Overview

The goal of this assignment is to create a Java application to manage “to-do lists”. Your application should support three specific types of to-do items: **tasks** containing a priority (either “URGENT”, “IMPORTANT”, or “SOMEDAY”); **meetings** containing a location and time; and **deadlines** containing a due date. Your to-do items will be stored and managed by a to-do planner. Every to-do item within a planner must have a unique title. The planner should allow you to add, remove, and retrieve to-do items. You’ll compile and run your application using the commands “javac TodoPlannerTextUI.java” and “java TodoPlannerTextUI”, respectively. The TodoPlannerTextUI is a helper class that implements a text-based user-interface for your planner application. An example run of your program should look something like this:

```
Welcome to the Todo Planner!
--------------------------------
Main Menu:
1. Create a Task
2. Create a Meeting
3. Create a Deadline
4. Remove a Todo Item
5. Display Todo Items
6. Exit
Choice: 1

Please enter a task title: Haircut
Please enter a priority (1-Urgent, 2-Important, 3-Someday): q
Oops, please try again: 2
Added Task:
      Haircut (Priority: Important)

Main Menu:
1. Create a Task
2. Create a Meeting
3. Create a Deadline
4. Remove a Todo Item
5. Display Todo Items
6. Exit
Choice: exit
Oops, please try again: 2

Please enter a meeting title: Study group
Please enter a meeting location: CS Lab
Please enter a meeting time: Jan. 29 at 3:30
Added Meeting:
        Study group (Location: CS Lab, Time: Jan. 29 at 3:30)
```

Main Menu:
1. Create a Task
2. Create a Meeting
3. Create a Deadline
4. Remove a Todo Item
5. Display Todo Items
6. Exit
Choice: 6

Goodbye!

You can treat the time format as a plain string (for meetings and deadlines). If the user enters a to-do item that has the same title as one already entered, then you should tell the user that the title already exists and prompt for a new title. Displaying to-do items should print the items in the order they were created, e.g.,

1. Haircut (Priority: Important)
2. Study group (Location: CS Lab, Time: Jan. 29 at 3:30)

When a user chooses to remove an item, first display all the items and then prompt the user for an item number to remove. All prompts in your program should check that valid input was entered.

Instructions

You will need to create six classes to implement your program: TodoPlannerTextUI, TodoPlanner, Todo, Task, Meeting, and Deadline. The TodoPlanner holds and manages a list of to-do items. The Todo class is a base class for other planner items and simply holds a title. The Task, Meeting, and Deadline classes each extend (i.e., are subclasses of) Todo, and each add additional fields and methods. Finally, the TodoPlannerTextUI is the application class (it has the main() method) and should handle all of the user-interface functionality for the TodoPlanner class. You should not include any user interface code in any of the classes other than TodoPlannerTextUI.

Part of your job is to determine the methods needed for each of these classes. It is recommended that you create a preliminary design before you start coding (e.g., identify the methods and write simple “stub” implementations). In your design, you should make sure each method does (essentially) one task. You also need to modularize your TodoPlannerTextUI class, so that it is not one large main method, but instead calls separate helper methods.

Test your program to ensure it works correctly. Make sure you try a sufficient number of cases, and provide the results of your test runs. Create a directory named hw2 and place your class source (.java) files in it, and submit this directory using the class submission system. Hand in hard-copy consisting of a cover sheet, your design document, your source code, and your test runs showing your program works correctly in class on (or before) the due date.