✘ elijahkorneffel@Elijahs-MacBook-Pro  ~/Git/CS2261-JAVA/Proj$

ct4   04\_12\_19\_Proj4 ●  Script Ex13\_9\_TestOutput

Script started, output file is Ex13\_9\_TestOutput

elijahkorneffel@Elijahs-MacBook-Pro  ~/Git/CS2261-JAVA/Projec$

4   04\_12\_19\_Proj4 ●  cat Ex13\_9\_Test.java

//Class: CS2261-JAVA

//Due Date: 04/15/19

//Author: Elijah Korneffel

//Description: This script contains the GeometricObject and Cir$

le classes

import java.util.Arrays;

public class Ex13\_9\_Test

{

public static class GeometricObject {

private String color = "white";

private boolean filled;

private java.util.Date dateCreated;

/\*\* Construct a default geometric object \*/

protected GeometricObject() {

dateCreated = new java.util.Date();

}

/\*\* Construct a geometric object with color and filled $

alue \*/

protected GeometricObject(String color, boolean filled)

{

dateCreated = new java.util.Date();

this.color = color;

this.filled = filled;

}

/\*\* Return color \*/

public String getColor() {

return color;

}

/\*\* Set a new color \*/

public void setColor(String color) {

this.color = color;

}

/\*\* Return filled. Since filled is boolean,

\* the get method is named isFilled \*/

public boolean isFilled() {

return filled;

}

/\*\* Set a new filled \*/

public void setFilled(boolean filled) {

this.filled = filled;

}

/\*\* Get dateCreated \*/

public java.util.Date getDateCreated() {

return dateCreated;

}

@Override

public String toString() {

return "created on " + dateCreated + "\ncolor: "

+ color +

" and filled: " + filled;

}

}

public static class Circle extends GeometricObject implement

s Comparable<Circle>

{

private double radius;

public Circle() {

}

public Circle(double radius) {

this.radius = radius;

}

public Circle(double radius,

String color, boolean filled) {

this.radius = radius;

setColor(color);

setFilled(filled);

}

/\*\* Return radius \*/

public double getRadius() {

return radius;

}

/\*\* Set a new radius \*/

public void setRadius(double radius) {

this.radius = radius;

}

/\*\* Return area \*/

public double getArea() {

return radius \* radius \* Math.PI;

}

/\*\* Return diameter \*/

public double getDiameter() {

return 2 \* radius;

}

/\*\* Return perimeter \*/

public double getPerimeter() {

return 2 \* radius \* Math.PI;

}

/\*\* Print the circle info \*/

public void printCircle()

{

System.out.println("The circle is created " + getDat

eCreated() + " and the radius is " + radius);

}

//Overrides equals method

@Override

public boolean equals(Object o)

{

return this.compareTo((Circle)o) == 0;

}

//Overrides compareTo method

@Override

public int compareTo(Circle o)

{

if(this.radius < o.radius)

{

return -1;

}

else if(this.radius > o.radius)

{

return 1;

}

else

{

return 0;

}

}

}

//Tests out class

public static void main(String[] args)

{

Circle[] circleList = new Circle[3];

circleList[2] = new Circle(1.0);

circleList[1] = new Circle(2.0);

circleList[0] = new Circle(3.0);

/\*for(int i = 0; i < circleList.length; ++i)

{

for(int j = 0; j < circleList.length; ++j)

{

if(circleList[i] < circleList[j])

{

Arrays.sort(circleList);

}

}

}\*/

Arrays.sort(circleList);

for(int i = 0; i < circleList.length; ++i)

{

circleList[i].printCircle();

}

Circle circle1 = new Circle(1.0);

//Checks to see if circle1 is equal to circleList[0] to de

monstrate equals

System.out.println(circle1.equals(circleList[0]));

}

}

elijahkorneffel@Elijahs-MacBook-Pro  ~/Git/CS2261-JAVA/Project

4   04\_12\_19\_Proj4 ●  javac Ex13\_9\_Test.java

elijahkorneffel@Elijahs-MacBook-Pro  ~/Git/CS2261-JAVA/Projec$4   04\_12\_19\_Proj4 ●  java Ex13\_9\_Test

The circle is created Wed Apr 17 13:51:43 CDT 2019 and the radi$s is 1.0

The circle is created Wed Apr 17 13:51:43 CDT 2019 and the radi$s is 2.0

The circle is created Wed Apr 17 13:51:43 CDT 2019 and the radi$s is 3.0

true

elijahkorneffel@Elijahs-MacBook-Pro  ~/Git/CS2261-JAVA/Projec$4   04\_12\_19\_Proj4 ● 

Script done, output file is Ex13\_9\_TestOutput