Team: Ali Guzelyel, Nathan Valdez, Brian Xu

Nouns and noun phrases:
Appliance
SmartAppliance
PowerUsageSimulationSystem
Type
Location
WattsOn
WattsOff
Probability
On
SmartOn
ID

Classes:

PowerUsageSimulationSystem

- Private int powerLimit;
- Private final int POWER_LIMIT = 50000;
- private final int DEFAULT_TIME_LENGTH = 24 * 60;
- Private final int DEFAULT_INTERVAL = 5;

ApplianceList

- private ArrayList<Appliance> appliances;
- private int sizeOfElements;
- private boolean empty;
- private int sizeOfArray;
- private final int DEFAULT_SIZE = 10000;
- private int probability

Appliance

- private static int counter = 1;
- private String type = "";
- private int wattsOff;
- private int wattsOn;
- private int applianceID;
- private boolean on;

SmartAppliance extends Appliance

- private double percentageSaving = 0;
- private boolean smartOn;

Operations:
Find Appliance (F):
Looks through the array list of objects and find a certain appliance
Add Appliance (A):
Adds an appliance to the array list
Delete Appliance (D) :
Deletes an appliance from the array list
Input File needed to add appliances, can change file used
**Input File Generation (ApplianceGenerator.java)
Users can add their own file, but it needs to be in a certain format or it re-prompts the user to type a different file.
PowerInDangerZone (this method checks each interval):

Run this method each interval time of the simulation. Find if the power exceeds the limit.
If it exceeds, call the FindExtremeLoc method.
Depending on ExtremeLoc's return, call TurnOffLocation on it.

ExtremeCaseLocation:

Look through the locations and find the extreme cases.
Return the smallest location between them.

TurnOffLocation:

Turn off smart appliances in the location. (Can't turn off normal appliances.)

Appliances can turn off/on:
When turned off/on, there is a probability of it being opened/closed. Implement that into simulation

- When the simulation is complete, output the following:
  - A summary reports to the screen the total number of locations affected in each interval for the entire simulation run, and the max affected location.
  - A detailed report to a file containing the appliances/locations were affected during each interval.

Subset:
User can interact with
Can find, add or delete appliances
Static count: total number of locations and number of appliances

6. The algorithm works like this:

While the warning level is true, find the location with the smallest smart appliance wattage usage and turn off the smart appliance. The smallest wattage is subtracted from the sum of the wattage to check if it is greater than the warning level wattage. This repeats until the sum is less than the warning wattage.

7.

Test Case (Appliance)	Test Case Reason:
Big WattsOn	Try big wattsOn so we know it can store up to that number
Small WattsOn	Try small wattsOn so we know it can store small numbers
Try valid and invalid type	Try valid type, try invalid type (invalid should return an error).
Big WattsOff	Try big wattsOff so we know it can store up to that number
Small WattsOff	Try small wattsOff so we know it can store small numbers

Test Case (ApplianceList)	Test Case Reason:
Large amount of appliances	Check if the program can make an arraylist containing a large amount of appliances
Small amount of appliances	Check if the program can make an arraylist containing a small amount of appliances
Valid and invalid .csv file	Check if the program can read valid csv files and when it receives an invalid one it asks for another one.
Find Appliances	Ensures that the program is able to locate the specified appliance within the arraylist
Add Appliances	Ensure that the program adds them into the arraylist

Delete Appliances	Ensure the program removes them from the arraylist
Sum	Ensure that method adds all the wattage of all appliances that are on.
PowerInDangerZone	Ensure that method Finds if the power usage is in danger zone
ExtremeCaseLocation	Ensure that method Finds the location that is using too much power
TurnOffLocation	Ensure that method turns off the location that is using too much power
	Check if Appliances can turn off/on by themselves

Test Case (SmartAppliance)	Reason
Try super() method for calling Appliance constructor	Inheritance tree, SmartAppliance inherits all the methods and attributes from Appliance
Try PercentageSaving	SmartAppliance appliance save wattages
Try SmartOn	Boolean to tell whether the SmartAppliance is on or off

Test Case (Simulator)	Reason
Try default time intervals	Ensure it works
Try custom time intervals	Ensure that we can use our own time intervals
Test if all called methods are working	methods:Add, Delete, Sum, PowerInDangerZone, ExtremeCaseLocation, TurnOffLocation
Invalid time	Can break the program

Questions:

Can we only turn off the smart appliances and not the normal ones?

What should the default power limit be?

Can appliances turn on/off by themselves?

Do we need to have an array list in the PowerUsageSimulationSystem?

Do we need to make an incursion or a for loop, for the simulation?