CS 1073

FR03A

Assignment #4

Ethan A. McCarthy

3573807

# Section 1

/\*\*

 \* @author Ethan McCarthy 3573807

 \*/

public class CirclePan{

    private double radius;

    private double depth;

    /\*\*

     \* constructor to make the circle pan

     \* @param radiusIn radius input

     \* @param depthIn depth input

     \*/

    public CirclePan (double radiusIn, double depthIn){

        radius = radiusIn;

        depth = depthIn;

    }

    /\*\*

     \* get the radius of the pan

     \* @return the radius of the pan

     \*/

    public double getRadius(){

        return radius;

    }

    /\*\*

     \* get the depth of the pan

     \* @return the depth of the pan

     \*/

    public double getDepth(){

        return depth;

    }

    /\*\*

     \* method to calculate the suface area of the pan and return it

     \* @return the total surface area

     \*/

    public double surfaceArea(){

        double area = (Math.PI \* radius \* radius) + (Math.PI \* 2 \* radius \* depth);

        return area;

    }

    /\*\*

     \* method to calulate the total volume of the pan

     \* @return the total volume of the pan

     \*/

    public double calcVolume(){

        double volume = (Math.PI \* radius \* radius \* depth);

        return volume;

    }

}

/\*\*

 \* @author Ethan McCarthy 3573807

 \*/

 public class HexagonPan{

    private double baseEdge;

    private double depth;

    private double baseArea;

    /\*\*

     \* hexagon pan constructor

     \* @param baseEdgeIn edge of base

     \* @param depthIn depth

     \*/

    public HexagonPan (double baseEdgeIn, double depthIn){

        baseEdge = baseEdgeIn;

        depth = depthIn;

        baseArea = (((3\*Math.sqrt(3))/2) \* (baseEdgeIn \* baseEdgeIn));

    }

    /\*\*

     \* get edge of base

     \* @return base edge

     \*/

    public double getBaseEdge(){

        return baseEdge;

    }

    /\*\*

     \* get the depth of the pan

     \* @return depth

     \*/

    public double getDepth(){

        return depth;

    }

    /\*\*

     \* get the area of the base of the pan

     \* @return area of the base

     \*/

    public double getBaseArea(){

        return baseArea;

    }

    /\*\*

     \* method to calculate the total surface area of the pan

     \* @return the total surface area

     \*/

    public double surfaceArea(){

        double surfaceArea;

        surfaceArea = baseArea + (baseEdge \* 6 \* depth);

        return surfaceArea;

    }

    /\*\*

     \* method to calculate the volume of the cake pan

     \* @return the total volume that the pan can hold

     \*/

    public double volume(){

        double volume;

        volume = baseArea \* depth;

        return volume;

    }

 }

/\*\*

 \* @author Ethan McCarthy 3573807

 \*/

import java.util.Scanner;

import java.text.NumberFormat;

 public class PanDriver{

    public static void main (String[] args){

        Scanner scan = new Scanner(System.in);

        NumberFormat formatter = NumberFormat.getNumberInstance();

        formatter.setMaximumFractionDigits(3);

        formatter.setMinimumFractionDigits(3);

        double leastArea = 0;

        String type = "N/A";

        double mostVolume = 0;

        System.out.println("Select one of the following options: \n1 - Enter information for a circle pan\n2 - Enter information for a hexagon pan\n3 - Quit");

        System.out.println("Enter your choice: ");

        int userIn = scan.nextInt();

        while(userIn != 3){

            if (userIn == 1){

                System.out.println("Input depth of the pan (in cm): ");

                double depthIn = scan.nextDouble();

                while(depthIn <= 0){

                    System.out.println("Invalid input. Please enter a number greater than 0.");

                    depthIn = scan.nextDouble();

                }

                System.out.println("Input the radius of the pan (in cm): ");

                double radiusIn = scan.nextDouble();

                while(radiusIn <= 0){

                    System.out.println("Invalid input. Please enter a number greater than 0.");

                    radiusIn = scan.nextDouble();

                }

                CirclePan circlePan = new CirclePan(radiusIn, depthIn);

                System.out.println("Suface area of the pan: " + formatter.format(circlePan.surfaceArea()) + "cm^2");

                System.out.println("Volume of the pan: " + formatter.format(circlePan.calcVolume()) + "cm^3");

                if(leastArea == 0){

                    leastArea = circlePan.surfaceArea();

                    type = "Circle";

                }

                else if(circlePan.surfaceArea() < leastArea){

                    leastArea = circlePan.surfaceArea();

                    type = "Circle";

                }

                if(mostVolume == 0){

                    mostVolume = circlePan.calcVolume();

                    type = "Circle";

                }

                else if(circlePan.calcVolume() > mostVolume){

                    mostVolume = circlePan.calcVolume();

                    type = "Circle";

                }

            }

            else if (userIn == 2){

                System.out.println("Input depth the depth of the pan: ");

                double depthIn = scan.nextDouble();

                while(depthIn <= 0){

                    System.out.println("Invalid input. Please enter a number greater than 0.");

                    depthIn = scan.nextDouble();

                }

                System.out.println("Input the length of the base edge of the pan: ");

                double baseEdgeIn = scan.nextDouble();

                while(baseEdgeIn <= 0){

                    System.out.println("Invalid input. Please enter a number greater than 0.");

                    baseEdgeIn = scan.nextDouble();

                }

                HexagonPan hexPan = new HexagonPan(baseEdgeIn, depthIn);

                System.out.println("Surface area of the pan: " + formatter.format(hexPan.surfaceArea()) + "cm^2");

                System.out.println("Volume of the pan: " + formatter.format(hexPan.volume()) + "cm^3");

                if(leastArea == 0){

                    leastArea = hexPan.surfaceArea();

                    type = "Hexagon";

                }

                else if(hexPan.surfaceArea() < leastArea){

                    leastArea = hexPan.surfaceArea();

                    type = "Hexagon";

                }

                if(mostVolume == 0){

                    mostVolume = hexPan.volume();

                    type = "Hexagon";

                }

                else if(hexPan.volume() > mostVolume){

                    mostVolume = hexPan.volume();

                    type = "Hexagon";

                }

            }

            else{

                System.out.println("Invalid input. Please input and accepted number.");

            }

            System.out.println("Select one of the following options: \n1 - Enter information for a circle pan\n2 - Enter information for a hexagon pan\n3 - Quit");

            System.out.println("Enter your choice: ");

            userIn = scan.nextInt();

        }

        System.out.println("\nThe pan with the smallest surface area is a: " + type + "\nSurface Area: " + formatter.format(leastArea) + "cm^2");

        System.out.println("The pan with the largest volume is a: " + type + "\nVolume: " + formatter.format(mostVolume) + "cm^3");

    }

 }

# Text Description automatically generatedSection 2

Text

Description automatically generated

# Section 3

/\*\*

 \* @author Ethan McCarthy 3573807

 \*/

import java.util.Scanner;

 public class NumberSystem{

    public static void main(String[] args){

        Scanner scan = new Scanner(System.in);

        System.out.println("Input a number between 1 and 99999:");

        int input = scan.nextInt();

        while(input < 1 || input > 99999){

            System.out.println("Invalid input. Please input a number within the range.");

            input = scan.nextInt();

        }

        int x, mLowercase, mUppercase, hLowercase, hUppercase, fLowercase, fUppercase, eLowercase, eUppercase, rLowercase, rUppercase;

        x = mLowercase = mUppercase = hLowercase = hUppercase = fLowercase = fUppercase = eLowercase = eUppercase = rLowercase = rUppercase = 0;

        while (input > 0){

            if((input/59049) != 0){

                x = input/59049;

                input = input % 59049;

            }

            if((input/19683) != 0){

                mUppercase = input/19683;

                input = input % 19683;

            }

            if((input/6561) != 0 ){

                mLowercase = input/6561;

                input = input % 6561;

            }

            if((input/2187) != 0){

                hUppercase = input/2187;

                input = input % 2187;

            }

            if((input/729) != 0){

                hLowercase = input/729;

                input = input % 729;

            }

            if((input/243) != 0){

                fUppercase = input/243;

                input = input % 243;

            }

            if((input/81) != 0){

                fLowercase = input/81;

                input = input % 81;

            }

            if((input/27) != 0){

                eUppercase = input/27;

                input = input % 27;

            }

            if((input/9) != 0){

                eLowercase = input/9;

                input = input % 9;

            }

            if((input/3) != 0){

                rUppercase = input/3;

                input = input % 3;

            }

            if((input/1) != 0){

                rLowercase = input/1;

                input = input % 1;

            }

        }

        for (int i = 0; i < rLowercase; i++){

            System.out.print("r");

        }

        for (int i = 0; i < rUppercase; i++){

            System.out.print("R");

        }

        for (int i = 0; i < eLowercase; i++){

            System.out.print("e");

        }

        for (int i = 0; i < eUppercase; i++){

            System.out.print("E");

        }

        for (int i = 0; i < fLowercase; i++){

            System.out.print("f");

        }

        for (int i = 0; i < fUppercase; i++){

            System.out.print("F");

        }

        for (int i = 0; i < hLowercase; i++){

            System.out.print("h");

        }

        for (int i = 0; i < hUppercase; i++){

            System.out.print("H");

        }

        for (int i = 0; i < mLowercase; i++){

            System.out.print("m");

        }

        for (int i = 0; i < mUppercase; i++){

            System.out.print("M");

        }

        for (int i = 0; i < x; i++){

            System.out.print("X");

        }

    }

 }

# Section 4

Text

Description automatically generatedText

Description automatically generated