

Progetto di Programmazione e modellazione a oggetti

Sessione invernale

Anno accademico 2020/2021

Sviluppatore:

Tommaso Cicco matricola: 282783

Sommario

Sommario	2
Specifica del software:	3
Studio del problema:	4
Funzionamento del timer:	4
Raccolta e visualizzazione delle statistiche:	4
Implementazione degli intervalli di tempo personalizzati:	4
Scelte architetturali:	5
Utilizzo del pattern comportamentale “State”:	5
Utilizzo del pattern creazionale “Singleton”:	5
Diagramma delle classi principali del sistema:	6
Principali use cases con relativo UML:	7

Specifica del software:

Programma con interfaccia grafica che implementa la tecnica di gestione del tempo "Pomodoro".

All'avvio del programma l'utente sceglie il numero di pomodori (intervalli da 25 minuti durante i quali dovrà rimanere concentrato), poi avvia il timer: tra un pomodoro e l'altro l'utente ha diritto a una pausa breve (5 minuti), mentre ogni 4 pomodori ha diritto a una pausa lunga (15 minuti).

Si ha la possibilità di mettere in pausa il timer e resettarlo, inoltre è possibile modificare gli intervalli di tempo a proprio piacimento e visualizzare i minuti di concentrazione totali e la data delle sessioni di lavoro precedenti.

Studio del problema:

Funzionamento del timer:

Il timer interno al programma deve essere in grado di alternare intervalli di tempo di concentrazione (di durata di default di 25 minuti), a intervalli di tempo di relax (di durata di 5 minuti, o di 15 ogni 4 intervalli di concentrazione). Per implementare questo comportamento ho deciso di utilizzare il pattern comportamentale "State" trattando il timer come una macchina a stati. Il timer rimarrà nello stato FocusTimer durante gli intervalli di tempo dedicati alla concentrazione per poi cambiare stato in RelaxTimer durante le pause, e viceversa, fino alla fine della sessione di lavoro; a questo punto l'utente potrà resettare il timer riportandolo allo stato iniziale per iniziare una nuova sessione o terminare l'esecuzione del programma.

Raccolta e visualizzazione delle statistiche:

Il programma deve essere in grado di calcolare i minuti totali di concentrazione per ogni sessione completata; i dati relativi a ogni sessione sono presentati all'utente tramite il modulo "DataGridView" di Windows Forms. Vista la semplicità dei dati da raccogliere ho deciso di optare per un semplice file di testo come mezzo di memorizzazione delle statistiche.

Implementazione degli intervalli di tempo personalizzati:

Per implementare la possibilità di personalizzare la durata degli intervalli di concentrazione e di relax ho deciso di introdurre due nuovi possibili stati per il timer (`CustomRelaxTimer` e `CustomFocusTimer`). Se l'utente sceglie di avviare il timer su un intervallo di tempo personalizzato, sarà modificato lo stato di partenza al fine di modificare il comportamento del timer.

Scelte architetturali:

Utilizzo del pattern comportamentale “State”:

La classe responsabile per il comportamento del timer interno al programma (`Context`) è stata implementata utilizzando il pattern “State”: ciò significa che i metodi messi a disposizione da quest’ultima cambieranno il loro effetto a seconda dello stato attivo (indicato dall’attributo `currentState`) al momento della chiamata.

Gli stati possibili sono 4 e sono implementati attraverso le classi `FocusTimer`, `RelaxTimer`, `CustomFocusTimer`, `CustomRelaxTimer` (ognuna di esse dispone dei metodi dell’interfaccia `ITimerState`).

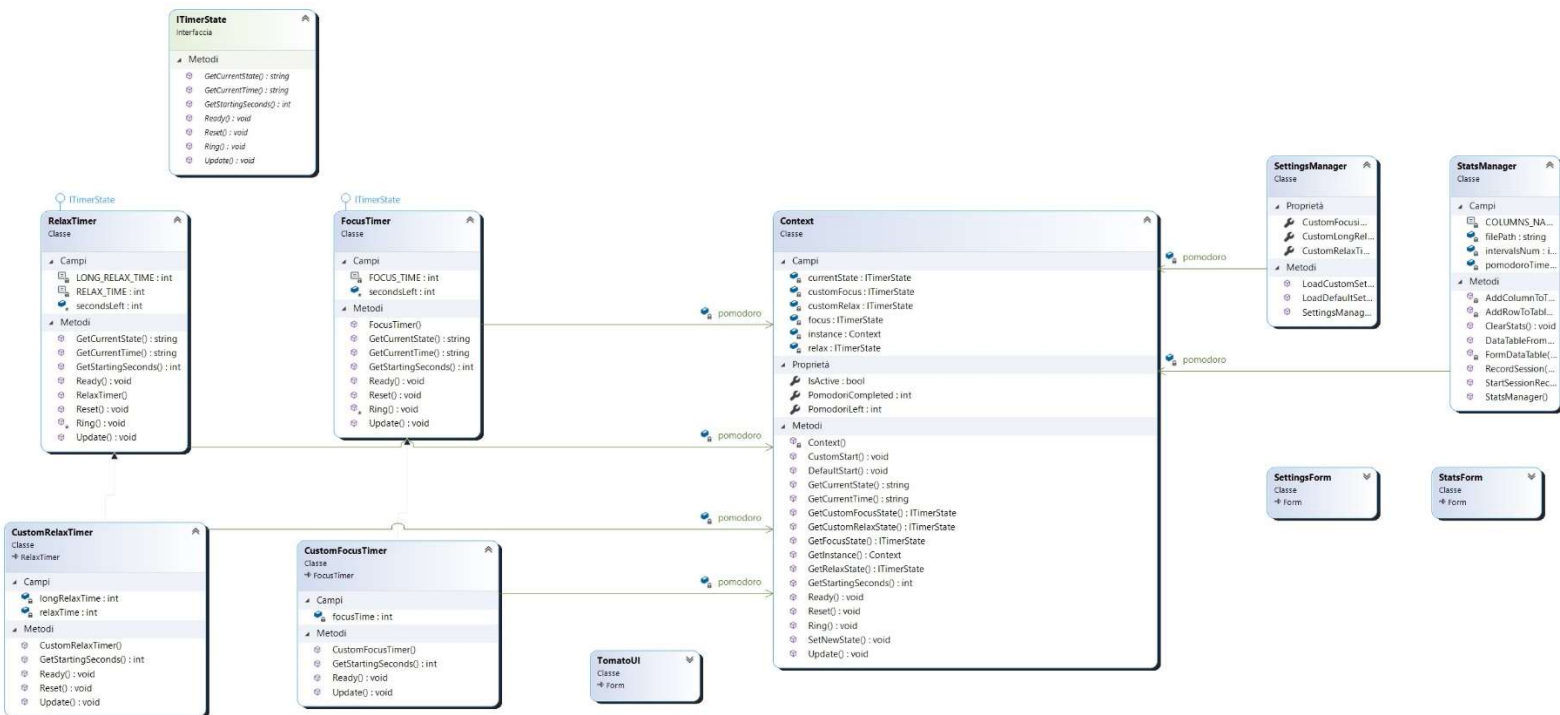
All’avvio di una nuova sessione il timer è impostato su `FocusTimer` se l’utente non ha specificato intervalli di tempo personalizzati, altrimenti sarà impostato su `CustomFocusTimer`.

I cambi di stato avvengono allo scadere dell’intervallo di tempo del timer attivo o richiamando il metodo di `reset` il quale reimposta il timer allo stato di partenza.

Utilizzo del pattern creazionale “Singleton”:

La classe `Context` è stata resa una classe singleton in modo da assicurarsi che ne esista una sola istanza.

Diagramma delle classi principali del sistema:



Principali use cases con relativo UML:

Name: Avvio nuova sessione

ID: UC01

Actors: A1 USER

Preconditions:

- Il timer è fermo.
- E' visualizzato l'intervallo di tempo di concentrazione.

Basic course of events:

- A1 clicca sul tasto "START" per avviare il timer.

Postconditions:

- Il timer è attivo.
- E' Possibile mettere in pausa il timer.
- Non è possibile modificare le impostazioni.
- Non è possibile modificare il numero di intervalli di concentrazione.

Alternative paths: /

Name: Scelta intervalli di tempo

ID: UC02

Actors: A1 USER

Preconditions:

- Il timer è fermo.
- Non è attiva una sessione.

Basic course of events:

- A1 clicca sul tasto “config” per aprire il menu delle impostazioni.
- A1 spunta l’opzione “Custom time (minutes)”.
- A1 sceglie gli intervalli di tempo.
- A1 clicca sul tasto “Apply” per tornare alla finestra principale.

Postconditions:

- Il timer è impostato sull’intervallo di concentrazione specificato dall’utente.

Alternative paths:

- A1 clicca sul tasto “Apply” senza spuntare l’opzione “Custom time (minutes)”;
In questo caso il timer rimarrà impostato sugli intervalli di default.

Name: Visualizzazione delle statistiche

ID: UC03

Actors: A1 USER

Preconditions:

- A1 si trova nella finestra principale del software.

Basic course of events:

- A1 clicca sul tasto “stats” per aprire la finestra delle statistiche.
- Se sono presenti sessioni registrate, A1 visualizza il tempo di concentrazione relativo ad ogni sessione.
- A1 clicca sul tasto “EXIT” per tornare al menu principale.

Postconditions:

- A1 si trova nella finestra principale e può avviare una nuova sessione.

Alternative paths:

- A1 può cancellare le sessioni registrate cliccando sul tasto “CLEAR”.