

1 Task Manager

1.1 Descrizione

L'applicazione è un task manager con un'interfaccia grafica basata su Qt. Le caratteristiche principali sono la possibilità di aggiungere, modificare e cancellare task, nonché di programmare l'esecuzione. Gli utenti possono interagire con l'applicazione sia attraverso la finestra principale che attraverso un'icona di sistema.

L'applicazione è strutturata in un'architettura a tre livelli: livello di presentazione, livello di applicazione e livello di dati. Ogni livello è separato dagli altri e comunica attraverso interfacce che consentono di apportare modifiche all'implementazione di un livello senza influenzare gli altri. All'interno di questa architettura, un livello può interagire direttamente solo con i livelli inferiori.

1.1.1 Task

Un task è un processo che deve essere eseguito, i cui attributi base sono il nome e la descrizione.

L'applicazione supporta due tipi di task: interni ed esterni. Un task interno significa che è stata definita una classe specifica all'interno dell'applicazione stessa, che estende la classe base "Task". Lo scheduler sarà in grado di eseguire il task appropriato in fase di esecuzione, utilizzando l'implementazione del design pattern "Strategy".

Una volta definita la classe, con lo stato e il comportamento corrispondenti, l'applicazione può essere ricostruita e distribuita con la nuova versione. Ciò richiede l'aggiornamento dell'applicazione e lo sviluppo del task esclusivamente con Qt. L'utente può così lanciare l'applicazione e aggiungere un nuovo task con un nome corrispondente alla nuova classe di task creata.

Per adattare l'interfaccia alla quantità variabile di dati da inserire per ogni attività e per poter modificare i valori dei parametri durante l'esecuzione dell'applicazione, i parametri dell'attività sono salvati in un oggetto JSON. In esso, per ogni attributo, vengono salvati il tipo (int, string, double) e il valore associato. Pertanto, un task interno deve accedere ai suoi attributi utilizzando un JSONObject.

Un task esterno, invece, offre maggiore flessibilità e scalabilità ed è progettato per essere utilizzato in un ambiente distribuito. In sostanza, un task esterno rappresenta l'esecuzione di un'istruzione da riga di comando. Ciò consente di definire task che utilizzano tecnologie diverse, che possano essere aggiunti senza dover aggiornare l'applicazione.

1.1.2 Scheduling

L'applicazione offre due tipi di programmazione di base: "Fixed" e "Repeatable". La pianificazione "Fixed" consente agli utenti di selezionare giorni specifici per l'esecuzione dell'applicazione e, per ogni giorno, di specificare gli orari di esecuzione. La pianificazione "Repeatable", invece, definisce un'attività che

inizia a una data e un'ora specifiche e viene ripetuta ogni ora/secondo, a scelta dell'utente.

Per supportare la possibilità di definire diverse schedulazioni, impostabili tramite l'interfaccia, è stato utilizzato il modello di progettazione "Builder". Ciò consente di definire in modo flessibile le schedulazioni utilizzando un "Director" per gestire la costruzione della finestra appropriata e diversi "Builder" a seconda del tipo di finestra necessaria. Nell'applicazione sono stati definiti due "Builder", ciascuno responsabile di un tipo specifico di schedulazione. Questi costruttori eseguono un numero predeterminato di passi per creare l'interfaccia utente necessaria per l'immissione di parametri validi. Per introdurre un nuovo tipo di schedulazione, è sufficiente definire un nuovo costruttore e implementare i passaggi necessari come desiderato.

Una volta completata la definizione dell'attività, questa viene aggiunta alla finestra principale e al menu dell'icona di sistema. Gli utenti possono quindi accedere e modificare i parametri dell'attività. Questa funzionalità è ottenuta attraverso l'implementazione di un "Decorator", che consente di utilizzare le strutture delle finestre esistenti per la creazione dei task, aggiungendo al contempo informazioni specifiche sull'attività.

Per determinare se visualizzare una schermata di creazione o di aggiornamento, la finestra principale comunica con un "Creator" che implementa un Factory Method che restituisce il tipo di oggetto corretto.

1.1.3 Database

Dopo il processo di configurazione, tutte le informazioni vengono salvate in un database esterno. È stata stabilita la connessione con un server MySQL. In un caso ipotetico, questo permette che il database venga impostato su un server diverso da quello dell'applicazione. Se il server dell'applicazione dovesse smettere di funzionare per qualsiasi motivo, sarebbe possibile avviare l'applicazione su un altro server e ripristinare l'intero stato in pochi semplici passaggi; ciò consentirebbe di riprendere le attività pianificate.

1.1.4 Scheduler

Dopo aver stabilito la pianificazione dei compiti, questi vengono memorizzati e consegnate allo Scheduler. Lo Scheduler crea un elenco di task da eseguire nel giorno corrente. Questo elenco viene aggiornato ogni giorno a mezzanotte, in modo che lo Scheduler possa raccogliere i task da eseguire e disporli secondo l'orario corretto.

I task non possono essere eseguiti in parallelo, ma in sequenza. Se un task è in esecuzione e si sovrappone all'orario di inizio del task successivo, quest'ultimo verrà "messo in attesa" e verrà eseguito subito dopo il completamento del task in esecuzione. Una volta completata, lo Scheduler aggiorna la pianificazione del task.

Ogni task ha un array di dimensione 7. Ogni voce dell'array rappresenta la programmazione di quel giorno, a cui è associata una lista di orari che verrà

iterata o aggiornata secondo le necessità.

1.2 Utilizzo dell'applicazione

Quando si apre l'applicazione, viene visualizzata l'interfaccia principale. Facendo clic sul pulsante "Create a new task" si aprirà la schermata di creazione dell'attività. Nella prima finestra è possibile definirne gli attributi. L'applicazione, come accennato in precedenza, supporta due tipi di attività: quelle esterne e quelle interne. Di seguito viene descritto come aggiungere questi due compiti all'applicazione ed eseguirli correttamente:

- Task1, consente di selezionare un file di testo e di creare una copia del suo contenuto all'interno di un altro file .txt. Definire un nome e una descrizione per il task e attivare il flag "External Task". Ora è possibile inserire l'istruzione da eseguire. Il comando è il seguente: `<Path To>\Task1\build\Desktop_Qt6_2_4_MinGW_64_bit-Debug\debug\Task1.exe`. Premere il pulsante "Next >". Nella nuova schermata, selezionare il tipo di pianificazione preferito, fisso o ripetibile, e poi continuare. Una volta raggiunta la schermata di pianificazione, selezionare i dati che si desidera includere e fare clic su "Finish". Una volta eseguito, all'interno della cartella `<Path To>\build\Desktop_Qt6_2_4_MinGW_64_bit-Debug\debug` sarà presente un nuovo file: "task1.txt".
- Task2, consente, invece, di verificare l'esistenza di un file in un percorso specifico. Per creare un nuovo task, fare nuovamente clic sul pulsante corrispondente. Questa volta, assicuratevi di assegnare al task il nome Task2, altrimenti non sarà possibile procedere con la verifica dell'esistenza del file di classe nella directory. Quindi, fare clic sul pulsante "Add Attribute". Nella nuova riga visualizzata, inserire "filePath" come nome dell'attributo e selezionare "string" dal menu a discesa. Infine, inserire il percorso di un file sul computer, ad esempio "C:\example.txt". Una volta eseguito, verrà stampato in debug se il file è stato trovato o meno.

Nella finestra principale è possibile selezionare le attività. È possibile eliminare le attività selezionate facendo clic sul pulsante "Delete selected tasks". Inoltre, è possibile visualizzare e modificare i dettagli di ogni attività, aggiungendo nuovi attributi o cambiando i valori di quelli esistenti.

Facendo clic con il tasto destro del mouse sull'icona di sistema corrispondente si possono eseguire 3 diverse azioni: è possibile richiamare la finestra principale se è stata chiusa; accedere al menu dell'icona di sistema, dove è possibile vedere un elenco di attività, visualizzarne i dettagli, apportare modifiche; ed infine chiudere definitivamente l'applicazione.