/\*Draw a class diagram and develop the logic for the following:-

scenario.consider the class cuboid with length, breadth and height

as a private attributes.code the parameterized constructor and

toString() method. Develop main method that store

data of cuboid as a collection and must be menu

driven with the following operation:

1. Add new Cuboid

2. Display all

3. Sort by length

4. Sort by Area

[Hint : Area of cuboid is length\*breath\*height]\*/

//Cuboid

**package** p1;

**public** **class** Cuboid {

**private** **int** length;

**private** **int** breadth;

**private** **int** height;

Cuboid(**int** length,**int** breadth,**int** height)

{

**this**.length=length;

**this**.breadth=breadth;

**this**.height=height;

}

**public** **int** getL()

{

**return** length;

}

**public** **int** getB()

{

**return** breadth;

}

**public** **int** getH()

{

**return** height;

}

**public** **int** area()

{

**return** length\*breadth\*height;

}

**public** String toString()

{

**return** String.*format*("Length=%d breadth=%d height=%d Area=%d",getL(),getB(),getH(),area());

}

}

//Cuboid Demo

**package** p1;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Scanner;

**public** **class** CuboidDemo {

**public** **static** Scanner *s*;

**public** **static** ArrayList<Cuboid>*cu*;

**public** **static** **void** initialise(**int** n)

{

*s*=**new** Scanner(System.***in***);

*cu*=**new** ArrayList<Cuboid>();

}

**public** **static** **int** menuOption()

{

System.***out***.println("<<<<<<Menu>>>>>>>>");

System.***out***.println("1.Add new Cuboid");

System.***out***.println("2.Display all");

System.***out***.println("3.Sort by length");

System.***out***.println("4.Sort by Area");

System.***out***.println("5.Exit");

System.***out***.println("Enter the choice");

**return** *s*.nextInt();

}

**public** **static** **void** main(String[] args) {

**int** ch;

*initialise*(3);

**while**(**true**)

{

ch=*menuOption*();

**switch**(ch)

{

**case** 1:

*cu*.add(*addCuboid*());

**break**;

**case** 2:

System.***out***.println("==============All cuboid detail=============");

**for**(Cuboid c:*cu*)

{

System.***out***.println(c);

}

**break**;

**case** 3:

//SortByLength();

System.***out***.println("============Sorting by Length===============");

Collections.*sort*(*cu*,**new** SortByLength());

**for**(Cuboid cl:*cu*)

{

System.***out***.println(cl);

}

**break**;

**case** 4:

//SortByArea();

System.***out***.println("================Sorting by Area==============");

Collections.*sort*(*cu*,**new** SortByArea());

**for**(Cuboid cb:*cu*)

{

System.***out***.println(cb);

}

**break**;

**case** 5:

System.*exit*(0);

}

}

}

/\*private static void SortByArea() {

SortByArea SBA=new SortByArea();

}

private static void SortByLength() {

SortByLength SBL=new SortByLength();

}\*/

**private** **static** Cuboid addCuboid() {

System.***out***.println("Enter the length,breadth,height:");

**return** **new** Cuboid(*s*.nextInt(),*s*.nextInt(),*s*.nextInt());

}

}

//SortByLength

**package** p1;

**import** java.util.Comparator;

**public** **class** SortByLength **implements** Comparator<Cuboid>{

@Override

**public** **int** compare(Cuboid c1, Cuboid c2) {

**if**(c1.getL()==c2.getL())

**return** 0;

**else** **if**(c1.getL()>c2.getL())

**return** 1;

**else**

**return** -1;

}

}

//SortByArea

**package** p1;

**import** java.util.Comparator;

**public** **class** SortByArea **implements** Comparator<Cuboid>{

**public** **int** compare(Cuboid c1, Cuboid c2)

{

**if**(c1.area()==c2.area())

**return** 0;

**else** **if**(c1.area()>c2.area())

**return** 1;

**else**

**return** -1;

}

}

Output:

Sort by length

Text

Description automatically generated with low confidence

Sort by area

Text

Description automatically generated