Main features to do:

- Crud operation for task
- Task categorization
- Filtering based on priority or due date
- Concurrent processing for reports (total task, completed task)
- Exception handling for inputs
-

Inputs from user:

- Title: String
- Description: String
- Priority: Enum(low, medium, high)
- Due Date: Date
- Completed: Boolean (if the task is completed or not)
- Category: String

Event-driven menu:

- Add Task
- View Tasks (All or by category or by due dates)
- Update Task
- Delete Task
- Category of task
- Filter Tasks (Priority/Due Date)
- Completed (just select option to complete it initially it was false)
- Generate Report
- Exit

Exception Handling:

- Invalid input: can't be empty
- Invalid dueDate: should be future date
- Invalid priority: other than low, medium, high
- TaskNotFound: attempt to update or delete task that don't exist

- Add task to arraylist
- First find unique category and add category to hashmap<category, list of task>
- Now use comparator with lambda function to filter with due date
- To filter by priority use lambda function and stream api
- Now use threads concept for generating reports for total task and completed task