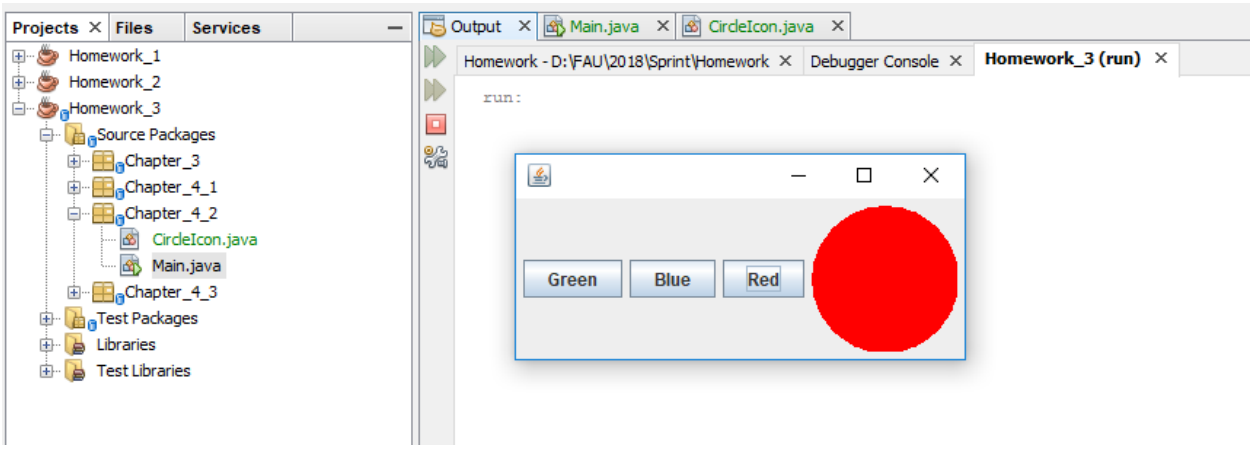
Student Class


Projects
Files
Services
Homework_1
Homework_2
Homework_3
Source Packages
Chapter_3
Chapter_4_1
Student.java
main.java
Chapter_4_2
Chapter_4_3
Test Packages
Libraries
Test Libraries
Output
main.java
Student.java
Source
History
package Chapter_4_1;
import java.util.Calendar;
import java.util.Comparator;
import java.util.Date;
/**
*
* @author Carlos Guisao
*/
public class Student {
private final String name;
private final Date enrollmentDate;
public Student(String name, Date whenEnrolled)
{
this.name = name;
this.enrollmentDate = new Date(whenEnrolled.getTime());
}
public String getName() {return name;}
public Date getEnrollmentDate() {
return new Date(enrollmentDate.getTime());
}
public static Comparator<Student> getCompByName() {
return (Student s1, Student s2) -> s1.getName().compareTo(s2.getName());
}
public static Comparator<Student> getCompByDate() {
return (Student s1, Student s2) -> s1.getEnrollmentDate().compareTo(s2.getEnrollmentDate());
}
public static Student createExampleStudent(String name, int month, int day, int year) {
Calendar c = Calendar.getInstance();
c.set(year, month, day);
return new Student(name, c.getTime());
}
}
getName - Navigator
Members
<empty>
Student
Student(String name, Date whenEnrolled)
createExampleStudent(String name, int mont
getCompByDate() : Comparator<Student>
getCompByName() : Comparator<Student>
getEnrollmentDate() : Date
getName() : String
enrollmentDate : Date
name : String

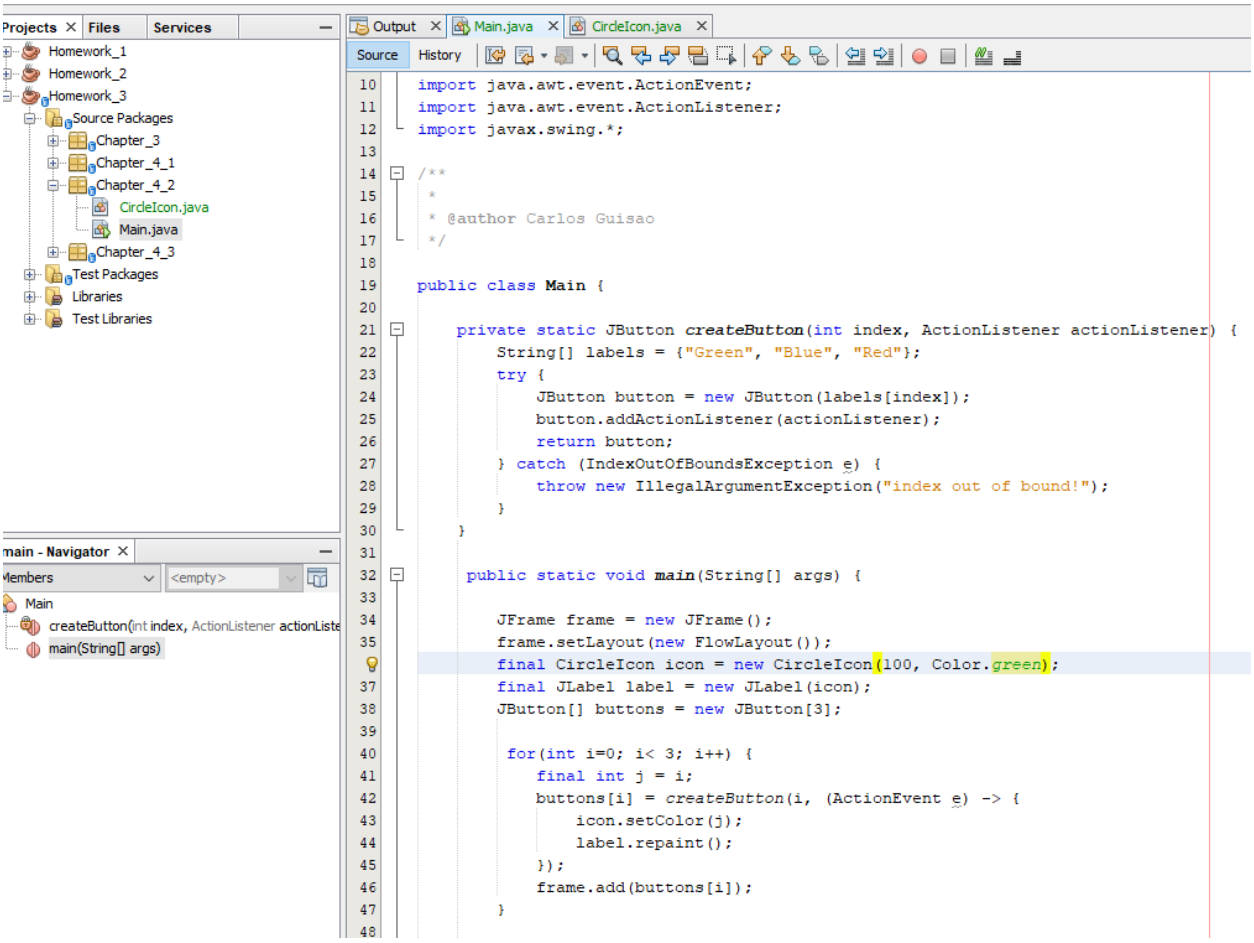
4.2


<default config>
Projects
Files
Services
Homework_1
Homework_2
Homework_3
Source Packages
Chapter_3
Chapter_4_1
Chapter_4_2
CircleIcon.java
Main.java
Chapter_4_3
Test Packages
Libraries
Test Libraries
Output
Main.java
CircleIcon.java
Homework - D:\FAU\2018\Sprint\Homework
Debugger Console
Homework_3 (run)
run:
Green
Blue
Red


Projects
Files
Services
Homework_1
Homework_2
Homework_3
Source Packages
Chapter_3
Chapter_4_1
Chapter_4_2
CircleIcon.java
Main.java
Chapter_4_3
Test Packages
Libraries
Test Libraries
Output
Main.java
CircleIcon.java
Homework - D:\FAU\2018\Sprint\Homework
Debugger Console
Homework_3 (run)
run:
Green
Blue
Red


Projects
Files
Services
Homework_1
Homework_2
Homework_3
Source Packages
Chapter_3
Chapter_4_1
Chapter_4_2
CircleIcon.java
Main.java
Chapter_4_3
Test Packages
Libraries
Test Libraries
Output
Main.java
CircleIcon.java
Homework - D:\FAU\2018\Sprint\Homework
Debugger Console
Homework_3 (run)
run:
Green
Blue
Red

Main




```java
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

/**
 *
 * @author Carlos Guisao
 */

public class Main {

    private static JButton createButton(int index, ActionListener actionListener) {
        String[] labels = {"Green", "Blue", "Red"};
        try {
            JButton button = new JButton(labels[index]);
            button.addActionListener(actionListener);
            return button;
        } catch (IndexOutOfBoundsException e) {
            throw new IllegalArgumentException("index out of bound!");
        }
    }

     public static void main(String[] args) {

        JFrame frame = new JFrame();
        frame.setLayout(new FlowLayout());
        final CircleIcon icon = new CircleIcon(100, Color.green);
        final JLabel label = new JLabel(icon);
        JButton[] buttons = new JButton[3];

         for(int i=0; i< 3; i++) {
            final int j = i;
            buttons[i] = createButton(i, (ActionEvent e) -> {
                icon.setColor(j);
                label.repaint();
            });
            frame.add(buttons[i]);
        }

        Component add = frame.add(label);
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}
```

CircleIcon Class

Projects | Files | Services

- Homework_1
- Homework_2
- Homework_3
  - Source Packages
    - Chapter_3
    - Chapter_4_1
    - Chapter_4_2
      - CircleIcon.java
      - Main.java
    - Chapter_4_3
  - Test Packages
  - Libraries
  - Test Libraries

CircleIcon - Navigator

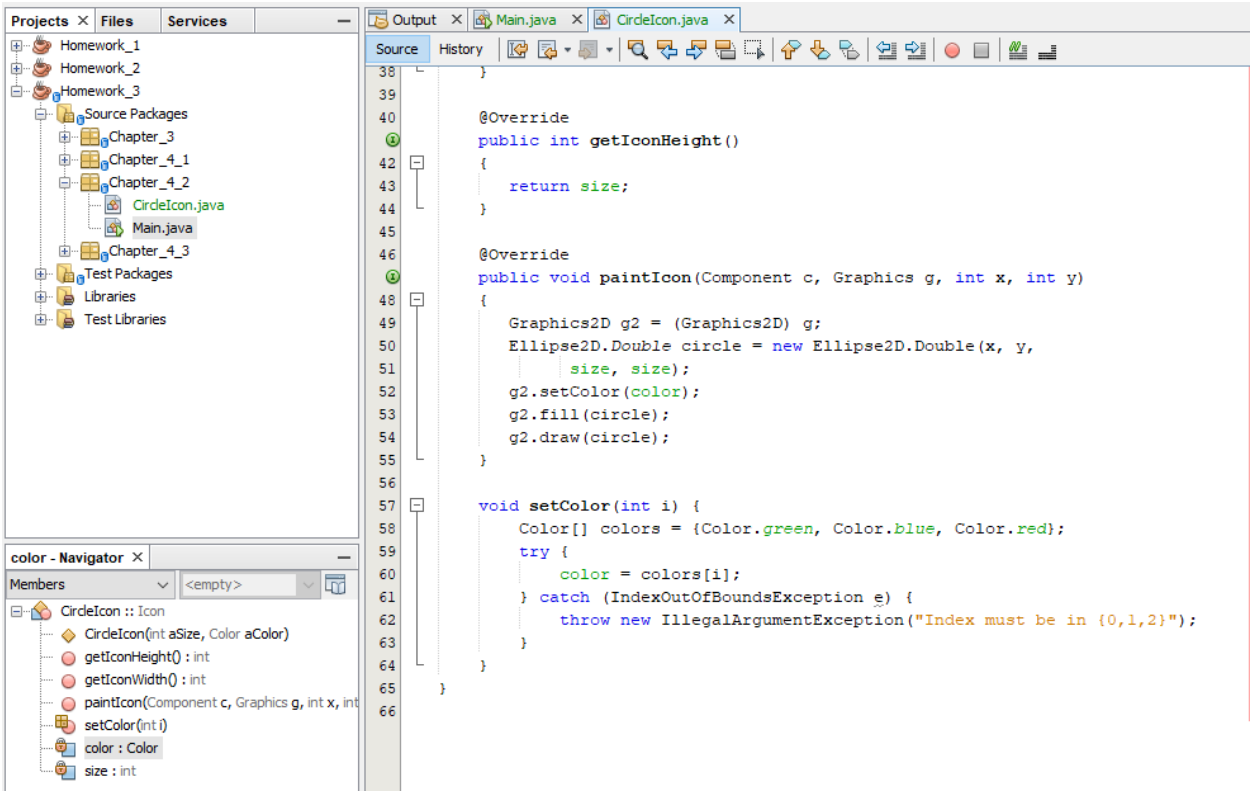
Members | <empty>

- CircleIcon :: Icon
  - CircleIcon(int aSize, Color aColor)
  - getIconHeight() : int
  - getIconWidth() : int
  - paintIcon(Component c, Graphics g, int x, int
  - setColor(int i)
  - color : Color
  - size : int

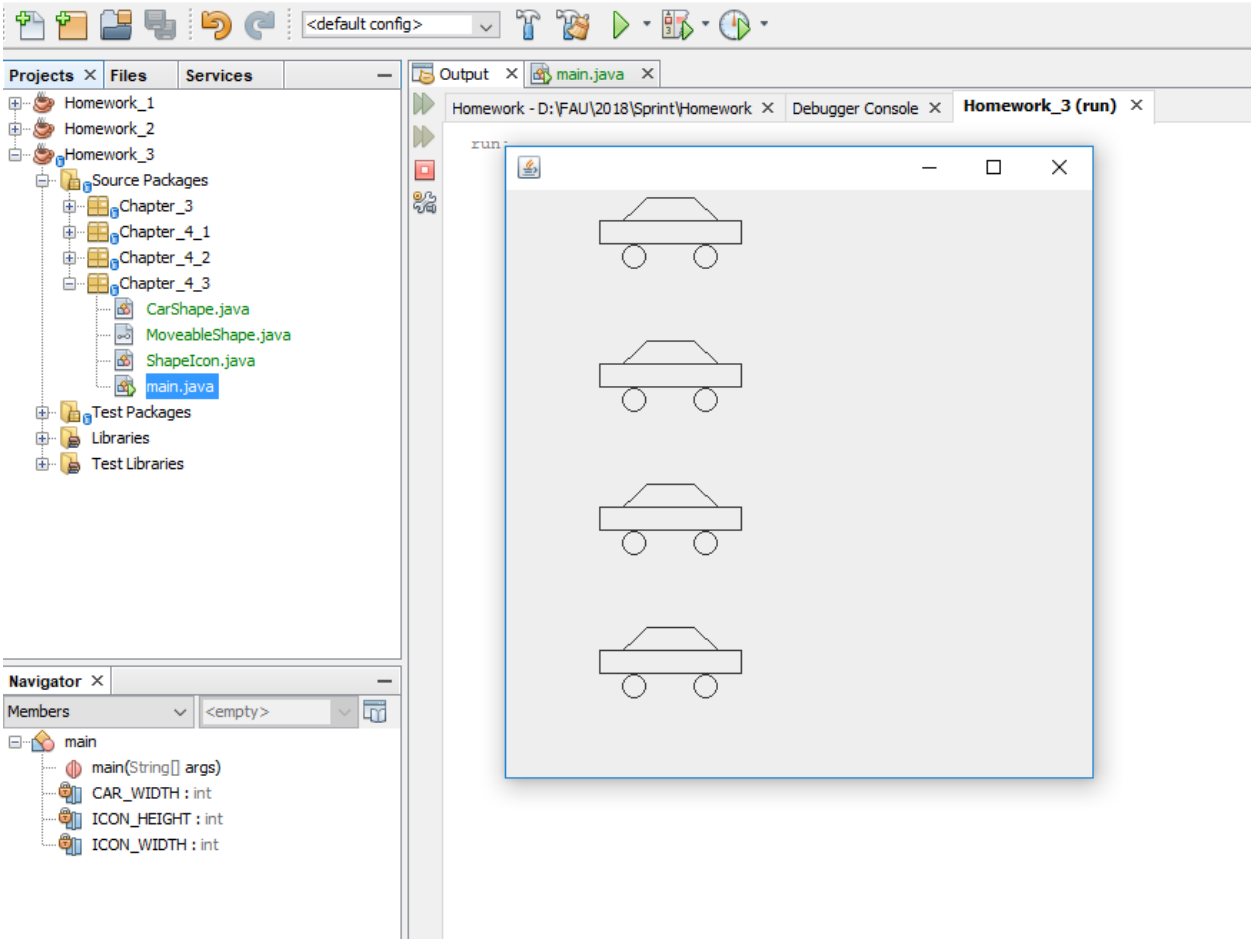
Output | Main.java | CircleIcon.java

Source | History

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Chapter_4_2;

import java.awt.Color;
import java.awt.Component;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import javax.swing.Icon;

/**
 *
 * @author Carlos Guisao
 */
public class CircleIcon implements Icon
{
    private final int size;
    private Color color;

    /**
     * @param aSize
     * @param aColor */

    public CircleIcon(int aSize, Color aColor)
    {
        size = aSize;
        color = aColor;
    }

    @Override
    public int getIconWidth()
    {
        return size;
    }
```


Projects
Files
Services
Homework_1
Homework_2
Homework_3
Source Packages
Chapter_3
Chapter_4_1
Chapter_4_2
CircleIcon.java
Main.java
Chapter_4_3
Test Packages
Libraries
Test Libraries
color - Navigator
Members
<empty>
CircleIcon :: Icon
CircleIcon(int aSize, Color aColor)
getIconHeight() : int
getIconWidth() : int
paintIcon(Component c, Graphics g, int x, int
setColor(int i)
color : Color
size : int
Output
Main.java
CircleIcon.java
Source
History
}
@Override
public int getIconHeight()
{
return size;
}
@Override
public void paintIcon(Component c, Graphics g, int x, int y)
{
Graphics2D g2 = (Graphics2D) g;
Ellipse2D.Double circle = new Ellipse2D.Double(x, y,
size, size);
g2.setColor(color);
g2.fill(circle);
g2.draw(circle);
}
void setColor(int i) {
Color[] colors = {Color.green, Color.blue, Color.red};
try {
color = colors[i];
} catch (IndexOutOfBoundsException e) {
throw new IllegalArgumentException("Index must be in {0,1,2}");
}
}
}

4.3


Projects
Files
Services
Homework_1
Homework_2
Homework_3
Source Packages
Chapter_3
Chapter_4_1
Chapter_4_2
Chapter_4_3
CarShape.java
MoveableShape.java
ShapeIcon.java
main.java
Test Packages
Libraries
Test Libraries
Navigator
Members
main
main(String[] args)
CAR_WIDTH : int
ICON_HEIGHT : int
ICON_WIDTH : int
Output
Homework - D:\FAU\2018\Sprint\Homework
Debugger Console
Homework_3 (run)

Main

Projects | Files | Services

- Homework_1
- Homework_2
- Homework_3
  - Source Packages
    - Chapter_3
    - Chapter_4_1
    - Chapter_4_2
    - Chapter_4_3
      - CarShape.java
      - MoveableShape.java
      - ShapeIcon.java
      - main.java
  - Test Packages
  - Libraries
  - Test Libraries

main - Navigator

Members | <empty>

- main

Output | main.java

Source | History

```java
/**
 *
 * @author Carlos Guisao
 */
/**
06:     This program implements an animation that moves
07:     a car shape.
08: */

package Chapter_4_3;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class main
 {
    public static void main(String[] args)
    {
        JFrame frame = new JFrame();

        final MoveableShape[] shapes = {
                new CarShape(0, 0, CAR_WIDTH), new CarShape(0, 100, CAR_WIDTH),
                new CarShape(0, 200, CAR_WIDTH), new CarShape(0, 300, CAR_WIDTH)
        };

        ShapeIcon icon = new ShapeIcon(shapes,
                ICON_WIDTH, ICON_HEIGHT);

        final JLabel label = new JLabel(icon);
        frame.setLayout(new FlowLayout());
        frame.add(label);

        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
```

Projects | Files | Services

- Homework_1
- Homework_2
- Homework_3
  - Source Packages
    - Chapter_3
    - Chapter_4_1
    - Chapter_4_2
    - Chapter_4_3
      - CarShape.java
      - MoveableShape.java
      - ShapeIcon.java
      - main.java
  - Test Packages
  - Libraries
  - Test Libraries

main - Navigator

Members | <empty>

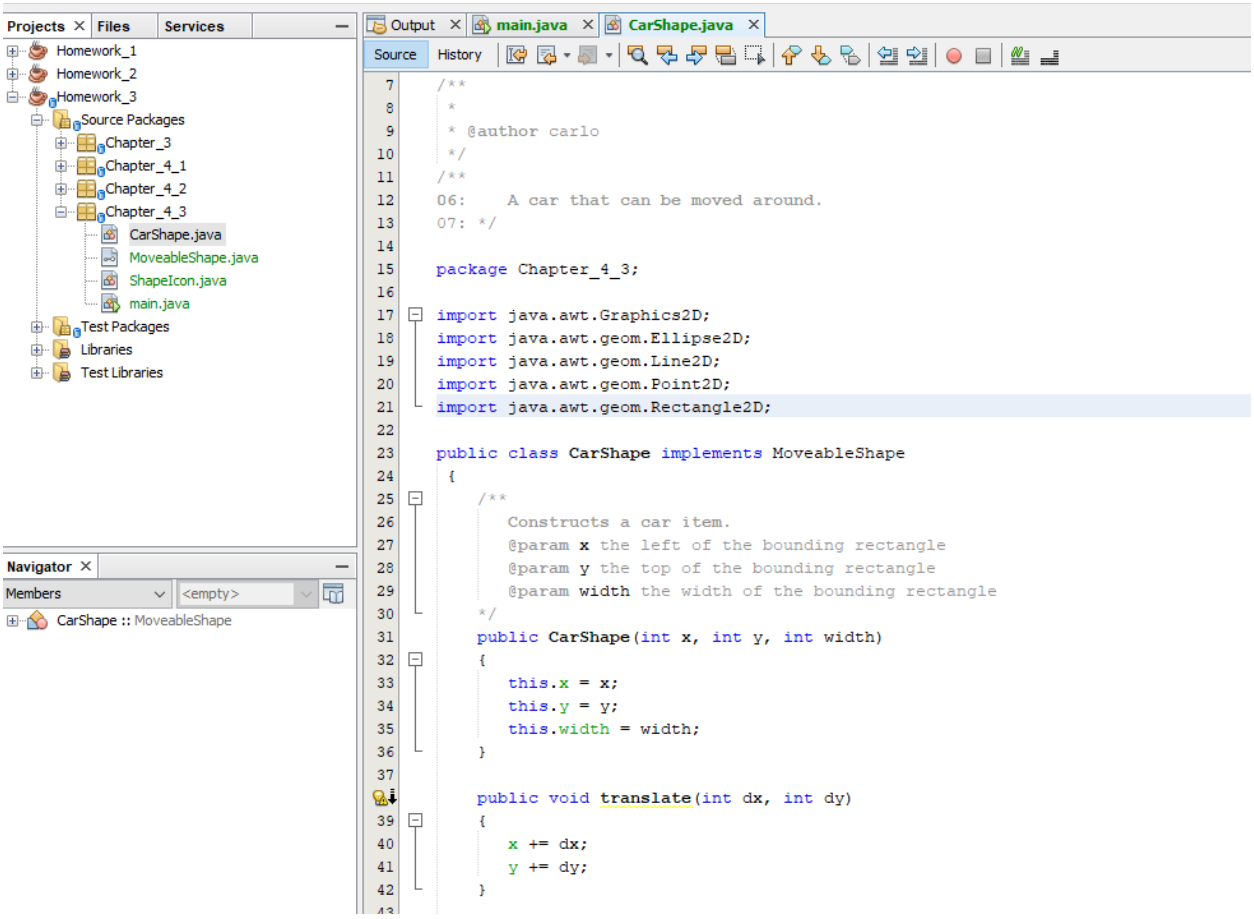
- main

Output | main.java

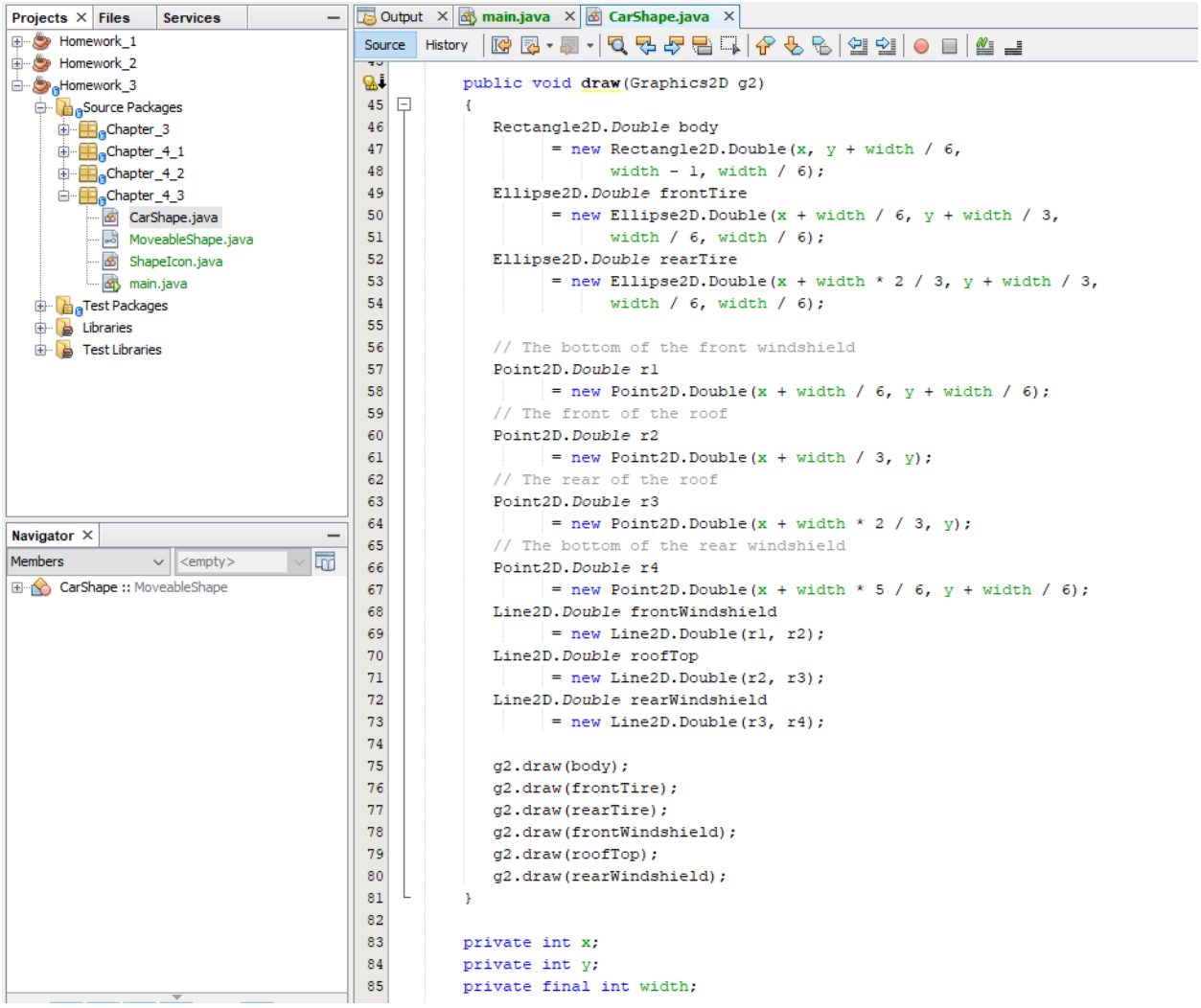
Source | History

```java
        final JLabel label = new JLabel(icon);
        frame.setLayout(new FlowLayout());
        frame.add(label);

        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);

        final int DELAY = 100;
        // Milliseconds between timer ticks
        Timer t = new Timer(DELAY, new
                ActionListener()
                {
                    public void actionPerformed(ActionEvent event)
                    {
                        for (MoveableShape shape : shapes) {
                            shape.translate(1, 0);
                        }
                        label.repaint();
                    }
                });
        t.start();
    }

    private static final int ICON_WIDTH = 400;
    private static final int ICON_HEIGHT = 400;
    private static final int CAR_WIDTH = 100;
}
```

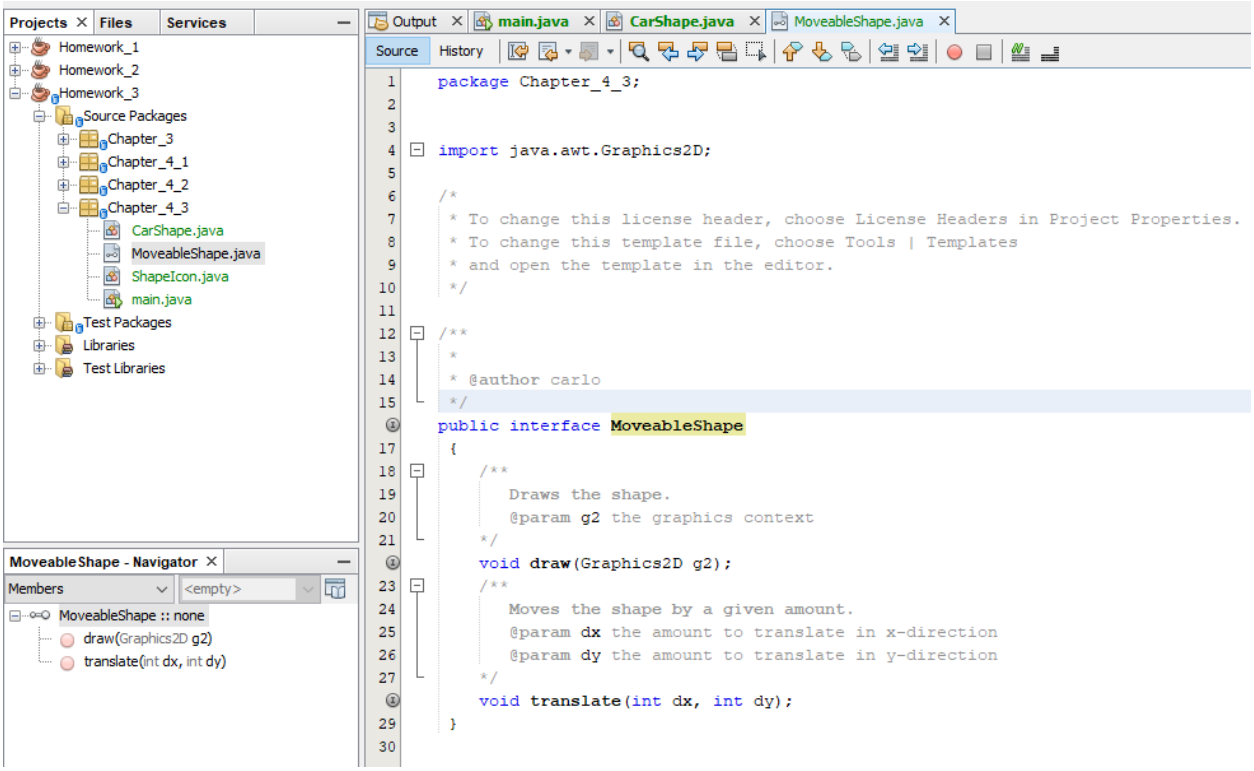
CarShape Class




```java
/**
 *
 * @author carlo
 */
/**
06:    A car that can be moved around.
07: */

package Chapter_4_3;

import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import java.awt.geom.Line2D;
import java.awt.geom.Point2D;
import java.awt.geom.Rectangle2D;

public class CarShape implements MoveableShape
 {
    /**
       Constructs a car item.
       @param x the left of the bounding rectangle
       @param y the top of the bounding rectangle
       @param width the width of the bounding rectangle
    */
    public CarShape(int x, int y, int width)
    {
       this.x = x;
       this.y = y;
       this.width = width;
    }

    public void translate(int dx, int dy)
    {
       x += dx;
       y += dy;
    }
```



MoveableShape Class



ShapeIcon Class

Projects | Files | Services

- Homework_1
- Homework_2
- Homework_3
  - Source Packages
    - Chapter_3
    - Chapter_4_1
    - Chapter_4_2
    - Chapter_4_3
      - CarShape.java
      - MoveableShape.java
      - ShapeIcon.java
      - main.java
  - Test Packages
  - Libraries
  - Test Libraries

Navigator

Members | <empty>

- ShapeIcon :: Icon
  - ShapeIcon(MoveableShape[] shapes, int wid
  - getIconHeight() : int
  - getIconWidth() : int
  - paintIcon(Component c, Graphics g, int x, int
  - height : int
  - shapes : MoveableShape[]
  - width : int

Source | History

```java
package Chapter_4_3;

import java.awt.Component;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.Icon;

/**
 *
 * @author carlo
 */
public class ShapeIcon implements Icon
 {
    public ShapeIcon(MoveableShape[] shapes,
       int width, int height)
    {
       this.shapes = shapes;
       this.width = width;
       this.height = height;
    }

    @Override
    public int getIconWidth()
    {
       return width;
    }

    @Override
    public int getIconHeight()
    {
       return height;
    }

    @Override
    public void paintIcon(Component c, Graphics g, int x, int y)
   {
       Graphics2D g2 = (Graphics2D) g;
       for(MoveableShape shape : shapes) {
           shape.draw(g2);
        }
    }
```

```java
    }

    private final int width;
    private final int height;
    private final MoveableShape[] shapes;
}
```