ESAME DI FONDAMENTI DI INFORMATICA T-2 dell' 8/1/2014
Proff. E. Denti – G. Zannoni
Tempo a disposizione: 4 ore MAX

NB: il candidato troverà nell'archivio ZIP scaricato da Esamix anche il software "Start Kit"

NOME PROGETTO ECLIPSE: CognomeNome-matricola (es. RossiMario-0000123456)

Per le prossime elezioni, il Consiglio della Gallia Capocciona deve stabilire se sia meglio un sistema elettorale proporzionale, un sistema maggioritario o un mix dei due: a tal fine è stata richiesta un'applicazione che simuli i risultati nei diversi casi.

DESCRIZIONE DEL DOMINIO DEL PROBLEMA

Nel sistema proporzionale puro, il numero di seggi da assegnare a ciascun partito è determinato in proporzione ai voti da esso complessivamente ottenuti sul territorio (questa parte non è dettagliata perché il relativo componente software è fornito già pronto).

Nel sistema maggioritario puro, invece, il territorio è diviso in tanti piccoli collegi elettorali, ognuno corrispondente a un solo seggio, che viene attribuito al partito che prende più voti ("the winner takes all"): i voti degli altri partiti in quel collegio vanno persi.

Nei sistemi misti maggioritario-proporzionali si applicano separatamente le due regole di calcolo alla quota-seggi da assegnare col maggioritario e alla quota-seggi da assegnare col proporzionale, sommando poi i due risultati.

È opportuno notare che nel sistema proporzionale i seggi da assegnare possono essere in numero qualsiasi, mentre nel sistema maggioritario uninominale sono necessariamente tanti quanti i collegi, dato che ogni collegio elegge esattamente un deputato. Di conseguenza, nei sistemi misti il numero di seggi da assegnare col maggioritario è bloccato (pari al numero di collegi), mentre quello che varia è il numero di seggi da assegnare col proporzionale.

ESEMPIO: nel caso della Gallia Capocciona, il territorio è diviso in 32 collegi: quindi, nel caso del metodo maggioritario anche i seggi da assegnare devono essere esattamente 32 (non uno di più, non uno di meno), mentre adottando il metodo misto potranno essere anche più di 32 (ma non meno), nel qual caso i primi 32 seggi saranno assegnati col maggioritario e gli altri col proporzionale (NB: il metodo proporzionale permetterebbe in teoria di assegnare anche meno di 32 seggi, ma si desidera che anch'esso attribuisca comunque almeno 32 seggi per permettere confronti omogenei).

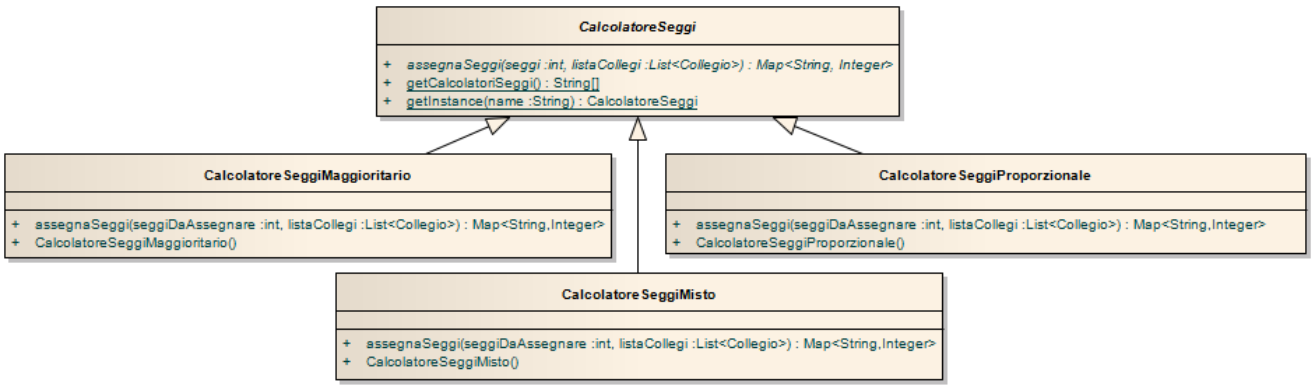
Il file di testo RisultatiGallia.txt contiene i voti riportati da due partiti nei 32 collegi uninominali, un collegio per riga.

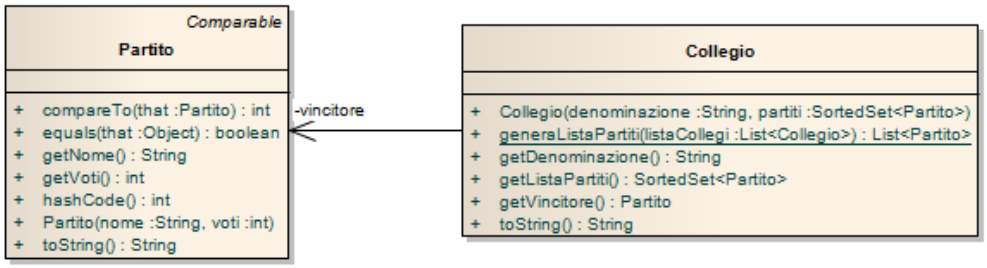
Per legge, nella Gallia Capocciona un partito che decide di presentarsi alle elezioni deve essere presente in tutti i collegi: non possono quindi esserci partiti presenti solo in qualche collegio ma non negli altri.

Parte 1 (punti: 20)

Dati (package galliaelections.model) *(punti: 12)*

Il modello dei dati deve essere organizzato secondo il diagramma UML più sotto riportato.





SEMANTICA:

a) la classe ***Partito*** (fornita nello start kit) rappresenta un partito coi suoi voti (coppia nome/voti) ed è ***Comparable*** rispetto al numero di voti (più voti = più "grande");

b) la classe ***Collegio*** (fornita) rappresenta un collegio elettorale con i corrispondenti partiti; i partiti sono contenuti in un set ordinato (***Partito*** è ***Comparable***) cosicché i partiti con più voti vengono prima degli altri; tale classe contiene un metodo statico ***generaListaPartiti*** che, data una lista di ***Collegio***, restituisce una lista di ***Partito***, ognuno con il totale dei voti ottenuti;

c) la classe astratta ***CalcolatoreSeggi*** (fornita) rappresenta il generico calcolatore seggi indipendente dallo specifico algoritmo di assegnazione; il metodo astratto ***assegnaSeggi*** prende in ingresso il numero di seggi da assegnare e una lista di ***Collegio***, assegna i seggi richiesti secondo un dato algoritmo e restituisce una mappa (nome partito, seggi) adeguatamente configurata. Questa classe contiene inoltre il metodo statico ***getInstance*** che istanzia e restituisce una specifica istanza di ***CalcolatoreSeggi*** in base al nome dell'algoritmo passato come argomento: i nomi leciti sono recuperabili tramite il metodo statico ***getCalcolatoriSeggi***.

d) la classe ***CalcolatoreSeggiProporzionale*** (fornita) concretizza ***CalcolatoreSeggi*** nel caso del metodo proporzionale;

e) la classe ***CalcolatoreSeggiMaggioritario*** concretizza ***CalcolatoreSeggi*** nel caso del metodo maggioritario puro; perciò, la sua implementazione di ***assegnaSeggi*** deve lanciare ***IllegalArgumentException*** con idoneo messaggio d'errore se il numero di seggi da assegnare è *diverso* dal numero dei collegi. [SUGGERIMENTO: per ottenere la mappa da restituire, costruire per prima cosa l'elenco dei nomi dei partiti (facile, perché un collegio contiene tutti i partiti con i relativi nomi e voti) e, mediante questo, inizializzare la mappa inserendo inizialmente zero seggi per tutti i partiti e aggiungendo poi 1 al numero dei seggi per il partito vincitore in ogni collegio.]

f) la classe ***CalcolatoreSeggiMisto*** concretizza ***CalcolatoreSeggi*** nel caso del metodo misto maggioritario-proporzionale; perciò, la sua implementazione di ***assegnaSeggi*** deve lanciare ***IllegalArgumentException*** con idoneo messaggio d'errore se il numero di seggi da assegnare è *minore* del numero dei collegi. Per il calcolo dei seggi assegnati con questo metodo, si utilizzino in modo opportuno ***CalcolatoreSeggiMaggioritario*** e ***CalcolatoreSeggiProporzionale***, ovviamente usando il secondo solo per assegnare i seggi non assegnati dal primo.

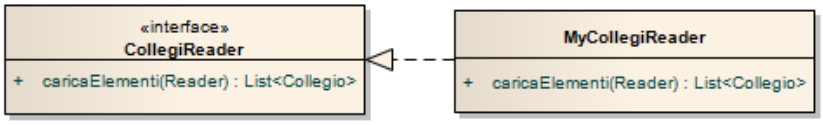
Lo Start Kit contiene anche i test (da includere nel progetto) per verificare il funzionamento di queste classi.

Persistenza (package galliaelections.persistence) *(punti 8)*

Come già anticipato, il file di testo RisultatiGallia.txt contiene i voti riportati da due partiti nei collegi uninominali, un collegio per riga. In particolare la prima riga riporta i nomi dei partiti, separati da tabulazioni (ATTENZIONE: possono esserci anche tabulazioni all'inizio, per incolonnare correttamente i nomi!), mentre le righe successive riportano ciascuna il numero del collegio e i voti ottenuti dai vari partiti, separati anch'essi da tabulazioni.

IMPORTANTE: anche se il file, per semplicità, prevede due soli partiti, l'implementazione deve essere in grado di funzionare con un qualunque numero di partiti, non solo nel caso che siano due.

L'architettura software è illustrata nel diagramma UML che segue:



SEMANTICA:

a) l'interfaccia ***CollegiReader*** (fornita) dichiara il metodo ***caricaElementi***;

b) la classe ***MyCollegiReader*** **(da realizzare)** concretizza ***CollegiReader*** prevedendo un metodo ***caricaElementi*** che legga tutto il contenuto del ***Reader*** ricevuto come parametro e restituisca la lista dei partiti con i relativi dati, lanciando, oltre alla "naturale" ***IOException***, una ***BadFileFormatException*** nel caso di errore nel formato del file. In particolare occorre verificare che:

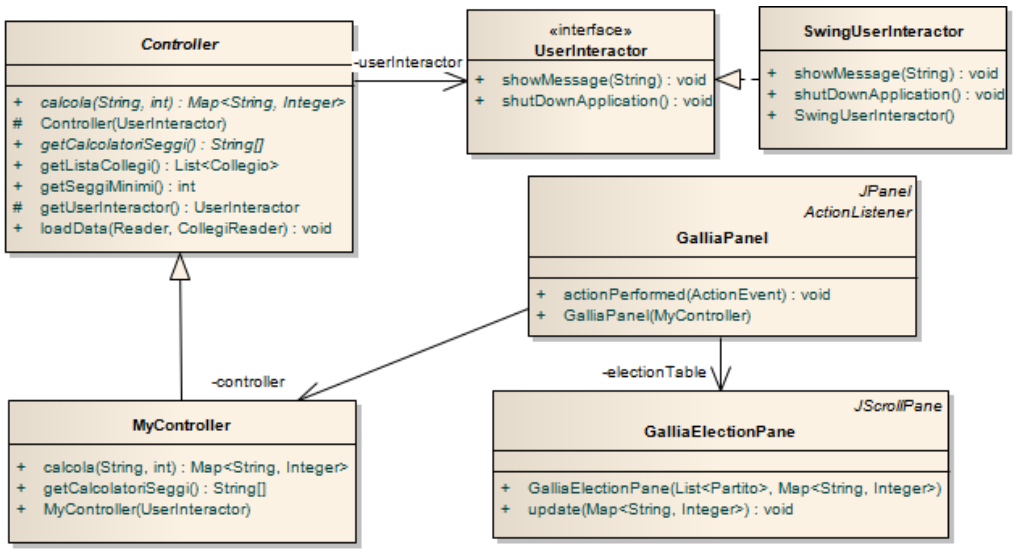
- nella prima riga ci siano almeno due partiti;
- il numero di campi numerici nelle righe successive alla prima sia uguale al numero di partiti;
- i campi numerici siano effettivamente numerici.

Lo Start Kit contiene anche i test (da includere nel progetto) per verificare il funzionamento di questa classe.

Parte 2 (punti: 10)

L'interfaccia grafica deve mostrare una tabella coi voti e i seggi ottenuti dai vari partiti (simile, ma non necessariamente identica, all'esempio mostrato in figura), inizialmente senza seggi assegnati (Fig. 1): sotto di essa, una casella combinata deve permettere di scegliere il metodo di calcolo e i seggi da assegnare (per default, tanti quanti i collegi). Quando l'utente effettua la scelta e preme il bottone, l'applicazione deve aggiornare la tabella mostrando i seggi assegnati corrispondenti (Figg. 2, 3, 4). Se il numero di seggi da assegnare è inferiore al numero di collegi (o, nel caso del maggioritario, anche superiore), l'applicazione deve segnalarlo tramite dialoghi (Figg. 5, 6).

La classe ***Program*** (fornita nello start kit) contiene il *main* di partenza dell'intera applicazione.



Controller (package galliaelections.ui) *(punti 4)*

L'interfaccia ***UserInteractor*** (fornita), implementata dalla classe *SwingUserInteractor* (fornita), consente di mostrare un messaggio all'utente (metodo ***showMessage***) o di terminare l'applicazione (metodo ***shutDownApplication***).

La classe astratta ***Controller*** (fornita) presenta un costruttore con un argomento di tipo ***UserInteractor*** e definisce/dichiara i seguenti metodi, ognuno riportato con la sua semantica:

- ***loadData*** (definito) prende in ingresso un Reader e un CollegiReader ed effettua il caricamento dei dati;
- ***getListaCollegi*** (definito) restituisce la lista delle istanze di ***Collegio*** **caricate da** ***loadData*** **e** gestite dal controller;
- ***getUserInteractor*** (definito) restituisce un oggetto che implementa UserInteractor che consente di mostrare messaggi all'utente;
- ***getCalcolatoriSeggi*** (astratto) restituisce un array di stringhe, contenente i nomi dei ***CalcolatoreSeggi*** disponibili;

- ***calcola*** (astratto) prende come argomento il nome dell'algoritmo di calcolo da impiegare (uno dei nomi restituiti dal metodo ***getCalcolatoriSeggi***) e il numero di seggi da assegnare ed opera sulla lista dei partiti gestita dal controller, riassegnando i seggi sulla base dell'algoritmo selezionato; il risultato è una mappa che mette in relazione il nome di un partito (una stringa) con il numero di seggi calcolati.

La classe ***MyController*** **(da realizzare)** estende la classe ***Controller*** implementando i metodi mancanti ***getCalcolatoriSeggi*** e ***calcola***: viene costruita con gli stessi argomenti di ***Controller*** richiamando il costruttore della classe base. Il metodo ***getCalcolatoriSeggi*** deve essere implementato utilizzando l'omonimo metodo statico di ***CalcolatoreSeggi***, mentre i metodo ***calcola*** deve:

- verificare che il numero di seggi ricevuti come argomento non sia inferiore al numero minimo di seggi assegnabili (metodo ***getSeggiMinimi*** della classe base); in caso contrario, l'esecuzione del metodo deve terminare mostrando all'utente un messaggio adeguato (mediante lo ***UserInteractor***) e restituendo *null*;
- verificare che il nome dell'algoritmo di calcolo dei seggi ricevuto come parametro sia valido (tentare di ottenere l'algoritmo mediante il metodo statico ***getInstance*** di ***CalcolatoreSeggi*** e trattare correttamente l'eventuale eccezione); in caso contrario, l'esecuzione del metodo deve terminare mostrando all'utente un messaggio adeguato (mediante lo ***UserInteractor***) e restituendo *null*.
- effettuate le verifiche di cui sopra, passare al metodo di calcolo del ***CalcolatoreSeggi*** il numero di seggi da assegnare e la lista dei collegi (metodo ***getListaCollegi*** della classe base), restituendo direttamente il risultato ottenuto; se si verificano eccezioni, mostrare il messaggio mediante lo ***UserInteractor*** e restituire *null*.

Lo Start Kit contiene anche i test (da includere nel progetto) per verificare il funzionamento di questa classe.

Interfaccia utente (package galliaelections.ui) *(punti 6)*

La classe ***GalliaElectionPane*** (fornita) incapsula una tabella a tre colonne specializzata per i risultati elettorali: il suo costruttore accetta una lista di ***Partito*** (coi seggi ancora tutti zero) e una mappa (nomi partiti, seggi), usate per popolare inizialmente la tabella; successivamente, i dati visualizzati possono essere modificati invocando il metodo ***update***, a cui va fornita una mappa (nomi partiti, seggi) aggiornata.

La classe ***GalliaPanel*** **(da realizzare)** rappresenta il pannello centrale dell'applicazione (deriva da ***JPanel***): la casella a discesa mostra gli algoritmi disponibili da caricare mediante l'uso del controller. Il campo che contiene il numero di seggi da assegnare deve essere inizialmente popolato con il numero minimo di seggi (ottenuto dal controller).

La pressione del bottone fa scattare l'attribuzione dei seggi, mostrata nella tabella sottostante; prima di procedere al calcolo, occorre verificare che il numero di seggi sia effettivamente un numero (in caso contrario occorre notificare l'utente reimpostando il valore di default) per poi tentare l'assegnazione dei seggi (mediante l'uso del controller). Tenere presente che, in caso di errore, il metodo ***calcola*** del controller notifica autonomamente il problema all'utente (Figg. 5 e 6) e restituisce *null*, quindi ***GalliaPanel*** deve solo reimpostare il numero di seggi di default.

(FIGURE ALLA PAGINA SEGUENTE)

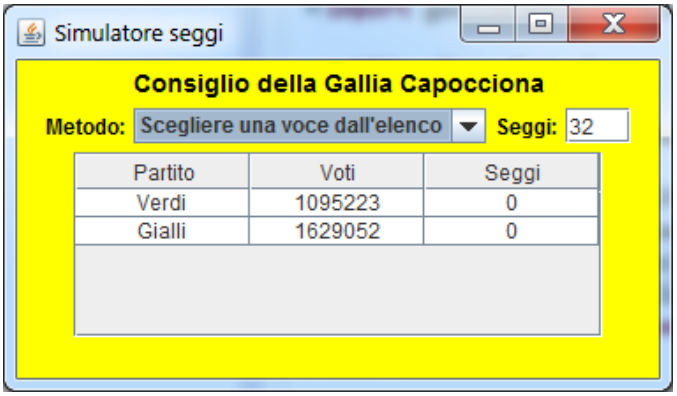


Figura 1

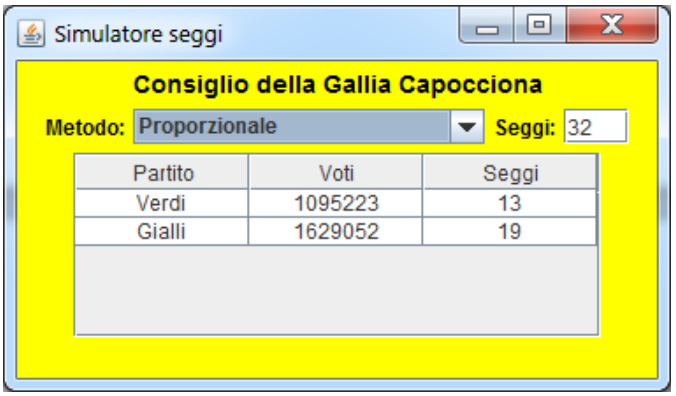


Figura 2

Figura 3

Figura 4

Figura 5

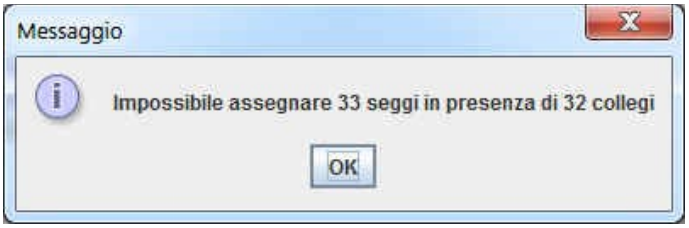


Figura 6