

EE333 P4–5 Spring 2019

Purpose

The purpose of this project is to produce a **Object-Oriented Design** and demonstration **Java program** that simulates a smart light system with a controller, hub and smart light bulbs.

Overview

The goal of this project is to demonstrate your ability to do **object-oriented design** and then implement a demonstration of the quality of **that design**. The discussion that follows supplies *some* input necessary to create the object model but more discovery is likely necessary.

Figure 1 shows the conceptual block diagram for the lighting system.

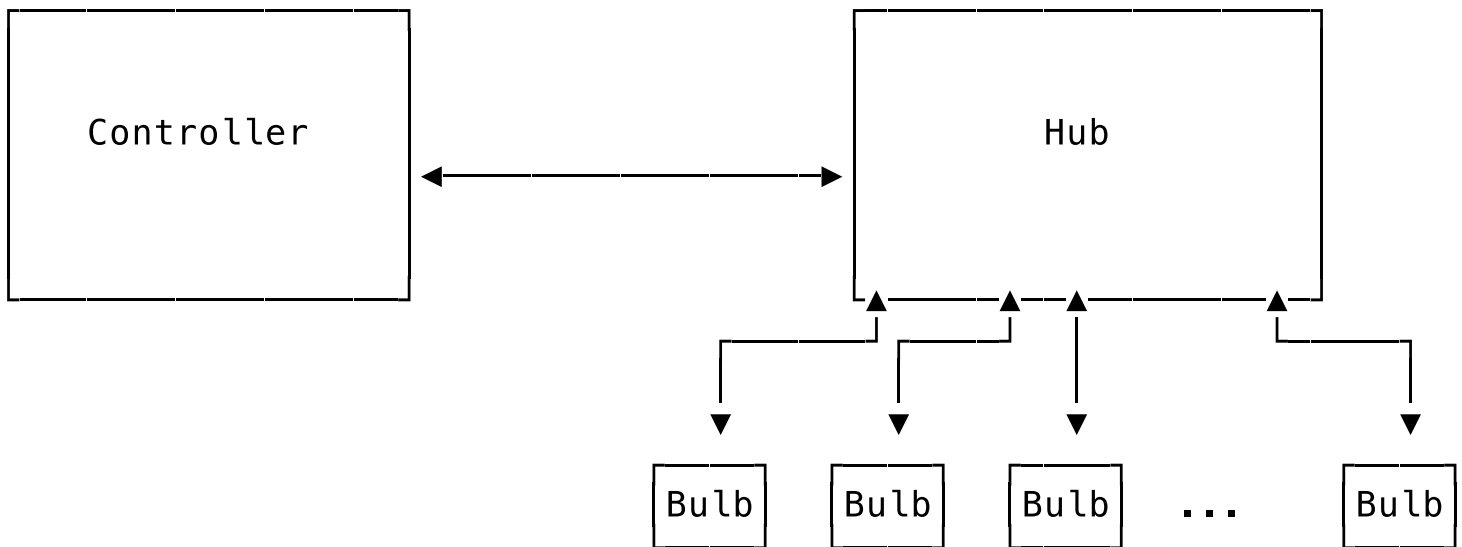


Figure 1. Conceptual Block Diagram of light system

Discussion

The system shall consist of a controller, hub and multiple smart light bulbs. The controller talks to the hub (bridge) which talks to the light bulbs. The lights shall have addressing to allow the control of lights. The modules are connected in a broadcast manner where all modules hear all the messages and addressing is used to determine the parties of the conversation.

The simulation should be such that the simulated controller could be reimplemented to talk to a real hub and lights system without changing the architecture. The controller is where there is an opportunity for near endless capabilities including **configuration, turning on/off bulbs, setting bulb color, setting bulb brightness, defining bulbs in groups and manipulating same**, defining moods to apply to bulbs or groups, doing actions on a time-basis, doing actions on an event basis, and having a sequence of activities. The **bold** items are considered the target for a base implementation of the simulation.

There are several implementations of hubs and smart lights on the market. The [Phillips product architecture](#) is one example.

Note: you are expected to have an object-oriented design and a matching implementation with a demonstration of quality. You may have to reduce scope in implementation but do not eliminate the object-oriented nature of the design nor the demonstration of quality.

Note: remember don't put specific I/O in (domain) model classes.

Obey Java Documentation Style

Continue to use the specified [documentation standard for Java source code](#). Ensure that the Javadoc comments contain

1. class responsibilities
2. class and instance behaviors

Pre- P4 submission Checklist

- ☐ Review your document to ensure all sections are covered and that you are submitting the required design evidence
- ☐ Check the phrasing of Constraints, Goals, Standards, MoSCoW list items to ensure they comply with examples
- ☐ Check that many design issues are identified, have multiple realistic alternatives, and that the chosen alternative is defended (usually with reference to supporting goals)
- ☐ Review the Rubric to ensure you have completed all items being reviewed
- ☐ Ensure that the submitted .zip file is correctly named
- ☐ Ensure that the .zip file will unzip into the correct folder name

P4 Delivery

You shall deliver an electronic copy of the files detailing your designed classes and the draft P5 report to Canvas. As a minimum, the submission should include:

1. Draft P5 Report including at least
 1. Relationships between classes
 2. Key interactions
 3. UML Diagrams expressing the above (at least one class diagram, one state diagram, and two interaction diagrams are minimum requirements)
 4. Statement of design problem
 - Goals
 - Constraints
 - Standards
 - MoSCoW list
 5. Design process
 6. Design alternatives thus far
2. Data 'maintained' by the class (will end up in code documentation)
3. Public methods to implement functionality (will end up as code)
4. One or more lines of text describing the data and methods. (will end up as JavaDoc comments)

Sections in the report that do not apply to the design phase (such as debugging) should be labeled TBD.

P4 is due at the start of class. After P4 is discussed in class, no further submissions will be accepted.

Word and LaTeX templates for the report are available.

Pre- P5 submission Checklist

- ☐ Ensure that all sections are covered and that evidence supporting the skill you are demonstrating is provided
- ☐ Ensure you have completed all items being reviewed by the Rubric
- ☐ Ensure that you have addressed the criticism of your P4 report
- ☐ Ensure all sections are covered and that you submitting the required design evidence
- ☐ Check the phrasing of Constraints, Goals, Standards, MoSCoW list items to ensure they comply with examples
- ☐ Ensure that your UML drawings match your code (if not, change one or the other)
- ☐ Check that many design issues are identified, have multiple realistic alternatives, and that the chosen alternative is defended (usually with reference to supporting goals)
- ☐ Ensure you have captured a sample run as requested
- ☐ Review sample run to ensure it is correct (and create a file discussing the output)
- ☐ Ensure program code has the required documentation (including proper use of `/**` vs `/*`)
- ☐ Ensure that the submitted .zip is correctly named
- ☐ Ensure that the .zip file will unzip into the correct folder name

P5 Report

1. Project definition based on specifications, constraints, goals, and applicable standards
2. Is this a good problem for object-oriented solution? Why or why not?
3. Alternatives considered in designing the project
4. Alternatives selected and why
5. Final object approach
 1. Relationships between classes
 2. Key interactions
 3. UML Diagrams expressing the above (at least one class diagram, one state diagram, and two interaction diagrams are minimum requirements)
 4. MoSCoW list
6. Process - the approach you chose to follow in developing your solution (beginning with receipt of assignment). You are trying to show you know how to do software design and development.
7. Discovery and use of online information - discuss at least one instance where you had to search, understand, and use information from an online source that was not furnished in the course infrastructure. Be sure to relate how you performed the search, obtained an understanding, as well how the information was used.
8. Debug - discuss one of the problems you had in implementation and the steps you took to debug the program
9. Results - how you tested your model and how well it worked. Relate to item 1.
0. Include actual data (listings of runs) to show your project's execution.
1. How would you do things different next design project?

P7 is expected to be of the quality that you would use the document as an example to a potential employer of the type of work that you can do.

Word and LaTeX templates for the report are available.

The makers of [LucidChart](#), a web-based tool for drawing things including UML diagrams, in an effort to support education and students, have made the premium version of their product available to students with @uab.edu addresses.

Bonus (for P5, up to 25 points on 100 point scale)

Deliver unit tests for all modeling classes in addition to fulfilling other requirements.

Bonus (for P5, up to 25 points on 100 point scale)

Design and implement one or more of the additional behaviors of the controller.

P5 Delivery

You shall produce source code that complies with the documentation standards. Your program **MUST** show your name and BlazerID near the top of source code files and display out your name and BlazerID at the start of the test run. Produce a `blazerid-p7.zip` file containing a directory `blazerid-p7` which contains all files (at least the report file, Javadoc files, `.java` and `.class` files) and submit it using the assignment tool of *Canvas*.

Recall that there are COURSE PERFORMANCE REQUIREMENTS RELATED TO THIS ASSIGNMENT.

Last Updated: 5 February 2019