Final Project - Task management system

The task management system is a small-scale application that operates by creating and managing tasks with specific due dates and levels of priority. The user will be able to create tasks and add them into a list or place them in a queue. They can then be sorted according to their priority, their due date, or a mix of both. Tasks can be set as completed and regular tasks will have their new due date refreshed according to the time interval between them. When obtaining the queue of tasks to do, completed non regular tasks will be ignored because they will not need to be done anymore. The system will be able to export the tasks into a CSV file and re-import a list of tasks from a CSV file. The user can also search for tasks with keywords in the title and a filter for its status (if it is completed or not).

- The Task class will be an abstract class with two children being the RegularTask and NonRegularTask class.
- The interface TaskOperations is going to contain abstract methods for managing a list of tasks (adding, updating a task, sorting) and will be necessary for the TaskManager class, which will manage a list of tasks.
- Besides the equals and tostring methods, runtime polymorphism will be used for the exportTasks method in the TaskManager, as it will simply call the getExportData method of Task to obtain the CSV format data.
- TextIO will be used in the TaskManager class as it will read and write data for Task objects in a CSV file.
- Comparable will be implemented for Task, which will compare the priority of the tasks.
- A TaskComparator class will implement Comparator<Task> to compare Task objects according to their priority, due date, or both.
- All classes will have getters, setters, equals, and tostring methods for their respective fields.

For deliverable 2, the Task class and its subclasses RegularTask and NonRegularTask will be implemented (including their methods and TaskComparator).

UML Class Diagram