Debugging Lightbulbs

En isenkræmmer har bedt om et program der udfra nogle matematiske formler kan bestemme den mest ”effektive” og mest ”komfortable” pære ud fra nogle givne parametre.

Producenterne opgiver følgende data om pæres attributter:

* Mærke
* Model
* Type
* Brightness
* Energy
* Life
* Warmth

En pære identificeres ved mærke og model.

Pærens effektivitet beregnes således:

Pærens komfort beregnes således:

Programmet er skrevet og kan findes på de følgende sider. (Programmet har ingen grafisk brugergrænseflade, kun konsol – Det er et designvalg, ikke en fejl).

Du kan også finde koden her: <https://github.com/Cphbusiness-dat1sem-org/Debugging_Lightbulbs>

**Det er din opgave at debugge systemet, finde og rette eventuelle fejl.**

Noter linienumre på alle fejl.

Du skal tage højde for både syntaktiske fejl, logiske fejl og uhensigtsmæssigheder i koden der gør den mindre robust og vanskeligere at vedligeholde.

Main.java

import java.util.ArrayList;

Public class Main {

public static void main(String[] args) {

new Main().go();

}

private void go() {

ArrayList<Bulb> l = new ArrayList<>();

l.add(new Bulb("Phillips", "8324198763", "LED", 806, 8, 15000, 80));

l.add(new Bulb("Phillips", "8324198232", "LED", 402, 4, 16000, 80));

l.add(new Bulb("Phillips", "8324194309", "CFL", 806, 16, 9000, 65));

l.add(new Bulb("Phillips", "8324194234", "CFL", 402, 8, 11000, 65));

l.add(new Bulb("Phillips", "8324194217", "Halogen", 1244, 8, 16000, 85));

l.add(new Bulb("Phillips", "8324194766", "Halogen", 806, 6, 20000, 85));

l.add(new Bulb("Osram", "814-9879", "LED", 810, 10, 20000, 90));

l.add(new Bulb("Osram", "814-9845", "LED", 470, 6, 20000, 90));

l.add(new Bulb("Osram", "814-5673", "CFL", 810, 14, 16000, 65));

l.add(new Bulb("Osram", "814-5612", "CFL", 470, 8, 14000, 65));

l.add(new Bulb("Osram", "814-8834", "Halogen", 810, 12, 16000, 85));

l.add(new Bulb("Osram", "814-8855", "Halogen", 470, 6, 20000, 85));

Bulb mostEfficient = MostEfficient(l);

System.out.println("Most efficient: "+mostEfficient);

Bulb mostPleasing = mostPleasant(l);

System.out.println("Most pleasant: "+mostEfficient);

}

public Bulb MostEfficient(ArrayList<Bulb> l){

double maxEfficiency = 0;

Bulb mostEfficient = l.get(0);

for(int i = 1; i < l.size(); i++){

Bulb b = l.get(i);

int brightness = b.getBrightness();

int energy = b.getWarmth();

int life = b.getLife();

double efficiency = brightness / energy \* (life / 1000);

if(efficiency > maxEfficiency)

maxEfficiency = efficiency;

mostEfficient = b;

}

return mostEfficient;

}

public Bulb mostPleasant(ArrayList<Bulb> l){

double maxPleasingScore = 0;

Bulb mostPleasing = l.get(0);

for(Bulb b : l){

int warmth = b.getWarmth();

int brightness = b.brightness;

double pleasingScore = warmth / brightness;

if(pleasingScore < maxPleasingScore){

maxPleasingScore = pleasingScore;

mostPleasing = b;

}

}

return mostPleasing;

}

}

## Bulb.java

import java.util.Objects;

public class Bulb {

String brand, model, type;

int brightness, energy, life, warmth;

public Bulb(String brand,

String type,

String model,

int brightness,

int energy,

int life,

int warmth) {

this.brand = brand;

model = this.model;

this.type = type;

this.brightness = brightness;

this.energy = energy;

this.life = life;

this.warmth = warmth;

}

public String getBrand() { return brand; }

public String getModel() { return model; }

public String getType() { return type; }

public int getBrightness() { return energy; }

public int getEnergy() { return energy; }

public int getLife() { return life; }

public int getWarmth() { return warmth; }

// @Override

public boolean equals(Bulb obj) {

if (this == obj) {

return true;

}

if (obj == null) {

return false;

}

if (getClass() != obj.getClass()) {

return false;

}

final Bulb other = (Bulb) obj;

if (this.brightness != other.brightness) {

return false;

}

if (!Objects.equals(this.brand, other.brand)) {

return false;

}

return true;

}

@Override

public String toString() {

return "Bulb{" +

"brand=" + brand + ", "

+ "model=" + model + ", "

+ "type=" + type + ", "

+ "brightness=" + brightness + ", "

+ "energy=" + energy + ", "

+ "life=" + life + ", "

+ "warmth=" + life + '}';

}