Dipartimento di Informatica Sistemistica e Comunicazione
Corso di Laurea in Informatica

Programmazione 2

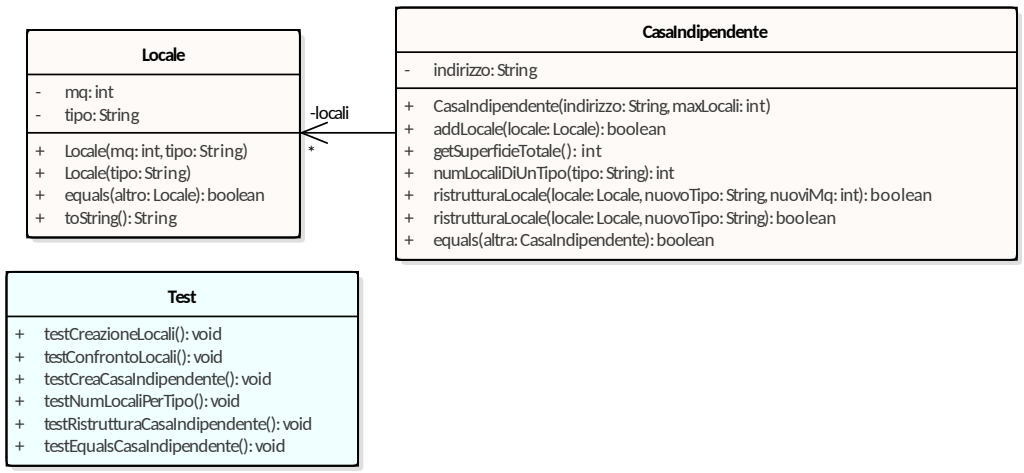
30 Giugno 2017 – Primo Compitino

Testo parte di pratica

Una casa indipendente è costituita da un insieme di locali. I locali sono caratterizzati da una destinazione d'uso (camera, sala, cucina, e così via) e dalla superficie espressa in mq. I locali di una casa indipendente, a seguito di ristrutturazione, possono modificare il mq e la destinazione d'uso di un locale pre-esistente, oppure la sola destinazione d'uso.

Implementare le classi come rappresentate dal seguente diagramma UML. **I diagrammi delle classi Locale e CasaIndipendente NON includono gli eventuali metodi di incapsulamento che DEVONO essere individuati correttamente e codificati.**

Infine, la classe `Test` (già fornita) contiene un insieme di test che devono essere fatti girare di volta in volta in modo da verificare la corretta realizzazione del software. **Come requisito minimo per ottenere una valutazione positiva, lo studente deve garantire che la sua implementazione non presenti errori di compilazione e superi almeno 3 casi di test fra quelli dati.**



Classe `Locale`:

- ✓ rappresenta un locale. È caratterizzato da una destinazione d'uso (`tipo`) e una superficie (`mq`)
- ✓ tutti gli attributi sono mutabili ed accessibili in lettura dall'esterno
- ✓ definisce un costruttore che inizializza gli attributi. Se la superficie è minore di 1, viene impostata a 5, se la destinazione d'uso e' non specificata o nulla, viene impostata a "camera".
- ✓ due locali sono uguali se hanno la stessa superficie e la stessa destinazione d'uso
- ✓ il metodo `toString` restituisce una stringa con l'informazione relativa al locale (superficie e destinazione d'uso)

Classe `CasaIndipendente`:

- ✓ rappresenta una casa indipendente costituita da un insieme di locali (associazione `locali`) e caratterizzata da un indirizzo (attributo `indirizzo`). I locali non sono accessibili in lettura dall'esterno e l'indirizzo è immutabile, ma accessibile in lettura dall'esterno
- ✓ definisce un costruttore che inizializza la casa indipendente specificando il suo indirizzo e di quanti locali è costituita. Si assume che i parametri passati in ingresso `indirizzo` e `maxLocali` siano rispettivamente non nullo e maggiore di zero
- ✓ il metodo `addLocale` permette di aggiungere il locale alla prima posizione libera dell'array. L'aggiunta va a buon fine se il locale passato in ingresso è non nullo e se c'è ancora posto per inserirlo. Il metodo ritorna `true` se l'aggiunta va a buon fine, o `false` in caso contrario.

- ✓ il metodo `getSuperficieTotale` restituisce la superficie totale della casa indipendente sommando le superfici dei locali attualmente costituenti la casa
- ✓ il metodo `numLocaliDiUnTipo(tipo)` restituisce il numero di locali della casa indipendente che abbiano la destinazione d'uso del `tipo` specificato in ingresso
- ✓ il metodo `ristrutturaLocale(locale, nuovoTipo, nuoviMq)` permette di modificare la superficie e la destinazione d'uso del `locale` che corrisponde a quello specificato in ingresso. Se un tale `locale` esiste fra quelli della casa, la modifica viene eseguita e il metodo restituisce `true`. Se il locale non esiste, allora deve essere restituito `false`.
- ✓ Il metodo `ristrutturaLocale(locale, nuovoTipo)` è un overload del precedente. Funziona allo stesso modo, ma mantiene invariati i metri quadri originari del `locale`, modificandone solo la destinazione d'uso
- ✓ due case indipendenti sono uguali se hanno lo stesso indirizzo e hanno la stessa superficie