

CENG 305
Object Oriented Programming With Java
Fall 2013
HOMEWORK 2

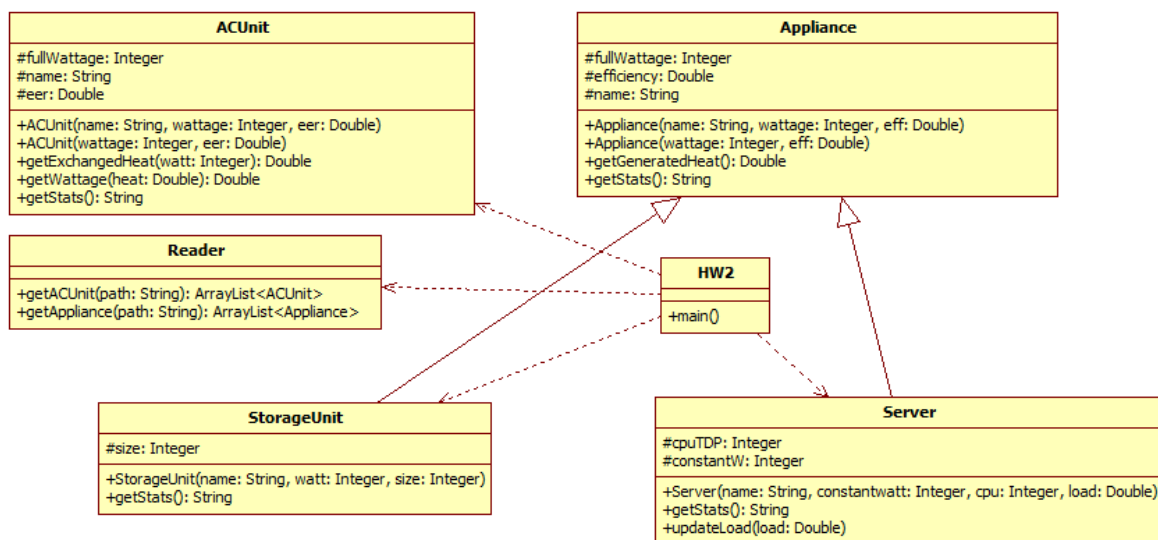
1. Regulations

- Due Date: 22 / 12 /2013
- Programming Language: You will use Java (version 1.6) to implement the assignment.
- Submission: Submit your homework through Cow (<https://cow.ceng.metu.edu.tr>). You will submit a ZIP file named "e1xxxxx.zip", which will include "Appliance.java" file, "ACUnit.java" file, "Server.java" file, "StorageUnit.java" file, "Reader.java".
- The homework should be done and submitted individually.
- Late Submission: You are allowed to submit your homework late with a penalty of 5*days*days.
- Cheating: The homework has to be done individually. In case of cheating, all involved (source(s) and receiver(s)) will get zero.
- Newsgroup: Check the metu.ceng.course.305 for discussions, announcements, etc. about the homework, regularly.
- Grading: Black box method will be used for evaluating and grading your assignments. Hence, please check sure that your codes can be compiled and run with the adequate output before submitting them. Furthermore, adequate indentation of the code, documentation and following submission rules are important since they will be considered in grading.

2. Specification

In this homework you will add classes and make small modification to the system you implemented for the previous homework.

You will drive two classes from the *Appliance* class, namely *Server* and *StorageUnit* classes. Also you will implement the *Reader* class. This class will read the specifications of the objects from a file and create appropriate objects.



Appliance class

Attributes

You should change the attributes to *protected*, then you will be able to change their values in the child classes.

Operation

The formula used in the *getGeneratedHeat()* This should be changed to $fullWattage \times (1 - efficiency) \times 3.41$. Although it is not still correct, at least it is more logical.

ACUnit Class

Attributes

You should change the attributes to *protected*, then you will be able to change their values in the child classes.

Server Class

Attributes

cpuTDP: Integer is the maximum power the cpu will use under normal circumstances.

constantWatt: Integer is the power used by the rest of the server.

Other attributes that comes from the base class will still be used.

Operation

Server(name: String, constantwatt: Integer, cpu: Integer, load: Double) Assigns the parameters to the appropriate attributes. For servers efficiency value is constant and it is 0.01. The power used by the server is $watt + (cpu \times load)$

updateLoad(load: Double) This function will update the power used by the system according to new load value.

getStats(): String This function overrides the super class and generates and returns a string in the following format where italic words signifies the values held in corresponding attributes:

Name: *name*

Watt (Constant): *constantW*

Efficiency: *efficiency*

Watt (CPU): *cpuTDP*

StorageUnit Class

Attributes

size: Integer is the maximum size of the storage unit.

Other attributes that comes from the super class will still be used.

Operation

StorageUnit(name: String, watt: Integer, size: Integer) Assigns the parameters to the appropriate attributes. Efficiency is taken to be 0.5 while calling the super class constructor.

getStats(): String This function overrides the super class and generates and returns a string in the following format where italic words signifies the values held in corresponding attributes:

Name: *name*

Watt: *fullWattage*

Efficiency: *efficiency*

Size: *size*

Reader Class

Operation

`getACUnit(path: String): ArrayList<ACUnit>` This function generate a list of `ACUnit` objects read from a file whose path given in the parameter `path`. An example file will be given.

`getAppliance(path: String): ArrayList<Appliance>` This function generate a list of `Appliance` objects read from a file whose path given in the parameter `path`. An example file will be given.

Both functions are *static*.

HW1 Class

This class will only contain the main function. You can use the following function.

```
public static void main(String[] args) {
    try{
        ArrayList<Appliance> l1=Reader.getAppliances("txt.txt");
        ArrayList<ACUnit> l2=Reader.getACUnit("txt.txt");
        for (Appliance a : l1) {
            System.out.println(a.getStats());
        }
        for (ACUnit a : l2) {
            System.out.println(a.getStats());
        }
    }
    catch (Exception e)
    {
        System.out.println(e.getMessage());
    }
}
```

3. Example File

```
Server
serverbir
500
50
0.30
StorageUnit
storageunitbir
500
500
ACUnit
klimabir
500
11.0
```

Sample output:

```
Name: serverbir
Watt (Full): 515
Efficiency: 0.01
Watt (CPU): 50
```

Name: storageunitbir
Watt: 500
Efficiency: 0.5
Size: 500

Name: klimabir
Watt: 500
EER: 11.0

4. Submission

You will submit a single exxxxxxx.zip file. File name have to be your user name. Only your class files ("Appliance.java", "ACUnit.java" "Server.java", "StorageUnit.java", "Reader.java") and generated java documentation have to be in this .zip file. All the package, class, attribute and operation names have to be **exactly** the same with the names given in this document. If you do not follow the submission rules a penalty will be applied to your grade.