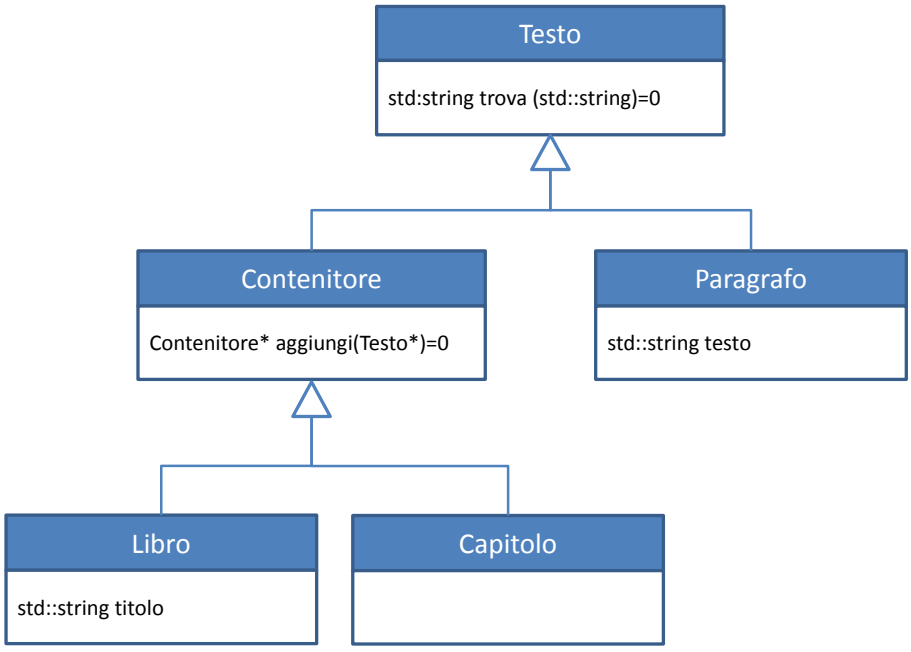
Programmazione di sistema

Anno accademico 2015-2016

Esercitazione 2

Il pattern Composite prevede la possibilità di creare gerarchie di oggetti in grado di rappresentare la relazione parte/tutto, gestendo in modo uniforme sia i diversi tipi di elementi che vengono tra loro composti che il processo di composizione stesso (http://it.wikipedia.org/wiki/Composite).

Utilizzando tale pattern, si modelli il concetto di "contenuto testuale", su cui è possibile eseguire l'operazione elementare di ricerca di un frammento di testo. Poiché il contenuto testuale può essere organizzato gerarchicamente in paragrafi, capitoli e libri, si implementi la seguente gerarchia di classi definendo in maniera polimorfica i metodi trova e aggiungi.



La classe Contenitore mantiene al proprio interno un elenco ordinato e dinamico di puntatori alle parti contenute; il metodo aggiungi(…) provvede ad inserire al fondo di tale elenco l'elemento specificato, diventandone il possessore. Nell'implementazione polimorfica di tale metodo fornita dalla classe Libro, si impedisca l'inserimento di elementi di tipo Libro o di tipo Paragrafo (lanciando eventualmente un'eccezione), mentre nella corrispondente implementazione della classe Capitolo, si impedisca solo l'inserimento di un Libro.

Per forzare l'allocazione degli oggetto di tipo Testo e derivati sullo heap, si definiscano "protected" i costruttori e si aggiungano nelle singole classi concrete opportuni metodi statici di tipo factory (ad esempio, nella classe Libro, static Libro* creaLibro(std::string titolo) che provvedono ad allocare l'oggetto dinamicamente).

Il metodo trova(…) cerca il testo specificato all'interno dell'elemento su cui è stato invocato ed eventualmente dei suoi sotto elementi. In caso di successo, restituisce una rappresentazione del cammino in cui il testo è stato trovato nella forma /<ChildN>/<ChildM>/<from>:<to>, dove

<ChildN> e <ChildM> rappresentano rispettivamente gli indici dei sottoelementi coinvolti (partendo da 1), mentre <from>:<to> rappresentano le posizioni iniziali e finali della stringa cercata all'interno del paragrafo (anche in questo caso, partendo da 1). In caso di fallimento, trova restituisce una stringa vuota (""). Ad esempio, se il Libro l contiene i capitoli c1 e c2 e all'interno di c1 sono presenti i paragrafi "alfa" e "beta", e all'interno di c2 sono presenti "gamma" e "delta", la ricerca di "eta" restituisce: /1/2/2:4, mentre la ricerca di "ma" restituisce /2/1/4:5. In caso di occorrenze multiple del testo cercato, viene restituita la descrizione della prima occorrenza.

Si impedisca la copia e l'assegnazione di oggetti di tipo Testo.

Si verifichi il corretto comportamento della gerarchia implementata e l'opportuno rilascio delle risorse acquisite.

Competenze da acquisire

- Gestione della memoria
- Ereditarietà
- Polimorfismo
- Gestione delle eccezioni
- Type casting