

Progetto Gestione Condominio

IDE utilizzato : NetBeans 6.8, **Database utilizzato:** MySQL 5.1.36

Il progetto consiste di:

- una classe principale Gestione Condominio, contenente il main e dei metodi di tipo static. Nel main verranno inizializzati gli oggetti utilizzati poi nel programma. Questa inizializzazione avviene mediante metodi di classe che effettuano una interrogazione del db e successivamente restituiscono strutture dati apposite (Vector) popolate con gli oggetti appropriati. Successivamente gli oggetti verranno manipolati in base alle scelte dell'utente, utilizzando istruzioni racchiuse nei metodi static di questa classe.
- Cinque classi che definiscono altrettanti oggetti: Appartamento, Proprietario, Pagamento, Spesa, SpesaSpeciale. In ognuna di esse sono definite opportune variabili di classe e opportuni metodi di istanza e di classe.
- Una classe di supporto Controlli, contenente alcuni metodi static di controllo e supporto.

Il programma, una volta lanciato, presenta all'utente diverse opzioni per la gestione del condominio mediante un'interfaccia a riga di comando, ad ognuna delle quali corrispondono diverse operazioni sugli oggetti. Una volta effettuate le operazioni, prima di uscire, il programma salva le modifiche agli oggetti scrivendo le opportune informazioni sul database. Tali informazioni verranno richiamate alla successiva esecuzione, quando gli oggetti verranno di nuovo inizializzati.

SCHEMA CLASSI (verranno omessi i getter e i setter per questioni di spazio; i metodi static verranno indicati con (S))

GestioneCondominio

METODO	DESCRIZIONE
main	Inizializzazione oggetti e gestione interazione con l'utente
(S) cambioProprietario	Cambio del proprietario di un appartamento
(S) nuovaSpesa	Inserimento di una nuova spesa
(S) nuovaSpesaSpeciale	Inserimento di una nuova spesa speciale
(S) nuovoPagamento	Inserimento di un nuovo pagamento
(S) visualizzaBilancio	Visualizzazione del bilancio (somma Pagamenti - somma Spese)
(S) visualizzaSpeseApp	Visualizza le spese di un appartamento

Appartamento

METODO	DESCRIZIONE
addToSpeseSostenute	Aggiunta della spesa all'elenco di spese sostenute
elencSpese	Fornisce l'elenco spese dell'appartamento
cambiaProprietario	Cambio del proprietario dell'appartamento
vectorSpeseToString	Restituisce una stringa con l'elenco spese
(S) getSupTotale	Calcola la somma delle superfici di tutti gli appartamenti

(S) getInfoAppartamenti	Visualizza le informazioni sugli appartamenti
(S) readDB	Lettura info da DB e inizializzazione oggetti
(S) updateDB	Scrittura DB

Proprietario

METODO	DESCRIZIONE
supPossedutaTotale	Superficie totale posseduta
calcolaQuotaSup	Quota percentuale di superficie posseduta
vectorAppIntToString	Restituisce una stringa con l'elenco appartamenti intestati
elencaDebitori	Elenca proprietari in debito (saldo negativo)
(S) readDB	Lettura info da DB e inizializzazione oggetti
(S) updateDB	Scrittura DB

Pagamento

METODO	DESCRIZIONE
effettuaPagamento	Gestione pagamento di un proprietario
(S) readDB	Lettura info da DB e inizializzazione oggetti
(S) updateDB	Scrittura DB

Spesa

METODO	DESCRIZIONE
dividiEaddebita	Gestione importo spesa in base alla superficie posseduta da ciascun proprietario
addToAppartamenti	aggiunge il codice di una spesa ad ogni appartamento del condominio
spesaPerProprietario	importo addebitato a ciascun proprietario per questa spesa
(S) readDB	Lettura info da DB e inizializzazione oggetti
(S) updateDB	Scrittura DB

SpesaSpeciale

METODO	DESCRIZIONE
addToAppartamenti	Aggiunta codice spesa al relativo appartamento
inserisciSpesaSpeciale	Addebita l'importo sul saldo del relativo proprietario
(S) readDB	Lettura info da DB e inizializzazione oggetti
(S) updateDB	Scrittura DB

Questa classe, essendo sottoclasse della classe Spesa, ne eredita i metodi. Per il metodo addToAppartamenti è stato effettuato l'override.

Controlli

METODO	DESCRIZIONE
(S) getDBConnection	Connessione al database
(S) arrotonda	Gestione arrotondamento double
(S) checkDate	Controllo date
(S) formatDate	Gestione formato date per db MySQL

