

**Non è consentito usare libri o appunti.**

**Esercizio 1. [15 punti]** Implementare un sistema per gestire i contratti di locazione di immobili. Un contratto è caratterizzato dal *nome del locatore*, *nome del conduttore*, e *indirizzo immobile*. I contratti possono essere: ad *UsoAbitativo* oppure ad *UsoCommerciale*.

1. Un contratto ad uso abitativo è caratterizzato da *numero di vani*, dal *piano* e dal *numero di persone* residenti. Lanciare un'eccezione controllata se il numero di persone è superiore a quattro volte il numero di vani.
2. Un contratto ad uso commerciale è caratterizzato da numero di metri quadri e tipologia di attività che può essere ad alto impatto ambientale o basso impatto ambientale.

Prevedere i metodi:

- a) `calcolaAffitto` che determina il costo di locazione tenendo conto che: ogni vano ha un costo di euro 130, un metro quadro costa euro 15.
- b) `calcolaTasse` che determina il costo annuale delle tasse da pagare tenendo conto che: per gli immobili uso abitativo per ogni vano si pagano 100€ di TARI, per ogni persona residente 35€ di TASI, per ogni vano 10€ di condominio. Per gli immobili ad uso commerciale ogni metro quadro si pagano 32€ a cui va sommata una quota base di € 45 per attività a basso impatto e € 75 per attività ad alto impatto.

**Esercizio 2. [10 punti]** Definire la classe `AgenziaImmobiliare` che ha i seguenti attributi: *nome*, la *lista degli immobili*, *numero massimo di immobili gestibili*. Corredare la classe con i metodi:

1. `aggiungiImmobile` che aggiunge un immobile alla lista. Lanciare una eccezione non controllata `NumMaxException` se la lista già contiene il numero massimo di immobili gestibili.
2. `affitta` che loca un immobile presente nella lista ad un conduttore con una famiglia di *n* persone. Lanciare una eccezione non controllata `GiaLocatoException` se l'immobile è già locato.
3. `dammiImmobiliVani(int x)` che dato un numero di vani *x* restituisce la lista di immobili non ancora locati che hanno *x* vani.

**Esercizio 3. [5 punti]** Implementare una classe starter che:

Crea un oggetto `AgenziaImmobiliare` con 5 immobili scegliendo 3 ad uso abitativo e 2 ad uso commerciale (possibilmente generare i valori degli immobili in modo random).

Provare tutti i metodi dell'Esercizio 2:

- `aggiungiImmobile`
- `affitta`

**Ogni violazione delle regole enunciate ai punti sotto elencati comporta l'annullamento della prova (l'elaborato viene valutato 0).**

1. Durante la prova d'esame è vietato usare:

- a. libri e appunti sia in forma cartacea che in forma digitale
  - b. supporti di memoria esterni
  - c. un font di dimensione maggiore di 10 punti.
2. Non è consentito modificare i file allegati alla traccia.
3. Il nome del progetto consegnato deve cominciare con COGNOME seguito dal carattere underscore e quindi dal NOME (tutto in maiuscole). Ad esempio, il nome del progetto di Marco Rossi può essere ROSSI\_MARCO, ROSSI\_MARCO\_P2, ROSSI\_MARCO\_ESERCIZIO, ROSSI\_MARCO\_549449384, etc.
4. Il file da consegnare deve essere creato da eclipse seguendo i passi:
  - a. Seleziona “export...” nel menu file
  - b. Seleziona “Archive File” in “General”
  - c. Pressa “Next”
  - d. Seleziona progetto da esportare
  - e. Controllare il percorso del file (nell’area di testo con etichetta “To archive file:”)
  - f. Assicurarsi che i pulsanti radio nel pannello Options siano selezionati su “Save in zip format” e “Create directory structure for files”
  - g. Pressa “Finish”

Assicurarsi che i progetti consegnati possono essere importati in eclipse come:  
 General → Existing Projects into Workspace