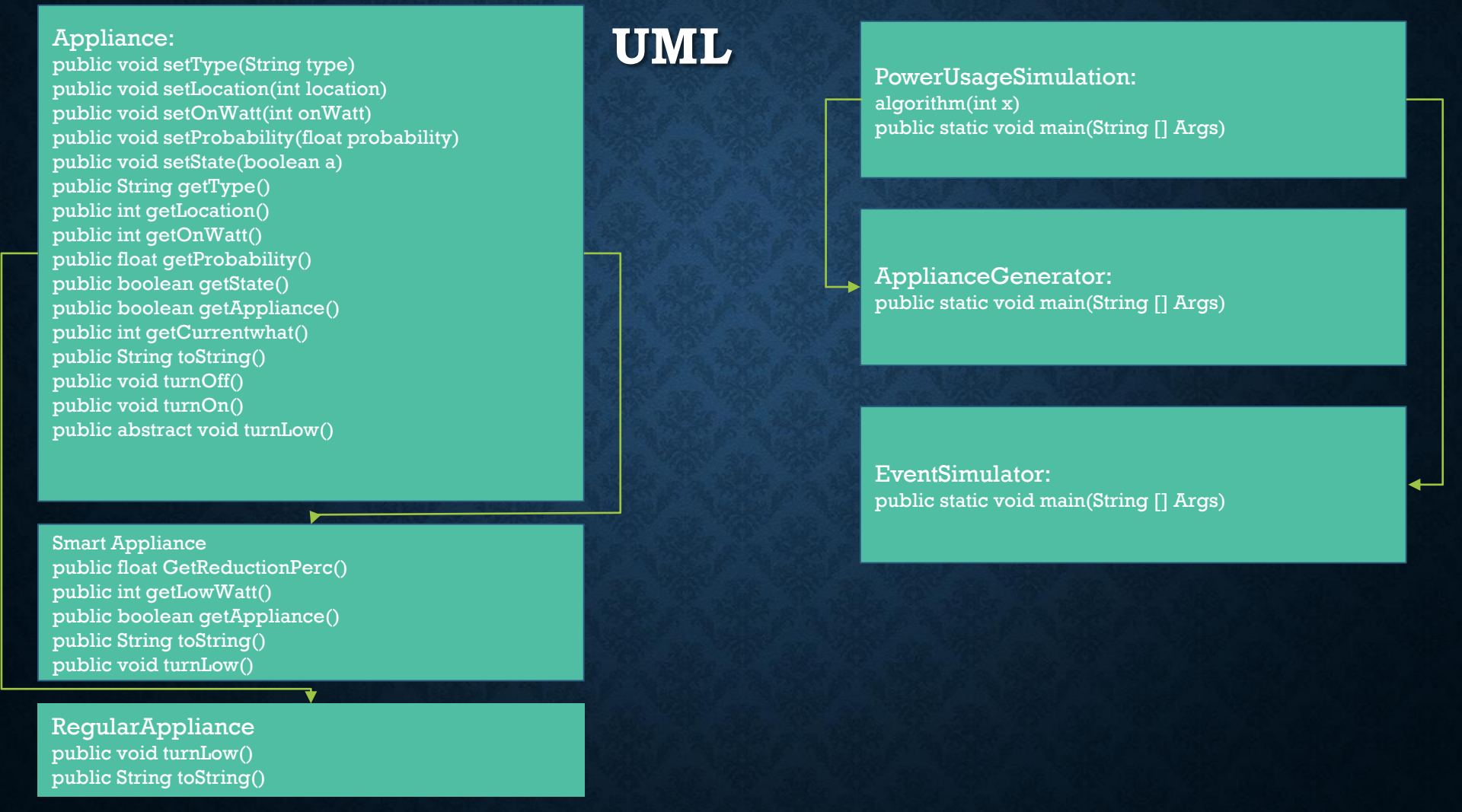
CS116 PROJECT

by Theo Guidroz

PROJECT INFO:

- Design a program which will take 2 types of appliances: Smart and Regular appliances.
- Store the appliances in a single database.
- Run test cases to optimize the power usage of all the appliances.
- The first power saving step is to set smart devices on low power.
- If all smart devices are on low power and the total power is still above the limit, implement an algorithm to burn out locations until total power is below the limit.

UML


Appliance:
public void setType(String type)
public void setLocation(int location)
public void setOnWatt(int onWatt)
public void setProbability(float probability)
public void setState(boolean a)
public String getType()
public int getLocation()
public int getOnWatt()
public float getProbability()
public boolean getState()
public boolean getAppliance()
public int getCurrentwhat()
public String toString()
public void turnOff()
public void turnOn()
public abstract void turnLow()
PowerUsageSimulation:
algorithm(int x)
public static void main(String [] Args)
ApplianceGenerator:
public static void main(String [] Args)
EventSimulator:
public static void main(String [] Args)
Smart Appliance
public float GetReductionPerc()
public int getLowWatt()
public boolean getAppliance()
public String toString()
public void turnLow()
RegularAppliance
public void turnLow()
public String toString()

DESCRIPTION OF ALGORITHM

- The first role of the algorithm is to check if the total power consumed is higher than the limit.
- If it is, its first step is to change the state of smart appliances to low.
- If the total power still exceeds the limit, the algorithm:
  - Finds the location with the smallest average probability of appliances being on.
  - Turns all the appliances in that location to off (brown out).
  - Continues doing so until the total power is under the limit.
  - It then prints all the locations that has been browned out.
  - It also outputs a file called fullReport.txt that gives a list of all the devices which has been turned to low and switched off.

Why use probability and not total wattage?

ALGORITHM

```
public static void algorithm(int x) {
    double totalPower=0;
    String fullDetail="";
    for(int i =0;i<generalArr.length&&generalArr[i]!=null;i++) {
        totalPower+=generalArr[i].getCurrentwhat();
    }
    System.out.println("The initial total power before implementing the algorithm is "+totalPower+"W");
    if(totalPower<x) {
        System.out.println("No location has been browned out and no smart devices has been switched to low.");
        fullDetail+="No location has been browned out and no smart devices has been switched to low."+System.lineSeparator();
    }
    else {
        fullDetail+="The following appliances have been set to low:"+System.lineSeparator();
        for(int i =0; i<generalArr.length&&generalArr[i]!=null;i++) {
            if(totalPower>x) {
                totalPower-=generalArr[i].getCurrentwhat();
                generalArr[i].turnLow();
                totalPower+=generalArr[i].getCurrentwhat();
                fullDetail+=generalArr[i].toString()+System.lineSeparator();
            }
            else {
                System.out.println("No location has been browned out.");
                fullDetail+=System.lineSeparator()+"No location has been browned out."+System.lineSeparator();
                break;
            }
        }
    }
    while(totalPower>x) {
```

ALGORITHM

```
while(totalPower>x) {
    fullDetail+=System.lineSeparator()+"The following devices have been browned out:"+System.lineSeparator();
    float sumProb=0;
    int minLocation=0;
    double ave=0;
    double minAverageProbability=2;
    int count3=0;
    int count7=0;
    while(count7<locationCount) {
        for(int i=0;i<generalCount;i++) {
            if((generalArr[i].getLocation()==locationArr[count7])&&(generalArr[i].getState())) {
                sumProb+=generalArr[i].getProbability();
                count3++;
            }
        }
        ave=sumProb/count3;
        if(ave<minAverageProbability) {
            minAverageProbability=ave;
            minLocation=locationArr[count7];
        }
        count7++;
    }
    System.out.println("Location "+minLocation+" has been browned out.");

    for(int i=0;i<generalArr.length&&generalArr[i]!=null;i++) {
        if(generalArr[i].getLocation()==minLocation&&generalArr[i].getState()) {
            fullDetail+=generalArr[i].toString()+System.lineSeparator();
            totalPower-=generalArr[i].getCurrentwhat();
            generalArr[i].turnOff();
            System.out.print("");
```

TEST CASES

Test Case	Output	Tested?
Add an appliance to database	Appliance toString shown	Yes
Delete the appliance added	Appliance toString shown	Yes
Load output.txt	To String of all appliances	Yes
Find one type	All same type to string	Yes
Simulator(10000000W)	No device set to low or browned	Yes
Simulator(80000W)	Some devices set to Low, some cases of brown out.	Yes
Simulator(10000W)	Lot of brown out for each events.	Yes