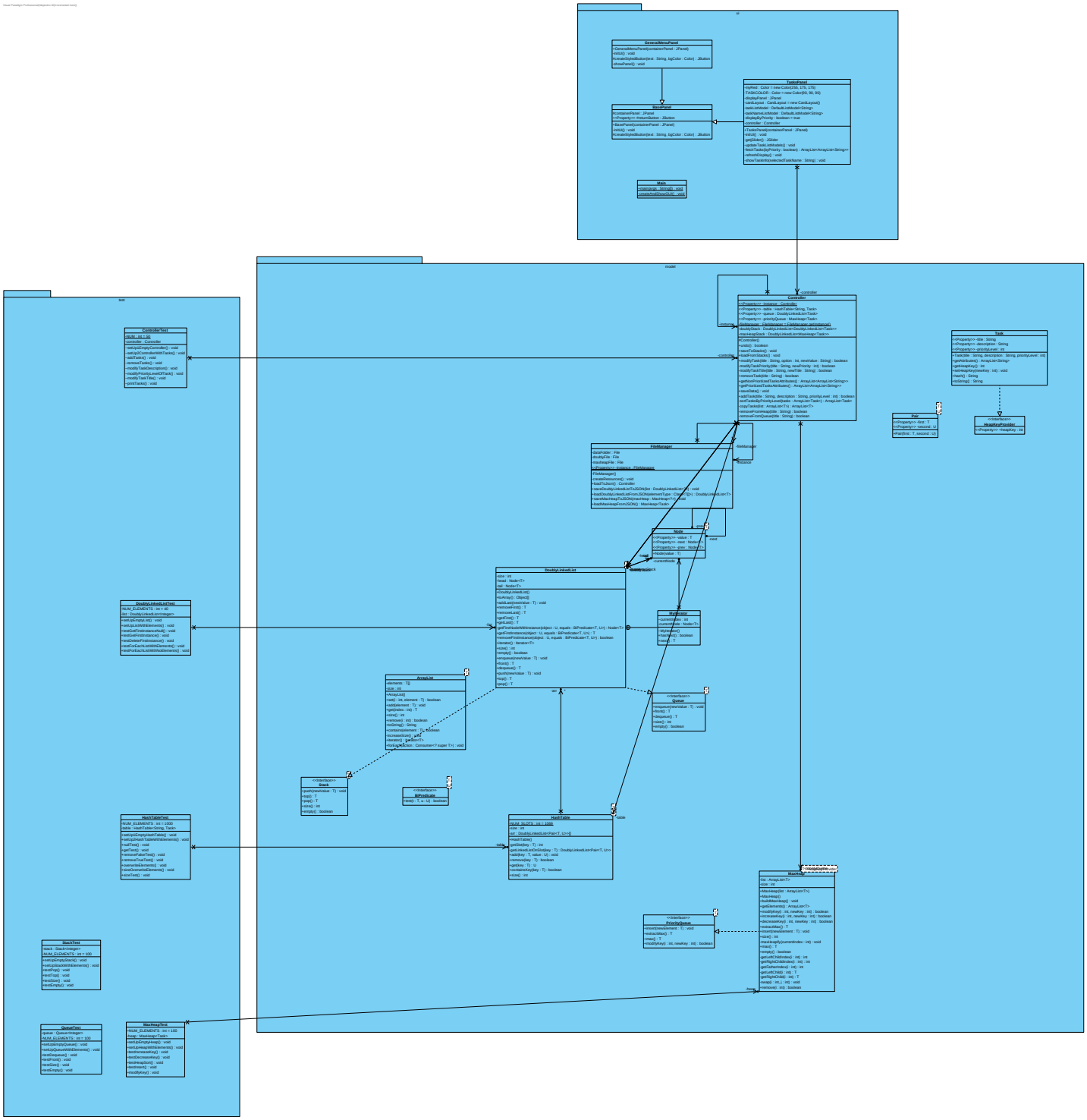
Visual Paradigm Professional(Objectiv B)(Licensed-Ivan)


ui
GeneralMenuPanel
+GeneralMenuPanel(containerPanel : JPanel)
+initUI() : void
+createStyledButton(text : String, bgColor : Color) : JButton
+showPanel() : void
BasePanel
#containerPanel : JPanel
#<<Property>> #returnButton : JButton
+BasePanel(containerPanel : JPanel)
+initUI() : void
+createStyledButton(text : String, bgColor : Color) : JButton
TasksPanel
-myRed : Color = new Color(216, 173, 173)
-TASKCOLOR : Color = new Color(90, 90, 90)
-displayPanel : JPanel
-cardLayout : CardLayout = new CardLayout()
-taskListModel : DefaultListModel<String>
-taskNameListModel : DefaultListModel<String>
-isDisplayByPriority : boolean = true
-controller : Controller
+TasksPanel(containerPanel : JPanel)
-initUI() : void
-getGlass() : JGlider
-updateTasksListModels() : void
-fetchTasksByPriority() : boolean[]
-refreshDisplay() : void
-showTasksInsideSelectedTaskName(taskName : String) : void
Main
+main(args : String[]) : void
-createAndShowGUI() : void
model
Controller
<<Property>> -instance : Controller
<<Property>> -table : HashTable<String, Task>
<<Property>> -queue : DoublyLinkedList<Task>
<<Property>> -priorityQueue : MaxHeap<Task>
-fileManager : FileManager = FileManager.getInstance()
-doublyStack : DoublyLinkedList<DoublyLinkedList<Task>>
-maxHeapStack : DoublyLinkedList<MaxHeap<Task>>
+Controller()
+setup() : boolean
+saveToStacks() : void
+loadFromStacks() : void
+modifyTaskTitle(title : String, option : int, newValue : String) : boolean
+modifyTaskPriority(title : String, newPriority : int) : boolean
+modifyTaskTitleDate(title : String, newTitle : String) : boolean
+removeTaskTitle(title : String) : boolean
+getNonPrioritizedTasksAttributes() : ArrayList<ArrayList<String>>
+getPrioritizedTasksAttributes() : ArrayList<ArrayList<String>>
+saveData() : void
+addTaskTitle(title : String, description : String, priorityLevel : int) : boolean
+sortTasksByPriorityLevel(tasks : ArrayList<Task>) : ArrayList<Task>
+copyTasks(list : ArrayList<T>) : ArrayList<T>
+removeFromHeap(title : String) : boolean
+removeFromQueue(title : String) : boolean
-controller
-controller
-instance
Task
<<Property>> -title : String
<<Property>> -description : String
<<Property>> -priorityLevel : int
+Task(title : String, description : String, priorityLevel : int)
+getAttributes() : ArrayList<String>
+getHeapKey() : int
+setHeapKey(newKey : int) : void
+hash() : String
+toString() : String
Pair
<<Property>> -first : T
<<Property>> -second : U
+Pair(first : T, second : U)
<<interface>>
HeapKeyProvider
<<Property>> +heapKey : int
FileManager
-dataFolder : File
-doublyFile : File
-maxHeapFile : File
-<<Property>> -instance : FileManager
-FileManager()
+getInstance() : void
+loadToJson() : Controller
+saveDoublyLinkedListToJSON(list : DoublyLinkedList<T>) : void
+loadDoublyLinkedListFromJSON(elementClass : Class<T>) : DoublyLinkedList<T>
+saveMaxHeapToJSON(maxHeap : MaxHeap<?>) : void
+loadMaxHeapFromJSON() : MaxHeap<Task>
-fileManager
-instance
Node
<<Property>> -value : T
<<Property>> -next : Node<T>
<<Property>> -prev : Node<T>
+Node(value : T)
-next
-next
-prev
-prev
-head
-tail
-currentNode
-currentNode
DoublyLinkedList
-size : int
-head : Node<T>
-tail : Node<T>
+DoublyLinkedList()
+toArray() : Object[]
+addLast(newValue : T) : void
+removeFirst() : T
+removeLast() : T
+getFirst() : T
+getLast() : T
+getFirstNodeWhiteInstance(object : U, equals : BiPredicate<T, U>) : Node<T>
+getFirstInstance(object : U, equals : BiPredicate<T, U>) : T
+getLastInstance(object : U, equals : BiPredicate<T, U>) : T
+iterator() : Iterator<T>
+size() : int
+empty() : boolean
+enqueue(newValue : T) : void
+front() : T
+dequeue() : T
+push(newValue : T) : void
+top() : T
+pop() : T
MapEntry
-currentValue : int
-currentNode : Node<T>
+MapEntry()
+hasNext() : boolean
+next() : T
-size
-size
-size