Kuljit Takhar

Term Project Proposal

November 11 2023

Project Proposal:

EAT THE FROG - Task Manager Application

EAT THE FROG is a task management application developed in Java using the Swing library, designed to implement the time management strategy advocating the completion of the most challenging task first. Rooted in a quote attributed to Mark Twain – "Eat a live frog first thing in the morning, and nothing worse will happen to you the rest of the day” — the application provides users with a powerful tool for efficient task organization and increased productivity.

Project Overview:

The application implements the ArrayList class as the central data structure for task management. This list facilitates the dynamic addition, removal, search, and sorting of tasks, prioritizing them based on their assigned value and due date.

The code utilizes Swing components (JFrame, JList, JTextField, JTextArea, JButton) to create an interactive user interface. Event-driven programming handles user actions like adding, removing, completing, sorting, searching, and editing tasks. Sorting algorithms maintain the task list in ascending priority order. A mouse event listener displays the title and description of task, allowing the user to read or edit the data. The code also integrates due date handling, expanding the application's capabilities for efficient task location and deadline management.

Project Purpose:

Efficiently managing tasks is crucial for success in both professional and academic pursuits. EAT THE FROG meets this requirement by providing a user-friendly platform that streamlines task organization through a proven strategy for managing to-do lists. With its intuitive interface, prioritization features, task completion tracking, search functionality, and due-date management, the application serves as an invaluable resource for users aiming to elevate their productivity with a comprehensive tool.

Sample Dialogue:

User Adds a New Task:

User enters name of task and description of task.
User selects priority from the dropdown menu.
User clicks the "+" button.
Application adds the task to the appropriate order in the task list.

User Removes a Task:

User selects a task in the list.
User clicks the "-" button.
Application promptly removes the selected task.

User Completes a Task:

User selects a task in the list.
User clicks the "Complete" button.
Application marks the task as completed and transfers it to the completed tasks list.

User Checks Task Description:

User double-clicks on a task in the list.
Application displays a dialog box with the task description.

User Searches Task

User enters task title or description into search bar.
Application returns tasks or description that includes keywords entered.

Main Algorithms and Data Structures:

Add Task Algorithm:

Accepts user input for task title, description, and priority.
Creates a new task object and adds it to the task list.

Priority Algorithm:

Allows users to set the priority of tasks from 1 to 5.
Orders the task list based on priority.

Remove Task Algorithm:

Removes the selected task from the task list.

Complete Task Algorithm:

Marks the selected task as completed.
Moves the completed task to the completed tasks list.

Check Description Algorithm:

Displays the description of the selected task in a dialog box.

Search Algorithm:

Allows user to search for specific task.

Due Date Algorithm:

Alerts users to upcoming due dates and sorts data accordingly.

Exception Handling:

- Error messages for removing or adding a task without selection.
- Confirmations to prevent unintended task removal or completion.
- Informative messages for successful operations.