ESAME DI FONDAMENTI DI INFORMATICA T-2 del 15/06/2021
Proff. E. Denti – R. Calegari – A. Molesini
Tempo a disposizione: 3 ore

NOME PROGETTO ECLIPSE: CognomeNome-matricola (es. RossiMario-0000123456)
NOME CARTELLA PROGETTO: CognomeNome-matricola (es. RossiMario-0000123456)
NOME ZIP DA CONSEGNARE: CognomeNome-matricola.zip (es. RossiMario-0000123456.zip)
NOME JAR DA CONSEGNARE: CognomeNome-matricola.jar (es. RossiMario-0000123456.jar)

Si devono consegnare DUE FILE: *l'intero progetto Eclipse* e *il JAR eseguibile*

Si ricorda che compiti *non compilabili* o *palesemente lontani da 18/30* NON SARANNO CORRETTI e causeranno la verbalizzazione del giudizio "RESPINTO"

L'amministratore del complesso residenziale *The Dent* ha richiesto lo sviluppo di un'app per calcolare il costo pro-quota del riscaldamento condominiale a gas, per ogni appartamento, su base sia mensile che annuale.

DESCRIZIONE DEL DOMINIO DEL PROBLEMA

Ogni appartamento è dotato di un *partitore di calore* che mensilmente registra il consumo in KWh dell'appartamento: tali registrazioni vengono poi fornite all'amministratore, su base *annuale*, attraverso un apposito file.

L'amministratore ha inoltre a sua disposizione, in un ulteriore file, i dati di ciascun appartamento, ovvero: *codice identificativo*, *nome del proprietario* (o inquilino nel caso l'appartamento sia concesso in locazione) e *consumo massimo contrattuale* di gas, espresso però in *standard m^3* di gas metano.

Infine, dalla bolletta trasmessa dal fornitore l'amministratore è in grado di estrarre i seguenti dati:

- Importo totale: importo totale della bolletta
- Costi fissi: costi imputabili alla gestione dell'utenza
- Costi variabili: costi che dipendono dal consumo di gas
- Consumo totale: totale di m^3 di gas riportati in bolletta
- Costo al m^3: costo di un singolo m^3 di gas
- Costo extra al m^3: costo di un singolo m^3 di gas oltre una certa soglia (maggiore del precedente)

Il calcolo della quota di costo di ciascun appartamento deve avvenire secondo il seguente algoritmo:

1. si recupera il consumo dello specifico mese registrato dal partitore di calore dell'appartamento
2. si recupera il consumo massimo contrattuale di m^3 dello specifico appartamento
3. si verifica preliminarmente se il consumo effettivo registrato dal partitore superi o meno il massimo contrattuale di tale appartamento: in tal caso, la *quota eccedente* dovrà essere tariffata usando il *costo extra* al m^3 (più elevato del costo standard) così da disincentivare i consumi eccessivi; altrimenti, tutto il gas sarà tariffato a tariffa standard.
4. si aggiungono i costi fissi, che vanno suddivisi in parti uguali tra i condomini
5. infine, se la somma di tutte le quote così calcolate differisce dal totale della bolletta (sia in eccesso che in difetto), si applica a tutte le quote una *correzione uniforme*: a tal fine si calcola dapprima la differenza tra il costo totale della bolletta e la somma delle quote, poi si suddivide tale differenza in parti uguali tra tutti i condomini.

Nel caso di quota annuale l'algoritmo viene ripetuto per ciascun mese dell'anno, in modo da fornire un calcolo quanto più preciso e accurato possibile.

TEMPO TOTALE STIMATO PER SVOLGERE L'INTERO COMPITO: 2h30

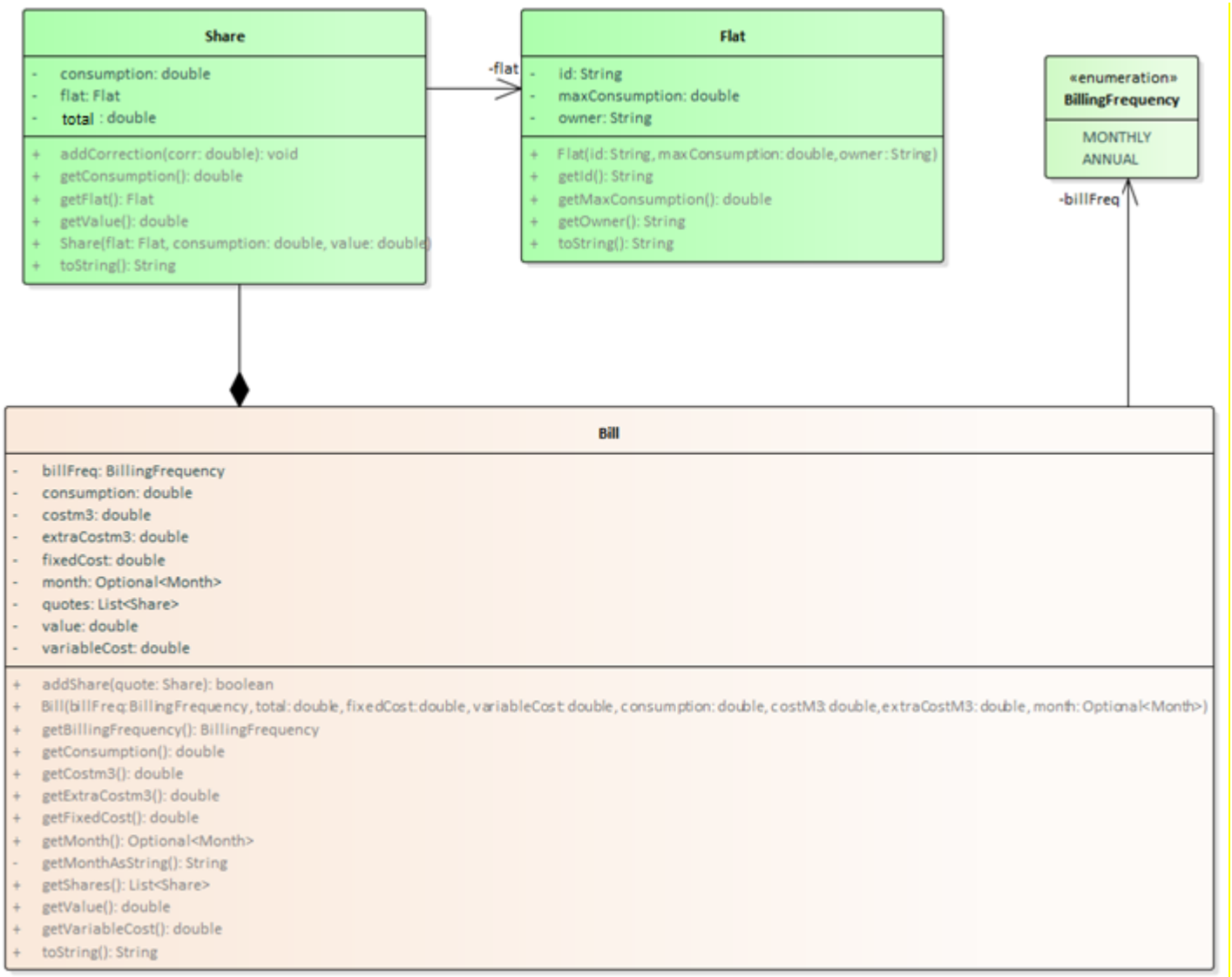
Cose da ricordare

- salva costantemente il tuo lavoro: l'informatica a volte può essere "subdolamente ostile"…
- in particolare: se ora compila e stai per fare modifiche, salva la versione attuale (non si sa mai)

Parte 1 (punti: 14)

Dati (namespace gasforlife.model) [TEMPO STIMATO: 30 MINUTI] *(punti: 6)*

Il modello dei dati deve essere organizzato secondo il diagramma UML di seguito riportato:



SEMANTICA:

a) L'enumerativo ***BillingFrequency*** (fornito) definisce i due valori MONTHLY e ANNUAL, rispettivamente per bollette mensili o annuali.

b) La classe ***Flat*** (fornita) modella un singolo appartamento e fornisce *l'identificativo* associato all'appartamento (composto da un codice alfanumerico che indentifica il civico, il piano e l'appartamento), il *nome del proprietario* (o dell'inquilino nel caso l'appartamento sia in locazione) ed il *consumo massimo contrattuale mensile* in termini di m^3 di gas; appositi accessor consentono di recuperare i valori dei parametri memorizzati dalla classe. Completa la classe un'ovvia implementazione di ***toString***.

c) La classe ***Share*** (fornita) modella *ciascuna delle quote* della bolletta associata ad *uno specifico* appartamento. Ogni quota memorizza il consumo e l'importo associati a un dato appartamento; appositi accessor consentono di recuperare i valori dei parametri memorizzati dalla classe, mentre il metodo ***addCorrection*** permette di aggiornare il valore della quota applicando il "fattore correttivo" specificato. Completa la classe un'ovvia implementazione di ***toString***.

d) La classe **Bill** (da realizzare) rappresenta la Bolletta e incapsula i seguenti dati:

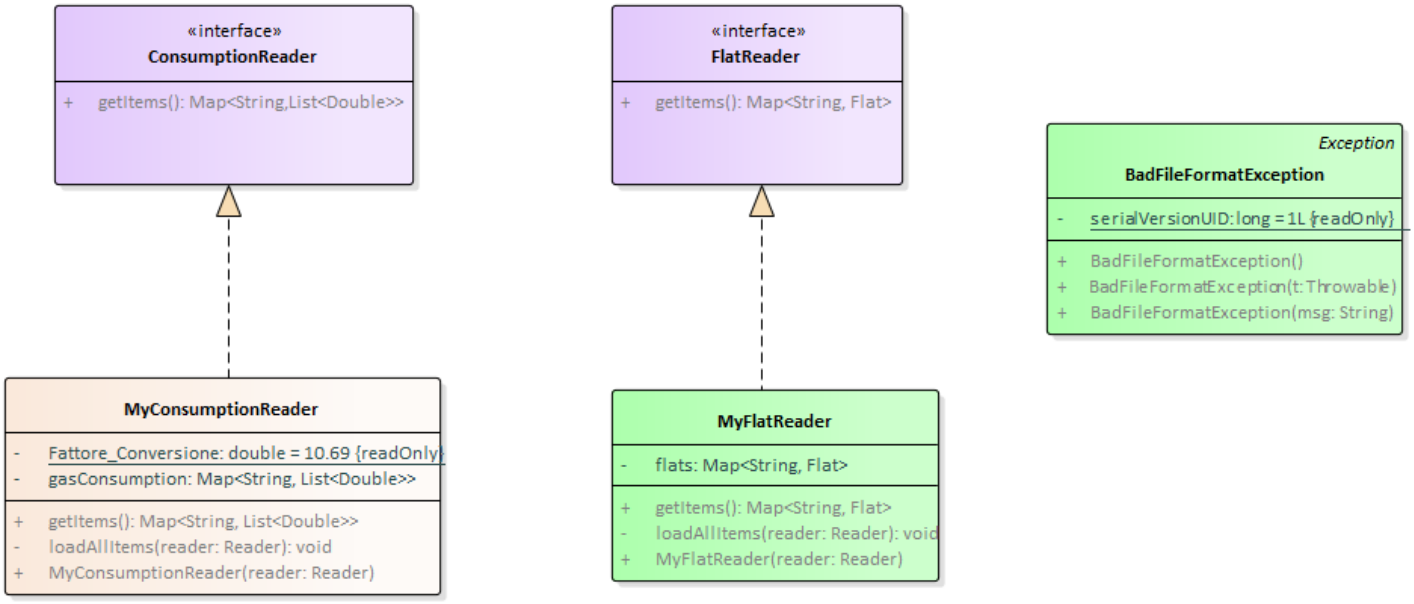
- Importo totale: importo totale della bolletta
- Costi fissi: costi imputabili alla gestione dell'utenza
- Costi variabili: costi che dipendono dal consumo di gas
- Consumo: numero totale di m^3 indicati nella bolletta
- Costo al m^3: costo di un singolo m^3 di gas
- Costo extra al m^3: costo di un singolo m^3 di gas extra quota (maggiore del precedente)
- Mese di riferimento: solo nel caso la bolletta sia mensile (***opzionale***)
- Quote: la lista delle quote (***Share***)

Requisiti:

- Il costruttore riceve tutti gli elementi sopra elencati **tranne l'insieme delle quote**, che verranno calcolate in un secondo momento. Deve controllare che gli argomenti ricevuti non siano null, zero, o numeri negativi, lanciando in tal caso apposita ***IllegalArgumentException*** con idoneo messaggio.
- Il metodo ***addShare*** aggiunge al ***Bill*** una quota (***Share***): restituisce l'esito dell'operazione
- Il metodo ***getMonthAsString*** deve fornire il nome del mese *secondo la cultura locale italiana*, o la stringa "mese non presente" se esso non è specificato. SUGGERIMENTO: DateTimeFormatter.ofPattern("MMMM")
- Il metodo ***toString*** deve fornire, *uno per riga,* non solo tutti i valori memorizzati nella classe (compreso il mese, nel caso di bolletta mensile) ma anche *tutte le singole quote, ordinate per codice di appartamento* (sempre una per riga).

Persistenza (namespace gasforlife.persistence) **[TEMPO STIMATO: 40 MINUTI]** *(punti: 8)*

Questo package definisce il reader per leggere da file gli appartamenti e i consumi.



SEMANTICA:

a) l'interfaccia ***FlatReader*** (fornita) dichiara il metodo ***getItems*** che restituisce una Mappa <String, Flat> con chiave il codice identificativo di ciascun appartamento.

b) La classe ***MyFlatReader*** (fornita) implementa ***FlatReader***: per ipotesi riceve un ***Reader*** nel costruttore, che provvede a leggere il file che memorizza i dati relativi ai vari appartamenti.

c) l'interfaccia ***ConsumptionReader*** (fornita) dichiara anch'essa un suo metodo ***getItems*** che restituisce una Mappa <String, List<Double>> in cui la chiave è il codice dell'appartamento, e il valore associato è la lista delle dodici letture mensili del consumo di quell'appartamento.

d) la classe **MyConsumptionReader** (da realizzare) implementa ***ConsumptionReader***: per ipotesi il costruttore riceve un ***Reader*** già aperto e si occupa della lettura del file memorizzando i dati in un'opportuna Mappa. In caso di problemi di I/O dev'essere lasciata uscire ***IOException***, mentre in caso di problemi nel formato delle righe si deve lanciare ***BadFileFormatException*** (fornita) con preciso e specifico messaggio d'errore.

FORMATO DEL FILE: ogni riga del file memorizza le dodici letture di un dato appartamento (una per ogni mese).

In ogni riga, il primo elemento è una stringa che rappresenta il *codice identificativo* dell'appartamento, seguita da uno spazio, un ":" e un ulteriore spazio. A seguire vengono le *dodici letture* dei consumi, separate tra loro da una barra (|). **È cruciale tenere presente che i valori memorizzati nel file sono espressi in Kwh,** mentre la rendicontazione in bolletta è espressa in **m^3** di gas: pertanto, prima della memorizzazione nella mappa occorre effettuare la *conversione tra le due unità di* misura applicando il fattore di conversione:

1 standard m^3 di gas metano = 10.69 Kwh

Esempio di file:

```
1-1A : 1300|1069|1069|780|780|0|0|0|0|780|1400|1400
...
7-1B : 1400|1069|1069|780|780|0|0|0|0|780|1400|1400
7-2A : 930|940|900|680|680|0|0|0|0|680|860|8600
...
```

Parte 2 (punti: 16)

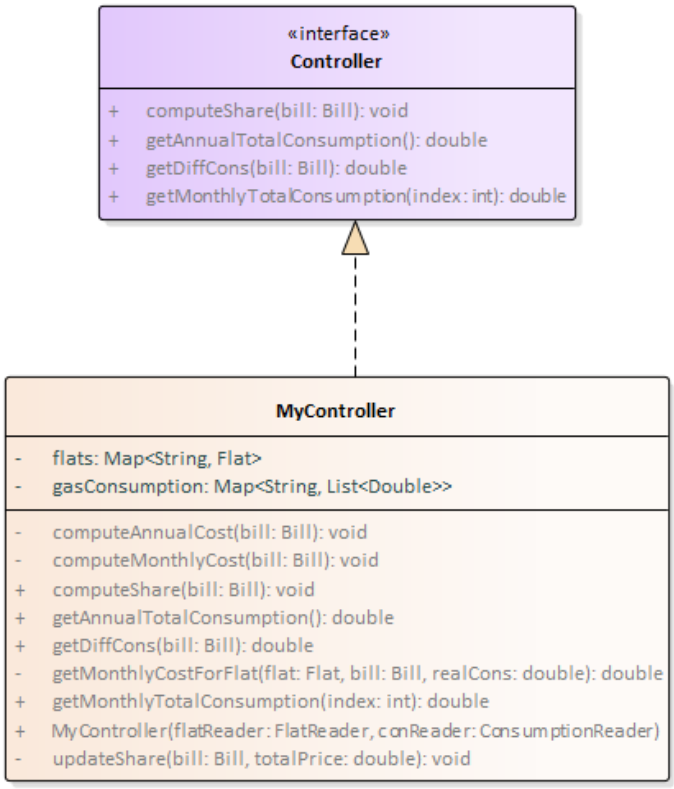
Controller (namespace gasforlife.controller) [TEMPO STIMATO: 60 MINUTI] *(punti: 12)*

Questo package contiene il controller dell'app.

SEMANTICA:

a) l'interfaccia ***Controller*** (fornita) dichiara i metodi:

- ***computeShare***: riceve in ingresso un ***Bill*** e calcola le quote di ogni appartamento memorizzandole all'interno del ***Bill*** ricevuto, secondo l'algoritmo presentato nell'analisi del dominio.
- ***getAnnualTotalConsumption***: calcola il consumo totale *annuale* di gas per tutto il complesso residenziale.
- ***getMonthlyTotalConsumption***: calcola il consumo totale di gas per tutto il complesso residenziale *per lo specifico mese* ricevuto come argomento.
- ***getDiffCons***: calcola la differenza tra il consumo registrato nella bolletta in ingresso e il consumo effettivo che proviene dalla lettura dei partitori. ATTENZIONE: la differenza è calcolata sulla stessa base del ***Bill*** (quindi, è annuale se ***Bill*** è annuale, è mensile se ***Bill*** è mensile).



b) la classe **MyController** (fornita parzialmente realizzata, ma **da completare**) implementa tale interfaccia, aggiungendo svariati metodi privati di ausilio (come ***updateShare***). In particolare devono essere realizzati:

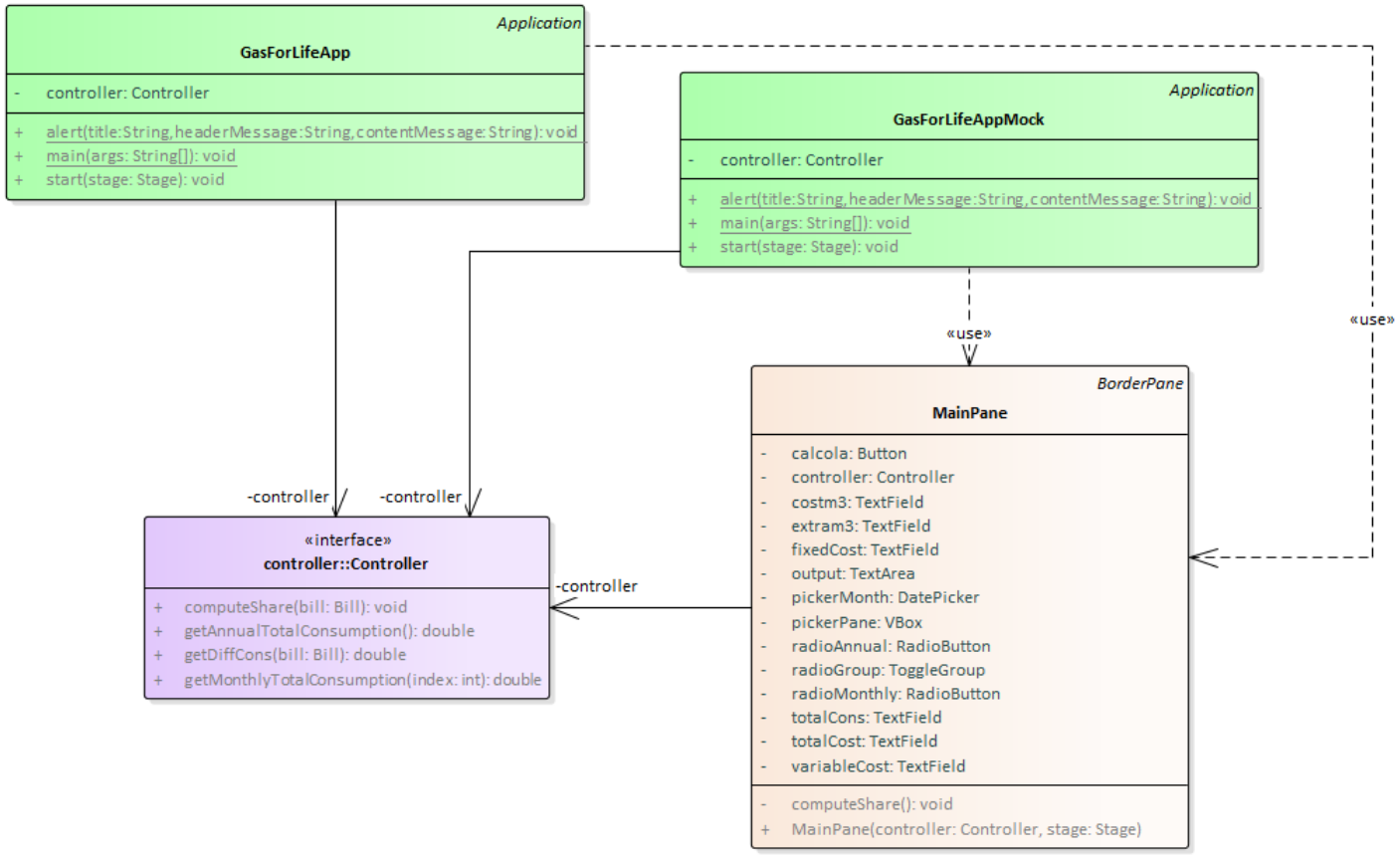
- il metodo privato ***computeMonthlyCost***, che incapsula l'algoritmo di calcolo descritto del Dominio del Problema (mentre l'analogo metodo ***computeAnnualCost*** è fornito già pronto);

SUGGERIMENTO 1: può essere utile usare il metodo fornito ***updateShare***
SUGGERIMENTO 2: può essere utile appoggiarsi a un proprio metodo ausiliario ***getMonthlyCostForFlat*** in cui incapsulare la logica di calcolo del costo mensile per appartamento

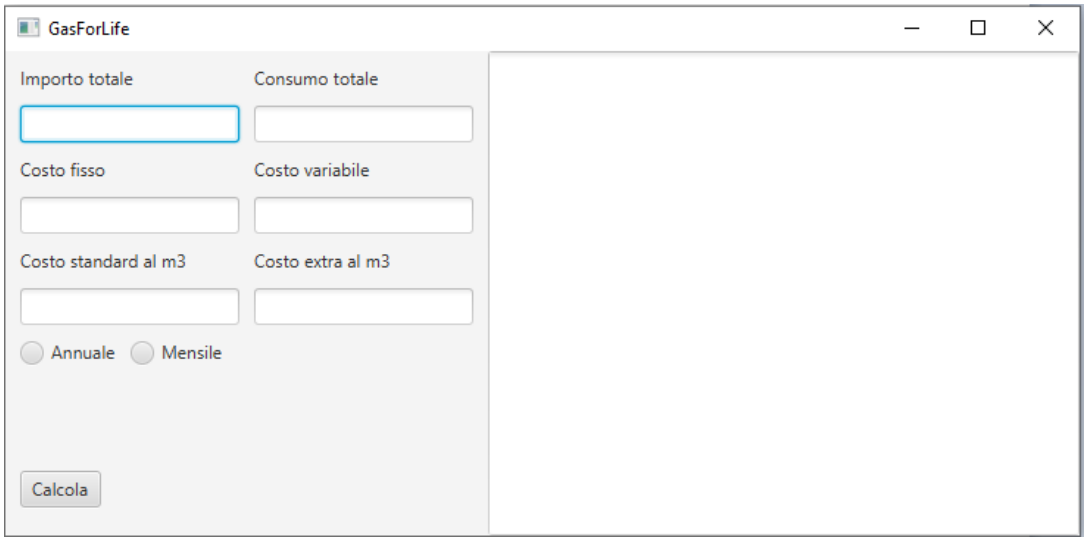
- il metodo pubblico ***getDiffCons***, destinato a essere invocato dalla GUI nella gestione dell'evento di calcolo, appoggiandosi sui metodi ***getAnnualTotalConsumption*** e ***getMonthlyTotalConsumption*** già presenti.

GUI (namespace gasforlife.ui) **[TEMPO STIMATO: 20 MINUTI]** *(punti: 4)*

Questo package contiene le classi che rappresentano l'interfaccia grafica. ***GasForLifeApp*** (fornita) crea la finestra grafica, le classi per la persistenza e il controller. ***GasForLifeAppMock*** (fornita) funge da mock in caso non sia stata svolta la parte relativa alla persistenza. ***MainPane*** (fornita parzialmente realizzata, ma **da completare**) è un ***BorderPane*** che contiene i widget grafici e permette la gestione dell'evento quando viene premuto il tasto "Calcola"



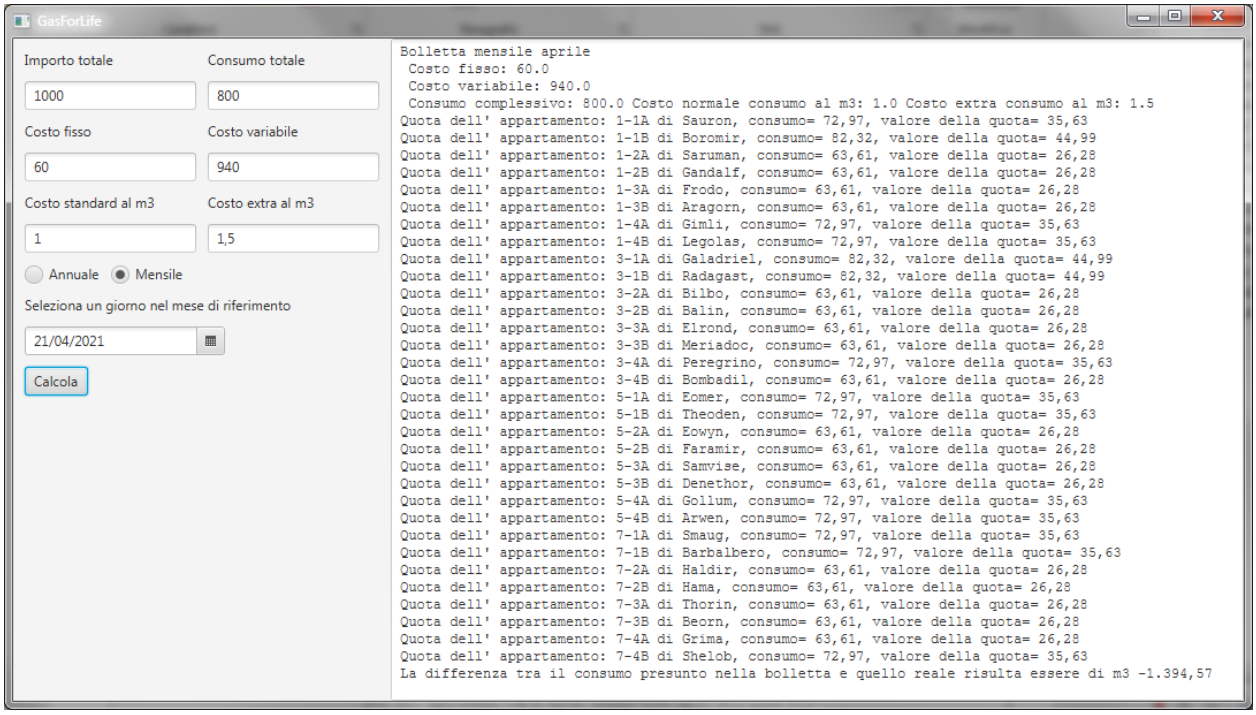
La GUI dev'essere simile (non necessariamente identica) a quella sotto illustrata:



Si tratta di un ***BorderPane***, in cui nel lato sinistro sono presenti *sei campi di testo*, sormontati da *apposite label*, che permettono di inserire i dati ricavati dalla Bolletta da ripartire. *Due radio button* (sotto) permettono di specificare se la Bolletta sia *mensile* o *annuale*: quando viene selezionato "mensile" deve comparire un ***DatePicker*** per selezionare *un giorno all'interno del mese* per il quale si vogliono calcolare le quote. Alla pressione del tasto "Calcola" vengono quindi calcolate le quote, mostrate nella ***TextArea*** presente sulla parte destra (v. figura sottostante).

Il metodo di gestione dell'evento, ***computeShare*** (fornito parzialmente realizzato ma **da completare**), dapprima recupera i e valida i parametri immessi nei campi di testo (parte fornita già pronta), poi agisce come segue:

- recupera dal ***RadioButton*** la frequenza di calcolo (mensile o annuale)
- istanzia il corrispondente ***Bill***
- *delega al controller il calcolo delle quote* ed emette sulla ***TextArea*** il risultato (completo di tutte le informazioni relative alla bolletta, incluse le quote di ciascun appartamento)
- *recupera dal controller l'eventuale differenza di consumi* ed emette sulla ***TextArea*** la frase finale relativa all'eventuale *differenza tra consumo stimato in bolletta e consumo effettivo, opportunamente formattata* con due cifre decimali.



NB: il metodo statico ausiliario ***alert*** consente di mostrare una finestra di dialogo utile a segnalare errori all'utente.

Cose da ricordare

- salva costantemente il tuo lavoro: l'informatica a volte può essere "subdolamente ostile"..
- in particolare: se ora compila e stai per fare modifiche, salva la versione attuale (non si sa mai)

Checklist di consegna

- Hai fatto un **JAR eseguibile**, che contenga cioè l'indicazione del main?
- Hai controllato che **si compili** e **ci sia tutto**? [NB: non includere il PDF del testo]
- Hai **rinominato** IL PROGETTO, lo ZIP e il JAR esattamente come richiesto?
- Hai **chiamato** la cartella del progetto esattamente come richiesto?
- Hai fatto un **unico file ZIP (NON .7z, rar o altri formati)** contenente l'intero progetto? In particolare, ti sei assicurato di aver incluso tutti i file .java (e non solo i .class)?
- **Hai consegnato DUE file distinti, ossia lo ZIP col progetto e il JAR eseguibile?**
- Su EOL, hai **premuto** il tasto "CONFERMA" per inviare il tuo elaborato?