

Todoist

Getting things done is an art. Most of us use a todo list to organize our tasks. A todoist is a java program to keep track of a list of tasks. A task in the todo list has the following properties.

1. a task title (String)
2. name of the person who should do the task (String)
3. time (in minutes) required to complete the task (int)
4. task is important or not important (boolean - true if important, false otherwise)
5. task is urgent or not urgent (boolean - true if urgent, false otherwise)
6. Task is pending or completed. (String - todo if pending, done if completed)

Question 1 (2 marks)

Create a class named Task with its 6 properties. Add an overloaded constructor with the parameters String title, String assignedTo, int timeToComplete, boolean important, boolean urgent, String status. Also, add a toString method to display the Task object.

The first 3 test case files input000, input001, input002 must pass for the 2 marks to be awarded. Main method is already provided and handles the input and output.

Question 2 (2 marks)

Check for errors in the values set for the properties of the Task object. The error validation should be performed in the constructor. In case of error, constructor should throw an Exception with an error message. There are 3 validations to be performed.

1. Task title should not be empty
2. Time required to complete the task should be non negative value
3. Task status should be either todo or done and can't have any other value.

The error messages for the above validations are:

1. Title not provided
2. Invalid timeToComplete <time value>
3. Invalid status <status value>

3 test case files input003, input004, input005 must pass for the 2 marks to be awarded. Main method is already provided and handles the input and output.

Question 3 (2 marks)

Create a class called Todoist. It should have an array of the type Task to store the Task objects. The constructor of the Todoist should initialize the array. Add a public method add that takes a Task object as a parameter. The add method should add the given task object to the array. If the array is full then resize the array.

Add a toString method to the Todoist class to print the list of all the tasks. The print format is one task per line. For each task print its title, name of the person, time to complete, Important or Not Important, Urgent or Not Urgent, task status i.e, todo or done.

2 test case files input006, input007 must pass for the 2 marks to be awarded. Main method is already provided and handles the input and output.

Question 4 (2 marks)

Add a public method getNextTask with a String parameter and Task object as return type. The parameter is the name of the person. The goal of this method is to return the next task todo for the person given in the parameter. There are two criteria for a task to qualify as the “next task”.

1. The task should be assigned to the given person, status of the task should be todo, it should be important and not urgent. There could be multiple tasks that qualify this criteria. Return the first task in the list that matches. If there are no such tasks then look for the next point.
2. The task should be assigned to the given person, status of the task should be todo, it should be important and urgent. Again, there could be multiple tasks that qualify this criteria. Return the first task in the list that matches.
3. If there are no such tasks then return null.

2 test case files input008, input009 must pass for the 2 marks to be awarded. Main method is already provided and handles the input and output.

Question 5 (1 mark)

Add a public overloaded method getNextTask with a String parameter and a integer parameter. String parameter is the name of the person and int parameter is the count of tasks. Return should be an array of type Task and the size of the return array should be count (i.e., passed as an argument to the method). The goal of this method is similar to the one in Question 4 with a simple extension to return not just one next task but count number of next tasks for the given person.

Test case file input010 must pass for the 1 mark to be awarded. Main method is already provided and handles the input and output.

Question 6 (1 mark)

Add a public method totalTime4Completion that returns an int. The goal of the method is to add up all the time to complete properties of the tasks that are todo and return the total.

Test case files input011, input012, input013 must pass for the 1 mark to be awarded. Main method is already provided and handles the input and output.