findNextDueTask() – No Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Expected Result |
| TaskList<Task> list = new TaskList<Tast> | To create an empty list of tasks | Size = 0 | An empty TaskList |
| List.findNextDueTask() | To find the task(s) with the earliest due date | Size = 0 | Returns null; |

findNextDueTasks() – Single Personal Task

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Expected Result |
| TaskList<Task> list = new TaskList<Tast> | To create an empty list of tasks | Size = 0 | An empty TaskList |
| PersonalTask task1 = new PersonalTask(“Appointment”, new GregorianCalendar(2016, 11, 01), 1, “oil change”, “garage”)); | To create a new personaltask |  |  |
| list.addTask(task1); | To add a personal task to the list | Size = 1  Head / tail = “  Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“ |  |
| TaskList<Task> nextDue = list.findNextDueTask() | To find the task(s) with the earliest due date |  | Returns  “Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage “ |
| nextDue.get(0).equals(task1) | To verify that the list contains the proper tasks |  | true |

findNextDueTask() – single homework task

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Expected Result |
| TaskList<Task> list = new TaskList<Tast> | To create an empty list of tasks | Size = 0 | An empty TaskList |
| HomeworkTask task1 = new HomeworkTask(“Assignment”, new GregorianCalendar(2016, 11, 01), 1, “420-B31”, “Sandra Stark”, “Assignment 2”, “Create a to-do list”)); | To create a new homework task |  |  |
| list.addTask(task1); | To add a Homework task to the list | Size = 1  Head / tail =  “Course Number: 420-B31  Teacher Name: Sandra Stark  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016  Task Description: Create a to-do list” |  |
| TaskList<Task> nextDue = list.findNextDueTask() | To find the task(s) with the earliest due date |  | Returns  “  Course Number: 420-B31  Teacher Name: Sandra Stark  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016  Task Description: Create a to-do list” |
| nextDue.get(0).equals(task1) | To verify that the list contains the proper tasks |  | true |

findNextDueTask() – Multiple Personal Task

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Expected Result |
| TaskList<Task> list = new TaskList<Tast> | To create an empty list of tasks | Size = 0 | An empty TaskList |
| PersonalTask task1 = new PersonalTask(new PersonalTask(“Appointment”, new GregorianCalendar(2016, 11, 01), 1, “oil change”, “garage”)); | To create a new PersonalTask |  |  |
| PersonalTask task2 = new PersonalTask(“Errand”, new GregorianCalendar(2016, 11, 01), 1, “groceries”, “Walmart”)); | To create a new PersonalTask |  |  |
| PersonalTask task3 = new PersonalTask( “Other”, new GregorianCalendar(2016, 12, 01), 1, “Walk Dog”)); |  |  |  |
| list.addTask(task1); | To add a personal task to the list | Size = 1  Head / tail = “  Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“ |  |
| list.addTask(task2); | To add a personal task to the list | Size = 2  Head =  Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“  Tail = “  Task Type: Errand  Task Name: Groceries  Priority: 1  Due Date: November 1, 2016  Task Location: Walmart“ |  |
| list.addTask(task3); | To add a personal task to the list | Size = 3  Head =  “Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“  “Task Type: Errand  Task Name: Groceries  Priority: 1  Due Date: November 1, 2016  Task Location: Walmart“  Tail = “  Task Type: Other  Task Name: Walk Dog  Priority: 1  Due Date: December 1, 2016“ |  |
| TaskList<Task> nextDue = list.findNextDueTask() | To find the task(s) with the earliest due date |  | Returns  “Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“  “Task Type: Errand  Task Name: Groceries  Priority: 1  Due Date: November 1, 2016  Task Location: Walmart“ |
| nextDue.get(0).equals(task1) | To verify that the list contains the proper task |  | true |
| nextDue.get(1).equals(task2) | To verify that the list contains the proper task |  | true |

findNextDueTask() – Multiple HomeworkTask

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Expected Result |
| TaskList<Task> list = new TaskList<Tast> | To create an empty list of tasks | Size = 0 | An empty TaskList |
| HomeworkTask task1 = new HomeworkTask(“Assignment”, new GregorianCalendar(2016, 11, 01), 1, “420-B31”, “Sandra Stark”, “Assignment 2”, “Create a to-do list”)); | To create a new HomeworkTask |  |  |
| HomeworkTask task2 = new HomeworkTask(“Assignment”, new GregorianCalendar(2016, 11, 01), 1, “420-C30”, “Allan McDonald”, “Assignment 2”)); | To create a new HomeworkTask |  |  |
| HomeworkTask task3 = new HomeworkTask(“Assignment”, new GregorianCalendar(2016, 12,01 ), 1, “420-E11”, “Nathalie Gaul”, “Assignment 3”)); | To create a new HomeworkTask |  |  |
| list.addTask(task1); | To add a Homework task to the list | Size = 1  Head / tail =  “Course Number: 420-B31  Teacher Name: Sandra Stark  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016  Task Description: Create a to-do list” |  |
| list.addTask(task2); | To add a Homework task to the list | Size = 2  Head =  “Course Number: 420-B31  Teacher Name: Sandra Stark  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016  Task Description: Create a to-do list”  Tail =  “Course Number: 420-C30  Teacher Name: Allan McDonald  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016 |  |
| list.addTask(new HomeworkTask(task3); | To add a Homework task to the list | Size = 3  Head =  “Course Number: 420-B31  Teacher Name: Sandra Stark  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016  Task Description: Create a to-do list”  “Course Number: 420-C30  Teacher Name: Allan McDonald  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016”  Tail =  “Course Number: 420-E11  Teacher Name: Nathalie Gaul  Task Type: Assignment  Task Number: Assignment 3  Priority: 1  Due Date: December 1, 2016” |  |
| TaskList<Task> nextDue = list.findNextDueTask() | To find the task(s) with the earliest due date |  | Returns  “Course Number: 420-B31  Teacher Name: Sandra Stark  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016  Task Description: Create a to-do list”  “Course Number: 420-C30  Teacher Name: Allan McDonald  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016” |
| nextDue.get(0).equals(task1); | To verify that the list contains proper task |  | True |
| nextDue.get(1).equals(task2) | To verify that the list contains proper task |  | True |

findNextDueTask() – multiple mixed tasks

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Expected Result |
| TaskList<Task> list = new TaskList<Tast> | To create an empty list of tasks | Size = 0 | An empty TaskList |
| PersonalTask task1 = new PersonalTask(“Appointment”, new GregorianCalendar(2016, 11, 01), 1, “oil change”, “garage”)); | To create a new PersonalTask |  |  |
| PersonalTask task2 = new PersonalTask(“Errand”, new GregorianCalendar(2016, 11, 01), 1, “groceries”, “Walmart”)); | To create a new PersonalTask |  |  |
| HomeworkTask task3 = new HomeworkTask(“Assignment”, new GregorianCalendar(2016, 11, 02), 1, “420-B31”, “Sandra Stark”, “Assignment 2”, “Create a to-do list”)); | To create a new HomeworkTask |  |  |
| PersonalTask task4 = new PersonalTask(“Other”, new GregorianCalendar(2016, 12, 01), 1, “Walk Dog”)); | To create a new PersonalTask |  |  |
| HomeworkTask task5 = new HomeworkTask(“Assignment”, new GregorianCalendar(2016, 11, 01), 1, “420-C30”, “Allan McDonald”, “Assignment 2”)); | To create a new HomeworkTask |  |  |
| HomeworkTask task6 = new HomeworkTask(“Assignment”, new GregorianCalendar(2016, 12,01 ), 1, “420-E11”, “Nathalie Gaul”, “Assignment 3”)); | To create a new HomeworkTask |  |  |
| list.addTask(task1); | To add a personal task to the list | Size = 1  Head / tail = “  Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“ |  |
| list.addTask(task2); | To add a personal task to the list | Size = 2  Head =  Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“  Tail = “  Task Type: Errand  Task Name: Groceries  Priority: 1  Due Date: November 1, 2016  Task Location: Walmart“ |  |
| list.addTask(task3); | To add a Homework task to the list | Size = 3  Head =  Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“  “Task Type: Errand  Task Name: Groceries  Priority: 1  Due Date: November 1, 2016  Task Location: Walmart“  Tail =  “Course Number: 420-B31  Teacher Name: Sandra Stark  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 2, 2016  Task Description: Create a to-do list” |  |
| list.addTask(task4); | To add a personal task to the list | Size = 4  Head =  Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  “Task Type: Errand  Task Name: Groceries  Priority: 1  Due Date: November 1, 2016  Task Location: Walmart“  “Course Number: 420-B31  Teacher Name: Sandra Stark  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 2, 2016  Task Description: Create a to-do list”  Tail =  “Task Type: Other  Task Name: Walk Dog  Priority: 1  Due Date: December 1, 2016“ |  |
| list.addTask(task5); | To add a Homework task to the list | Size = 5  Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“  “Task Type: Errand  Task Name: Groceries  Priority: 1  Due Date: November 1, 2016  Task Location: Walmart“  “Course Number: 420-B31  Teacher Name: Sandra Stark  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 2, 2016  Task Description: Create a to-do list”  “Task Type: Other  Task Name: Walk Dog  Priority: 1  Due Date: December 1, 2016“  Tail =  “Course Number: 420-C30  Teacher Name: Allan McDonald  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016 |  |
| list.addTask(task6); | To add a Homework task to the list | Size = 6  Head =  Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“  “Task Type: Errand  Task Name: Groceries  Priority: 1  Due Date: November 1, 2016  Task Location: Walmart“  “Course Number: 420-B31  Teacher Name: Sandra Stark  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 2, 2016  Task Description: Create a to-do list”  “Task Type: Other  Task Name: Walk Dog  Priority: 1  Due Date: December 1, 2016“  “Course Number: 420-C30  Teacher Name: Allan McDonald  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016  Tail =  “Course Number: 420-E11  Teacher Name: Nathalie Gaul  Task Type: Assignment  Task Number: Assignment 3  Priority: 1  Due Date: December 1, 2016” |  |
| TaskList<Task> nextDue = list.findNextDueTask() | To find the task(s) with the earliest due date |  | Returns  Task Type: Appointment  Task Name: oil change  Priority: 1  Due Date: November 1, 2016  Task Location: garage“  “Task Type: Errand  Task Name: Groceries  Priority: 1  Due Date: November 1, 2016  Task Location: Walmart“  “Course Number: 420-C30  Teacher Name: Allan McDonald  Task Type: Assignment  Task Number: Assignment 2  Priority: 1  Due Date: November 1, 2016 |
| nextDue.get(0).equals(task1); | To verify that the list contains proper task |  | True |
| nextDue.get(1).equals(task2); | To verify that the list contains proper task |  | True |
| nextDue.get(2).equals(task5) | To verify that the list contains proper task |  | True |