Tabor Alemu
Juhnyx Fetero
David Galicia
Alek Popovic

Design

Our UML class diagram is available as a PDF in the same folder that this document is in. You can also access it using the link below.
https://www.lucidchart.com/invitations/accept/c43ef179-2780-43b9-9842-613f77ac79bb

The lowerWatts method will manage the appliances using the following logic:

Sort the locations by the percent of the appliances that are on at each location. For the locations that only have 1/3 or less of their appliances turned on, sort their appliances by current wattage being used. Starting with the location with the lowest percent of appliances turned on, turn off the smart appliance that is using the most watts at each location until we are below the target wattage. We are assuming that the people at the locations with 1/3 or less of their appliances turned on are not home. That being said, we still do not want to turn off all of the smart appliances at one location since we can never be 100% sure that nobody is home and for the fact that some people may have left specific smart appliances on, on purpose even though they are not home. If we turn off the highest watt consuming smart appliance at each one of these locations and our target wattage still is not met, keep turning off the next highest watt consuming smart appliance at these locations. If we turn off all the smart appliances at all of these locations where we assume nobody is home and it is still not enough, we will start repeating the process for the rest of the locations in the same manner. The majority of people do not use close to all of their appliances at once, so the ones who are producing a high amount of watts most likely need to be doing so at that time. Therefore, we would like to leave these locations alone for as long as we can. If all the smart appliances in our system are turned off, and we are still over the target wattage, we will begin cutting the power to certain locations (a brown out) using the same sorting/priority as before.

Test Cases

Appliance

Test	Expected Outcome	Reason for Test

Appliance a1 = new Appliance("toaster", 50, 0, 0.2, 101);
System.out.println(a1);
	1 type=toaster loc=101 currentW=0 on?=false OnW=50 OffW=0	Test the constructor and toString() method

Appliance a2 = new Appliance(a1);
System.out.println(a2);
	2 type=toaster loc=101 currentW=0 on?=false OnW=50 OffW=0	Test the copy constructor

System.out.println(a1.getId());	1	Test the getId() method

Tabor Alemu
Juhnyx Fetero
David Galicia
Alek Popovic

System.out.println(a1.getIsOn());	false	Test the getIsOn() method
System.out.println(a1.getLocation());	101	Test the getLocation method
System.out.println(a1.getOddsIsOn());	0.2	Test the getOddsIsOn() method
System.out.println(a1.getOffWatts());	0	Test the getOffWatts() method
System.out.println(a1.getOnWatts());	50	Test the getOnWatts() method
System.out.println(a1.getType());	toaster	Test the getType() method

a1.setLocation(202);
System.out.println(a1);
	1 type=toaster loc=202 currentW=0 on?=false OnW=50 OffW=0	Test the setLocation() method

a1.setOffWatts(10);
System.out.println(a1);
	1 type=toaster loc=202 currentW=0 on?=false OnW=50 OffW=10	Test the setOffWatts() method

a1.setOnWatts(200);
System.out.println(a1);
	1 type=toaster loc=202 currentW=0 on?=false OnW=200 OffW=10	Test the setOnWatts() method

a1.setOddsIsOn(0.25);
System.out.println(a1.getOddsIsOn());
	0.25	Test the setOddsIsOn method

a1.turnOn();
System.out.println(a1.getCurrentWatts());
	200	Test the turnOn() method

a1.turnOff();
System.out.println(a1.getCurrentWatts());
	10	Turn the turnOff() method

System.out.println(a1.compareTo(a2));	-150	Test the compareTo() method

SmartAppliance

Test	Expected Outcome	Reason for Test

SmartAppliance s1 = new SmartAppliance("toaster", 100, 5, 0.25, 101, 0.5);
System.out.println(s1);
	1 type=toaster loc=101 currentW=5 on?=false OnW=100 OffW=5 PercentSaving=0.5 smartOn=true	Test the constructor and toString() method

Tabor Alemu
Juhnyx Fetero
David Galicia
Alek Popovic

SmartAppliance s2 = new SmartAppliance(s1);
System.out.println(s2);
	2 type=toaster loc=101 currentW=5 on?=false OnW=100 OffW=5 PercentSaving=0.5 smartOn=true	Test the copy constructor

System.out.println(s1.getPercentSaving());	0.5	Test the getPercentSaving() method
System.out.println(s1.getSmartOn());	true	Test the getSmartOn() method

s1.turnOn();
s1.setPercentSaving(0.75);
System.out.println(s1.getCurrentWatts());
	25	Test the setPercentSaving() and getCurrentWatts() methods

s1.smartOff();
System.out.println(s1.getCurrentWatts());
	100	Test the smartOff() method

s1.smartOn();
System.out.println(s1);
	1 type=toaster loc=101 currentW=100 on?=true OnW=100 OffW=5 PercentSaving=0.75 smartOn=false	Test the smartOn() method

PowerUsageSystem

Test	Expected Outcome	Reason for Test

PowerUsageSystem pus1 = new PowerUsageSystem();
System.out.println(pus1.isEmpty());
	true	Test the default constructor and isEmpty() method

pus1.insert(new Appliance("oven", 90, 0, 0.5, 101));
System.out.println(pus1.length());
	1	Test the insert(Appliance) and length() methods

System.out.println(pus1.isThere(1));	1 type=oven loc=101 currentW=0 on?=false OnW=90 OffW=0	Thest the isThere() method

PowerUsageSystem pus2 = new PowerUsageSystem(20);
pus2.insert("output.txt");
System.out.println(pus2.length());
	20	Test the second constructor and insert(String) method

Tabor Alemu
Juhnyx Fetero
David Galicia
Alek Popovic

System.out.println(pus2.isFull());	true	Test the isFull() method

pus2.delete(5);
System.out.println(pus2.length());
	19	Test the delete() method

System.out.println(pus2.getAppliance(0));	15 type=Electric Frying Pan loc=10000001 currentW=0 on?=false OnW=1500 OffW=0	Test the getAppliance() method
System.out.println(pus2.summary());	This system contains the following: 2 total locations 1x Electric Frying Pan(s) 1x Vacuum - Regular(s) 1x Clothes Washer(s) 1x Coffee Maker(s) 1x Computer Printer (Printing)(s) 1x Refrigerator - 19 cu. ft. - Frostfree(s) 1x Refrigerator - 14 cu. ft. - Frostfree(s) 1x Television Plasma 42' - 50'(s) 1x Television - 60 inch(s) 1x Fish Tank (50 Gallon)(s) 1x Dishwasher (Without Dry Cycle)(s) 1x Refrigerator - 1.7 cu. ft.(s) 1x Television - 25 inch(s) 1x Lighting - Fluorescent 2 bulb(s) 2x Laptop(s) 1x Lighting - 60 Watt(s) 1x DVD(s) 1x Clock - Electric(s)	Test the summary() method
System.out.println(pus2);	15 type=Electric Frying Pan loc=10000001 currentW=0 on?=false OnW=1500 OffW=0 13 type=Vacuum - Regular loc=10000001 currentW=0 on?=false OnW=1440 OffW=0 6 type=Clothes Washer loc=10000001 currentW=0 on?=false OnW=1200 OffW=0 PercentSaving=0.25 smartOn=true 3 type=Coffee Maker loc=10000001 currentW=0 on?=false OnW=900 OffW=0 9 type=Computer Printer (Printing) loc=10000001 currentW=10 on?=false OnW=600 OffW=10 16 type=Refrigerator - 19 cu. ft. - Frostfree loc=10000001 currentW=0 on?=false OnW=509 OffW=0 PercentSaving=0.25 smartOn=true 11 type=Refrigerator - 14 cu. ft. - Frostfree loc=10000001 currentW=0 on?=false OnW=383 OffW=0 PercentSaving=0.25 smartOn=true 4 type=Television Plasma 42' - 50' loc=10000001 currentW=5 on?=false OnW=375 OffW=5 2 type=Television - 60 inch loc=10000001 currentW=5 on?=false OnW=240 OffW=5 7 type=Fish Tank (50 Gallon) loc=10000001 currentW=0 on?=false OnW=230 OffW=0 8 type=Dishwasher (Without Dry Cycle) loc=10000001 currentW=0 on?=false OnW=200 OffW=0 PercentSaving=0.25 smartOn=true 17 type=Refrigerator - 1.7 cu. ft. loc=10000002	Test the toString() method

	currentW=0 on?=false OnW=126 OffW=0 PercentSaving=0.25 smartOn=true 21 type=Television - 25 inch loc=10000002 currentW=5 on?=false OnW=123 OffW=5 18 type=Lighting - Fluorescent 2 bulb loc=10000002 currentW=0 on?=false OnW=100 OffW=0 12 type=Laptop loc=10000001 currentW=10 on?=false OnW=75 OffW=10 PercentSaving=0.5 smartOn=true 19 type=Laptop loc=10000002 currentW=10 on?=false OnW=75 OffW=10 PercentSaving=0.5 smartOn=true 10 type=Lighting - 60 Watt loc=10000001 currentW=0 on?=false OnW=60 OffW=0 14 type=DVD loc=10000001 currentW=5 on?=false OnW=60 OffW=5 20 type=Clock - Electric loc=10000002 currentW=0 on?=false OnW=3 OffW=0	

PowerUsageSystem pus3 = new PowerUsageSystem(2000, "output.txt");
System.out.println(pus3.length());
	1726	Test the third constructor

Appliance[] apps = pus3.getAppsAtLoc(10000001);
for (Appliance a : apps) { System.out.println(a); }
	35 type=Electric Frying Pan loc=10000001 currentW=0 on?=false OnW=1500 OffW=0
	33 type=Vacuum - Regular loc=10000001 currentW=0 on?=false OnW=1440 OffW=0
	26 type=Clothes Washer loc=10000001 currentW=0 on?=false OnW=1200 OffW=0 PercentSaving=0.25 smartOn=true
	23 type=Coffee Maker loc=10000001 currentW=0 on?=false OnW=900 OffW=0
	29 type=Computer Printer (Printing) loc=10000001 currentW=10 on?=false OnW=600 OffW=10
	36 type=Refrigerator - 19 cu. ft. - Frostfree loc=10000001 currentW=0 on?=false OnW=509 OffW=0 PercentSaving=0.25 smartOn=true
	31 type=Refrigerator - 14 cu. ft. - Frostfree loc=10000001 currentW=0 on?=false OnW=383 OffW=0 PercentSaving=0.25 smartOn=true
	24 type=Television Plasma 42' - 50' loc=10000001 currentW=5 on?=false OnW=375 OffW=5
	25 type=Computer with Monitor loc=10000001 currentW=10 on?=false OnW=270 OffW=10 PercentSaving=0.5 smartOn=true
	22 type=Television - 60 inch loc=10000001 currentW=5 on?=false OnW=240 OffW=5
	27 type=Fish Tank (50 Gallon) loc=10000001 currentW=0 on?=false OnW=230 OffW=0
	28 type=Dishwasher (Without Dry Cycle) loc=10000001 currentW=0 on?=false OnW=200 OffW=0 PercentSaving=0.25 smartOn=true
	32 type=Laptop loc=10000001 currentW=10 on?=false OnW=75 OffW=10 PercentSaving=0.5 smartOn=true
	30 type=Lighting - 60 Watt loc=10000001 currentW=0 on?=false OnW=60 OffW=0
	34 type=DVD loc=10000001 currentW=5 on?=false OnW=60 OffW=5
		Test the getAppsAtLoc() method

Tabor Alemu
Juhnyx Fetero
David Galicia
Alek Popovic

apps = pus3.getSmartAppsAtLoc(10000001);
for (Appliance a : apps) {
System.out.println(a); }
	26 type=Clothes Washer loc=10000001 currentW=0 on?=false OnW=1200 OffW=0 PercentSaving=0.25 smartOn=true
	36 type=Refrigerator - 19 cu. ft. - Frostfree loc=10000001 currentW=0 on?=false OnW=509 OffW=0 PercentSaving=0.25 smartOn=true
	31 type=Refrigerator - 14 cu. ft. - Frostfree loc=10000001 currentW=0 on?=false OnW=383 OffW=0 PercentSaving=0.25 smartOn=true
	25 type=Computer with Monitor loc=10000001 currentW=10 on?=false OnW=270 OffW=10 PercentSaving=0.5 smartOn=true
	28 type=Dishwasher (Without Dry Cycle) loc=10000001 currentW=0 on?=false OnW=200 OffW=0 PercentSaving=0.25 smartOn=true
	32 type=Laptop loc=10000001 currentW=10 on?=false OnW=75 OffW=10 PercentSaving=0.5 smartOn=true
		Test the getSmartAppsAtLoc() method

apps = pus3.getAppsOfType("oven");
for (Appliance a : apps) {
System.out.println(a); }
	113 type=Oven loc=10000006 currentW=5 on?=false OnW=5000 OffW=5
	134 type=Oven loc=10000007 currentW=5 on?=false OnW=5000 OffW=5
	178 type=Oven loc=10000010 currentW=5 on?=false OnW=5000 OffW=5
	245 type=Oven loc=10000014 currentW=5 on?=false OnW=5000 OffW=5
	355 type=Oven loc=10000019 currentW=5 on?=false OnW=5000 OffW=5
	400 type=Oven loc=10000022 currentW=5 on?=false OnW=5000 OffW=5
	458 type=Oven loc=10000026 currentW=5 on?=false OnW=5000 OffW=5
	902 type=Oven loc=10000051 currentW=5 on?=false OnW=5000 OffW=5
	1034 type=Oven loc=10000059 currentW=5 on?=false OnW=5000 OffW=5
	1059 type=Oven loc=10000060 currentW=5 on?=false OnW=5000 OffW=5
	1145 type=Oven loc=10000065 currentW=5 on?=false OnW=5000 OffW=5
	1279 type=Oven loc=10000073 currentW=5 on?=false OnW=5000 OffW=5
	1413 type=Oven loc=10000081 currentW=5 on?=false OnW=5000 OffW=5
	1439 type=Oven loc=10000082 currentW=5 on?=false OnW=5000 OffW=5
	1582 type=Oven loc=10000091 currentW=5 on?=false OnW=5000 OffW=5
	1637 type=Oven loc=10000094 currentW=5 on?=false OnW=5000 OffW=5
	1666 type=Oven loc=10000096 currentW=5 on?=false OnW=5000 OffW=5
	1732 type=Oven loc=10000100 currentW=5 on?=false OnW=5000 OffW=5
		Test the getAppsOfType() method

System.out.println(Arrays.toString(pus3.getLocs()));	[10000001, 10000002, 10000003, 10000004, 10000005, 10000006, 10000007, 10000008, 10000009, 10000010, 10000011, 10000012, 10000013, 10000014, 10000015, 10000016,	Test the getLocs() method

	10000017, 10000018, 10000019, 10000020, 10000021, 10000022, 10000023, 10000024, 10000025, 10000026, 10000027, 10000028, 10000029, 10000030, 10000031, 10000032, 10000033, 10000034, 10000035, 10000036, 10000037, 10000038, 10000039, 10000040, 10000041, 10000042, 10000043, 10000044, 10000045, 10000046, 10000047, 10000048, 10000049, 10000050, 10000051, 10000052, 10000053, 10000054, 10000055, 10000056, 10000057, 10000058, 10000059, 10000060, 10000061, 10000062, 10000063, 10000064, 10000065, 10000066, 10000067, 10000068, 10000069, 10000070, 10000071, 10000072, 10000073, 10000074, 10000075, 10000076, 10000077, 10000078, 10000079, 10000080, 10000081, 10000082, 10000083, 10000084, 10000085, 10000086, 10000087, 10000088, 10000089, 10000090, 10000091, 10000092, 10000093, 10000094, 10000095, 10000096, 10000097, 10000098, 10000099, 10000100]	
System.out.println(pus3.currentWattsAtLoc(10000002));	135	Test the getCurrentWattsAtLoc() method

pus3.turnOnByPercent();
System.out.println(pus3.currentWattsAtLoc(10000002));
	960	Test the turnOnByPercent() method

pus3.turnOffAllApps();
System.out.println(pus3.currentWattsAtLoc(10000002));
	135	Test the turnOffAllApps() method

pus3.turnOnByPercent();
System.out.println(pus3.totalCurrentWatts());
	154454	Test the totalCurrentWatts() method

System.out.println(pus3.percentAppsOnAtLoc(10000003));	0.2777777777777778	Test the percentAppsOnAtLoc() method