ABSTRACT

Focus Planner

This project explores the design and implementation of a **To-Do List Application** using multithreading in Java, aimed at enhancing efficiency in task management. The application maintains a list of tasks using an ArrayList, which is shared across multiple threads to perform various operations such as adding, removing, and displaying tasks concurrently. Each of these operations is handled by separate threads, improving performance by executing them in parallel. To ensure the consistency and correctness of the shared data structure, thread synchronization techniques are applied. The synchronized keyword is used to prevent race conditions and ensure that only one thread can access or modify the task list at a time, preserving data integrity.

The project demonstrates the practical use of multithreading in Java, highlighting how multiple processes can work together to achieve efficient and smooth task management. This approach allows the system to handle real-world scenarios where users may simultaneously update, remove, and view tasks without encountering conflicts or inconsistencies. Moreover, the use of synchronization ensures that critical sections of code—those that modify the shared task list—are properly protected, thereby avoiding issues such as data corruption and improper task handling.

This multithreaded To-Do List Application is particularly useful in environments where multiple users or processes need to interact with a shared task list concurrently, making it a suitable solution for collaborative task management systems. The project also emphasizes the importance of multithreading in modern application development, showcasing its capability to improve responsiveness and performance while managing shared resources effectively. The code structure not only adheres to best practices in concurrent programming but also demonstrates how Java's built-in synchronization mechanisms can be leveraged to create robust, thread-safe applications.

Through this project, students gain hands-on experience in managing concurrency, handling shared resources, and preventing data inconsistencies, all critical aspects of multithreaded programming. This implementation provides valuable insights into designing efficient and reliable task management systems for real-world applications, particularly those requiring simultaneous access to shared data.

-Tirumalaraju Deepika Varma(2320030013)

B.Mrinalini(2320030309)

Ganji Subhashini(2320030241)