Corso di Laboratorio di Programmazione

Esercitazione 6
Document
29/11/2021

Implementare la classe `Document` per la gestione di documenti testuali. `Document` immagazzina i documenti come lista di stringhe: una `std::list<std::string>` è quindi l'unico dato membro. La classe `Document` non si limita a immagazzinare questo tipo di dati, ma effettua anche letture da stream ed elaborazioni delle stringhe contenute in essa.

In questa esercitazione dovrete usare un elemento nuovo, non visto a lezione: gli stream da file. Questi sono gestiti tramite tre classi della libreria standard: `ofstream, ifstream, fstream`, come descritto qui:

https://www.cplusplus.com/doc/tutorial/files/

Uno stream da file non è concettualmente diverso dagli stream di standard input/output, a parte la necessità di aprire e chiudere tali stream (quelli standard sono aperti e chiusi automaticamente). Una volta aperto uno stream da file, sarà possibile leggere con l'`operator>>`, nello stesso modo in cui si legge usando l'`operator>>` applicato a `cin`. Sia uno stream da file in lettura che `cin` sono degli `istream` – e questo è il tipo con cui `Document` gestisce gli input da stream.

Prima di implementare `Document`, visionate la documentazione degli stream da file, e capite come aprirli e chiuderli.

Le funzionalità che `Document` deve fornire sono le seguenti:

- `add_line`: aggiunge una nuova stringa, passata come argomento (per const reference, al documento);
- `read_line`: accetta un argomento di class `istream`, da cui legge una nuova stringa e la aggiunge al documento;
- `operator<<`: scrive l'intero documento, con l'indicazione delle diverse stringhe;
- `size`: restituisce il numero di linee del documento;
- `char_count`: restituisce il numero di caratteri del documento;
- `char_count_no_whitespaces`: come la precedente, ma non conta gli spazi;
- `find_replace`: accetta due stringhe: la prima è da cercare, la seconda da sostituire a ciascuna occorrenza della prima.

Oltre a quanto richiesto, devono essere implementate le operazioni essenziali (lezione 9).

Note
Il progetto deve essere correttamente diviso in più file – ricordando le **include guards** per gli header; la compilazione può avvenire usando **CMake**.

Per il test di `read_line` è necessario creare un file di testo. L'apertura di un `ifstream` è in modalità testo di default, perciò non deve essere indicata.

Per lo sviluppo è possibile fare ricorso a qualsiasi elemento di STL – in particolare, prima di cominciare a sviluppare si consiglia di visionare le funzioni membro delle classi `std::list` e `std::string`

https://www.cplusplus.com/reference/list/list/
https://www.cplusplus.com/reference/string/string/

e la funzione std::getline:

https://www.cplusplus.com/reference/string/string/getline/