

# Programmazione 2

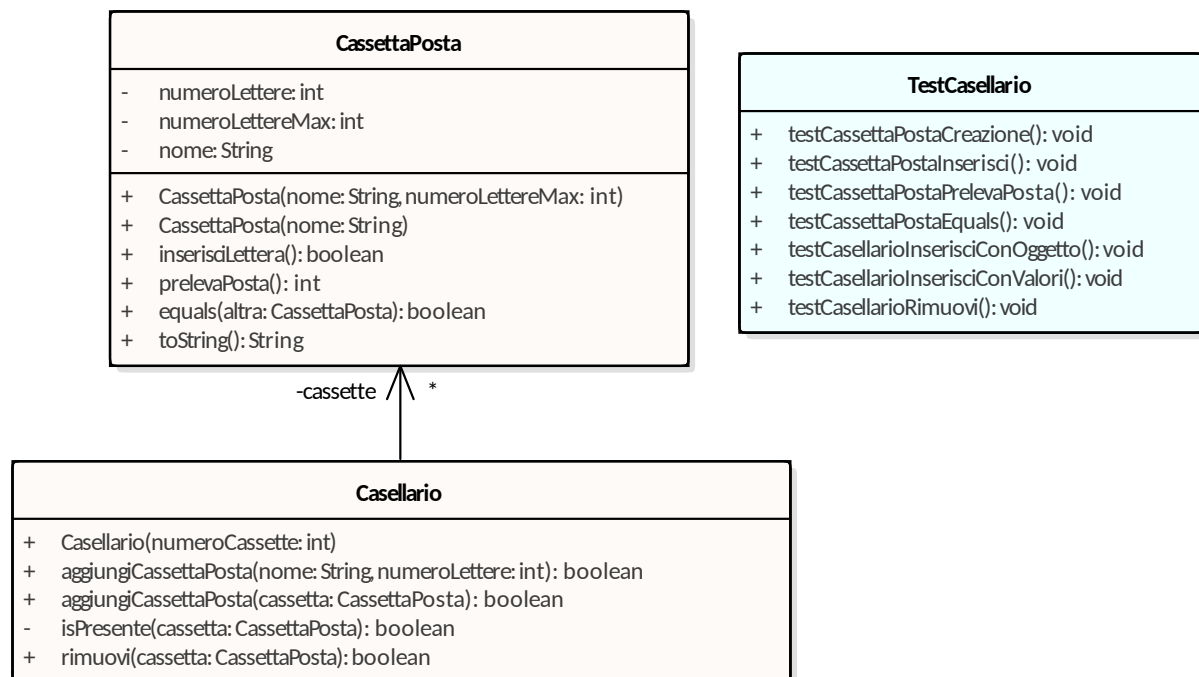
18 Aprile 2018 – Primo Compitino

Testo parte di pratica

Un casellario installato in un condominio contiene un insieme di cassette della posta. Una cassetta viene creata ed inserita nel casellario se c'è ancora spazio e se non esiste già un'altra cassetta della posta uguale (cioè di proprietà della stessa persona). Quando la persona a cui è intestata una cassetta della posta cambia residenza, allora la cassetta viene rimossa dal casellario. Ogni cassetta della posta ha una propria capienza massima in termini di numero massimo di lettere che può contenere. Questo vuol dire che all'interno di un casellario ci possono essere cassette con dimensioni differenti.

Implementare le classi come rappresentate dal seguente diagramma UML. **I diagrammi delle classi contengono tutti e i soli metodi richiesti, compresi quelli di incapsulamento.**

Infine, la classe `TestCasellario` (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 CassettaPosta:

- ✓ rappresenta una cassetta per la posta cartacea. È caratterizzata da un nome del proprietario (`nome`), una capienza massima di lettere che può contenere (`numeroLettereMax`) e un numero di lettere attualmente inserite nella cassetta (`numeroLettere`)
- ✓ definisce un costruttore che inizializza gli attributi. Assumete che i valori passati in ingresso siano validi (stringa diversa da `null` e numero lettere max maggiore di 0)
- ✓ definisce un ulteriore costruttore che inizializza il nome con il valore passato in ingresso ed imposta la capienza massima a 5
- ✓ il metodo `inserisciLettera` restituisce `true` se l'inserimento di una lettera va a buon fine (cioè se c'è ancora spazio) e `false` in caso contrario
- ✓ il metodo `prelevaPosta` restituisce il numero complessivo di lettere all'interno della cassetta rimuovendole
- ✓ due cassette della posta sono uguali se hanno lo stesso nome

- ✓ il metodo `toString` restituisce una stringa con l'informazione relativa alla cassetta della posta (nome, numero di lettere attualmente inserite e numero di lettere max)

#### Classe Casellario:

- ✓ rappresenta un casellario in grado di contenere cassette della posta (associazione `cassette`). Il numero di cassette totali possibili è specificato in fase di costruzione
- ✓ definisce un costruttore che inizializza il casellario
- ✓ il metodo privato `isPresente(cassetta)` restituisce `true` se `cassetta` è già presente nel casellario (cioè se esiste nel casellario una cassetta uguale a quella passata in ingresso). `False` in tutti gli altri casi
- ✓ il metodo `aggiungiCassettaPosta(cassetta)` aggiunge nella prima posizione libera del casellario la cassetta passata in ingresso se diversa da `null` e se non già presente nel casellario. Restituisce `true` in caso di inserimento, `false` in caso contrario
- ✓ il metodo `aggiungiCassettaPosta(nome, numeroLettere)` effettua l'overloading dell'omonimo metodo accettando però in ingresso il nome e il numero di lettere massimo della nuova cassetta che si vuole aggiungere. Il metodo segue le stesse regole dell'omonimo `aggiungiCassettaPosta(cassetta)`
- ✓ il metodo `rimuovi(cassetta)` rimuove la cassetta all'interno del casellario uguale a `cassetta` se esiste e se `cassetta` è diversa da `null`. Il metodo restituisce `true` se rimuove la cassetta, `false` se `cassetta` era `null` oppure non era presente