

ESAME DI FONDAMENTI DI INFORMATICA T-2 del 20/07/2017

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 e CARTELLA: CognomeNome-matricola (es. RossiMario-0000123456)

NOME ZIP DA CONSEGNARE : CognomeNome-matricola.zip (es. RossiMario-0000123456.zip)

L'agenzia di sondaggi ZannoPoll ha richiesto un'applicazione che agevoli il calcolo delle percentuali rilevate nei sondaggi che le vengono commissionati sui più diversi argomenti, visualizzandone graficamente i risultati.

DESCRIZIONE DEL DOMINIO DEL PROBLEMA

Nei **sondaggi di opinione** si somministra alla popolazione campione una serie di domande, registrandone ed elaborandone le risposte sulla base di parametri come l'*età* e il *sex* di chi risponde.

I risultati sono tipicamente espressi sotto forma di **percentuali sul totale delle risposte**, filtrando volta per volta (ossia, escludendo dal calcolo) determinate categorie di persone del campione (es. giovani sotto i 25 anni, ultrasessantenni, uomini/donne, etc.) per analizzare meglio la distribuzione delle risposte per sotto-popolazioni o fasce d'età.

ESEMPIO

Si supponga di aver somministrato un sondaggio elettorale sulle prossime elezioni e di aver ottenuto dal campione, inteso, rispettivamente 910, 720, 488, 410, 232, 100 scelte per i partiti A,B,C,D,E,F (totale intervistati: 2860).

Le percentuali corrispondenti, considerando tutti gli intervistati, risultano perciò A=31.82%, B=25.17%, C=17.06%, D=14.34%, E=8.11%, F=3.5%.

Se però si limita l'analisi ai soli intervistati che soddisfano certe caratteristiche di età e sesso (ad esempio *uomini ultracinquantenni*, o *giovani donne sotto i trent'anni*), il risultato può essere molto diverso.

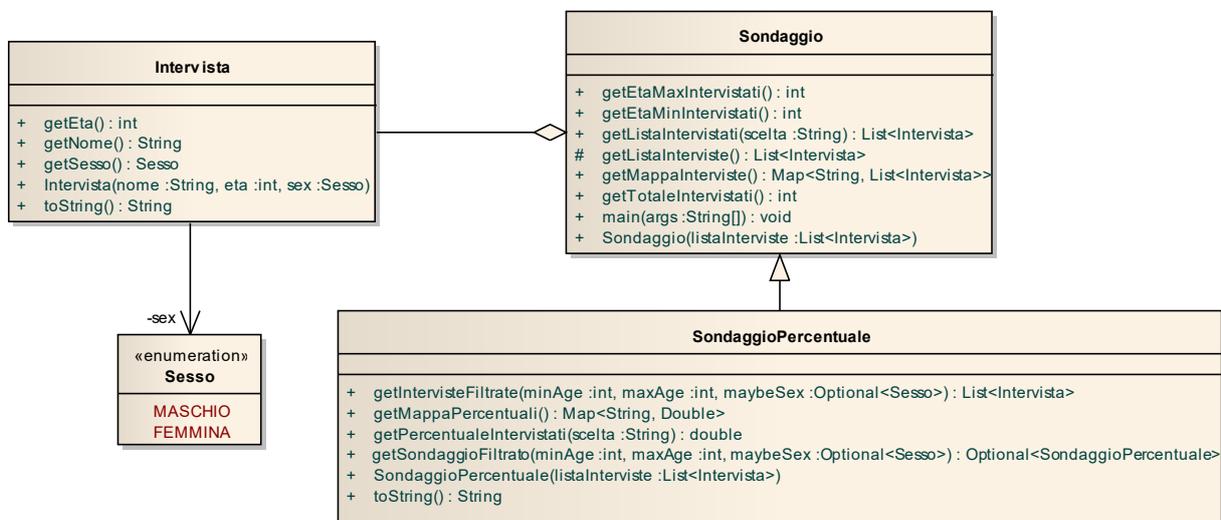
Il file di testo [RisultatiSondaggio.txt](#) contiene i risultati di un sondaggio (nel formato dettagliato più oltre): ogni riga riporta la risposta di un singolo intervistato, con indicazione della sua età e sesso.

Parte 1

(punti: 17)

Dati (package zannopoll.model)

(punti: 10)



SEMANTICA:

a) L'enumerativo **Sesso** (fornito) definisce le costanti per i due sessi;

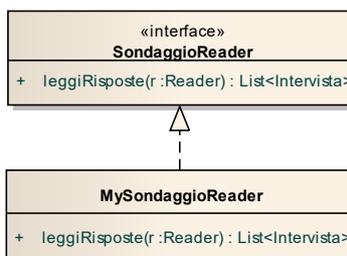
- b) La classe **Intervista** (fornita) rappresenta il risultato di una intervista, caratterizzato dalla scelta (una stringa), età e sesso dell'intervistato/a.
- c) la classe **Sondaggio** (fornita) incapsula le interviste in una mappa interna: il costruttore riceve una lista di interviste, che usa per popolare una mappa avente per chiave i nomi delle possibili scelte; tale mappa è ottenibile dal metodo `getMappaInterviste`. Il metodo `getListeIntervistati(String scelta)` restituisce invece la sotto-lista contenente le sole interviste relative alla scelta specificata, o **IllegalArgumentException** se non ve ne sono. I due metodi `getEtaMaxIntervistati` e `getEtaMinIntervistati` restituiscono l'età max/min degli intervistati.
- d) la classe **SondaggioPercentuale** (da realizzare) specializza **Sondaggio** calcolando e mantenendo internamente (in una mappa <String,Double>) le percentuali ottenute da ciascuna scelta (in questo caso, percentuale è un numero reale compreso fra 0 e 1). Il costruttore riceve la stessa lista di interviste della classe base, poi popola la mappa interna avente per chiave i nomi delle possibili scelte e per valori le corrispondenti percentuali: essa è recuperabile tramite il metodo `getMappaPercentuali`. Il metodo `getPercentualeIntervistati(String scelta)` restituisce la percentuale delle sole interviste relative alla scelta specificata, o **IllegalArgumentException** se non ve ne sono o se il parametro è null o la stringa vuota. I due metodi `getIntervisteFiltrate` e `getSondaggioFiltrato` restituiscono rispettivamente una lista di interviste e un nuovo **SondaggioPercentuale** riferiti alle sole interviste che soddisfano i parametri (età minima, età massima, sesso) ricevuti come argomento. Mentre i parametri di età sono sempre presenti, il parametro relativo al sesso degli intervistati è opzionale: se non specificato, si intende che devono essere considerate le interviste di entrambi i sessi.
- Il metodo `toString` deve stampare il risultato del sondaggio mostrando una voce per riga, con le percentuali correttamente formattate (tramite opportuno formatter), con due cifre decimali (si veda l'output mostrato nelle figure Swing per un esempio concreto).

Persistenza (zannopoll.persistence)

(punti 7)

Il file di testo `RisultatiSondaggio.txt` contiene i risultati di un sondaggio (nel formato dettagliato più oltre): ogni riga riporta la risposta di un singolo intervistato, con indicazione della sua età (intero) e sesso ('M'/'F'), separati da virgole

```
Insieme per il salume nostrano,54,F
Pizza è progresso,89,F
Movimento vaniglia democratica,38,F
Insieme per il salume nostrano,62,F
Vegetarianesimo domani,87,F
Vegetarianesimo domani,71,M
...
```



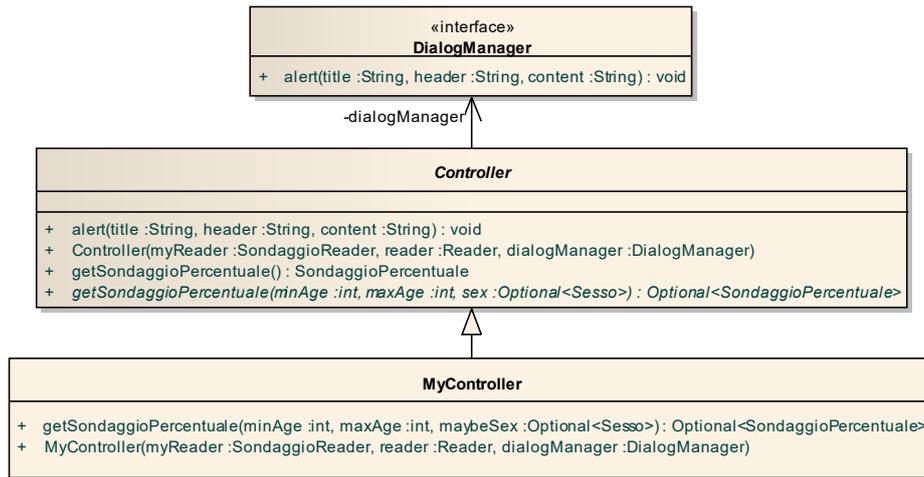
SEMANTICA:

- a) L'interfaccia **SondaggioReader** (fornita) dichiara il metodo `leggiRisposte` che restituisce una lista di **Intervista** lette dal `reader` ricevuto come argomento; in caso di problemi di I/O, il metodo propaga l'opportuna **IOException**, mentre eventuali problemi nel formato del file sono incapsulati in **BadFileFormatException**.
- b) La classe **MySondaggioReader** (da realizzare) implementa **SondaggioReader** secondo tali specifiche.

Controller (zannopoll.ui.controller)

(punti 3)

Il Controller è organizzato secondo il diagramma UML in figura.



SEMANTICA:

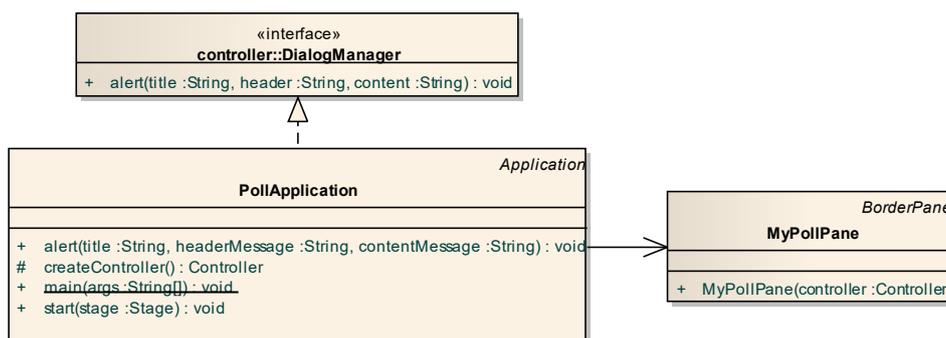
- a) La classe astratta **Controller** (fornita) dichiara l'interfaccia del controller, costituita da due varianti del metodo *getSondaggioPercentuale* (descritte sotto); il metodo ausiliario *alert* può mostrare avvisi all'utente.
- il metodo *getSondaggioPercentuale* nella versione senza argomenti restituisce semplicemente l'istanza di **SondaggioPercentuale** predisposta dal costruttore;
 - nella versione a tre argomenti, *getSondaggioPercentuale* è astratto: le sue concretizzazioni dovranno restituire un **Optional<SondaggioPercentuale>** che costituisca una *versione filtrata* del sondaggio, contenente le sole interviste che rispettano i limiti di età e sesso ricevuti come argomento (analogamente a sopra, il parametro relativo al sesso è opzionale perché se non viene specificato si devono considerare entrambi i sessi).
- b) La classe **MyController (da realizzare)** completa l'implementazione realizzando:
- il costruttore: delega interamente la costruzione alla super classe;
 - il metodo *getSondaggioPercentuale* con tre argomenti restituisce una nuova istanza di **SondaggioPercentuale** (ottenuta mediante il metodo *getSondaggioFiltrato* di **SondaggioPercentuale**): nel caso in cui il risultato (un **Optional**) sia vuoto occorre segnalare il problema chiamando il metodo *alert* del **Controller**.

Interfaccia Utente JavaFX (zannopoll.ui.javafx) per studenti A.A. 2016/17

(punti 10)

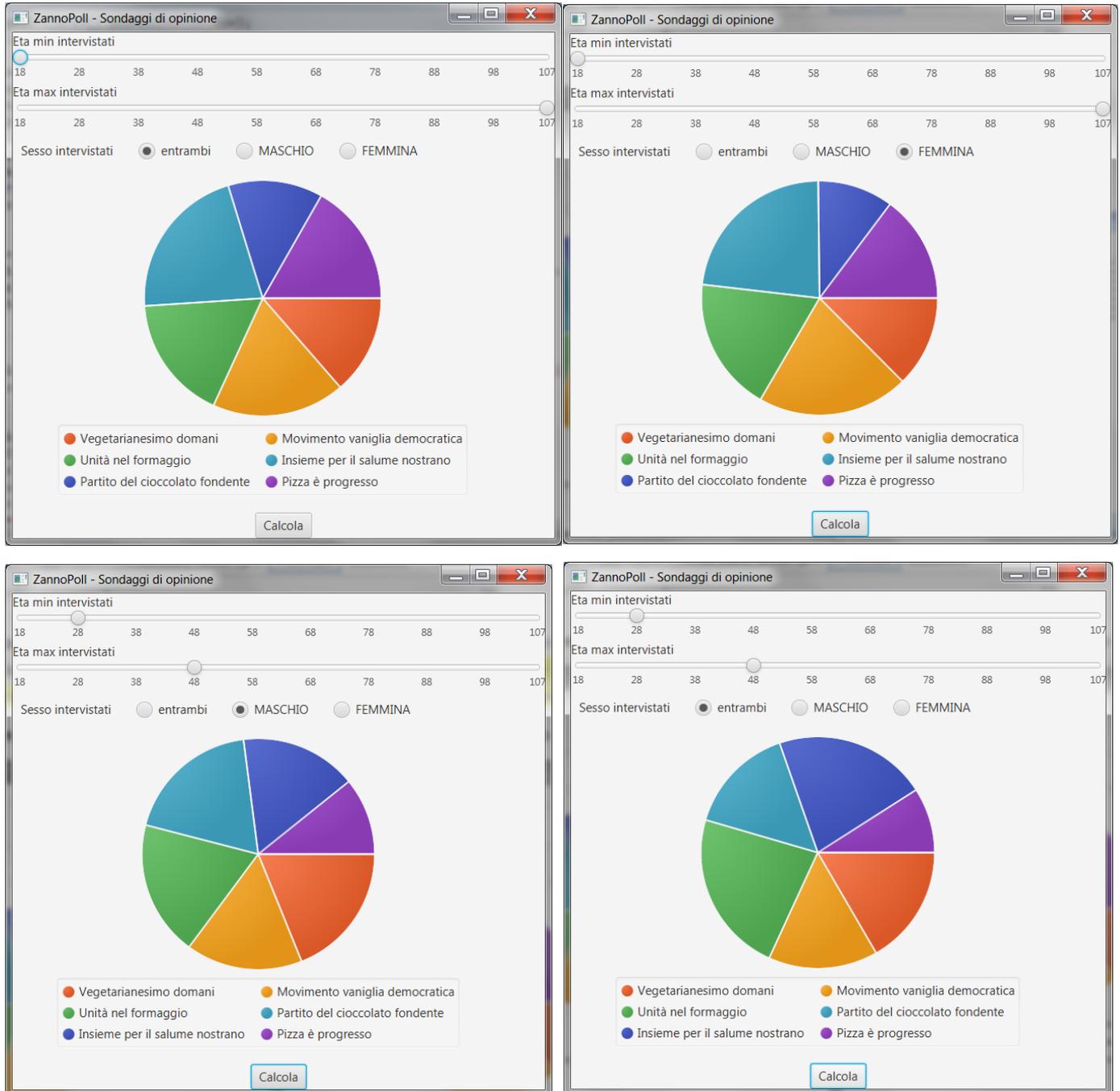
L'interfaccia utente deve essere simile (non necessariamente identica) all'esempio mostrato nelle figure seguenti.

L'architettura segue il modello sotto illustrato:



La classe **PollApplication** (fornita) costituisce l'applicazione JavaFX che si occupa di aprire il file, il controller e incorporare l'apposita istanza di **MyPollPane** (da realizzare). Per consentire di collaudare la GUI anche in assenza della parte di persistenza, è possibile avviare l'applicazione mediante la classe **PollApplicationMock**.

L'interfaccia utente deve essere simile (non necessariamente identica) all'esempio mostrato nella figura seguente.



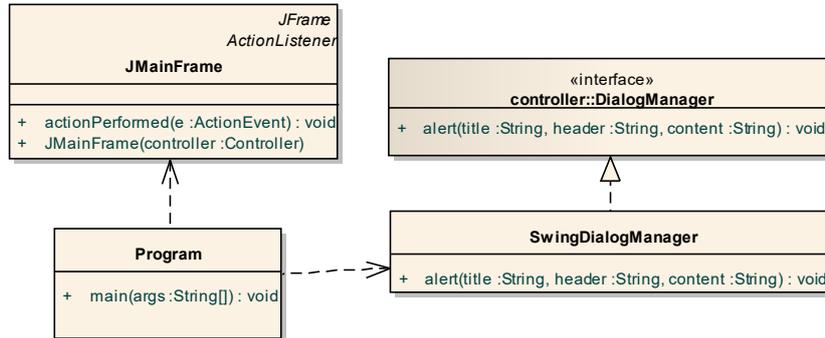
La classe **MyPollPane** (da realizzare) deve estendere **BorderPane**.

- 1) In alto, due **Slider** configurati dall'età minima a quella massima degli intervistati (recuperabili dal controller), passo 10, inizialmente agli estremi min/max rispettivamente, permettono di impostare le età degli intervistati da considerare nel grafico.
- 2) Subito sotto, tre **Radiobutton** consentono analogamente di selezionare le interviste per sesso, scegliendo se considerarle tutte o solo quelle degli uomini o delle donne. Di default, dev'essere selezionato "entrambi".
- 3) Al centro un **PieChart** (senza etichette) mostra la distribuzione percentuale delle scelte, in forma grafica. Esso non si aggiorna a ogni modifica dei filtri di età/sesso, ma solo premendo il pulsante **Calcola**.

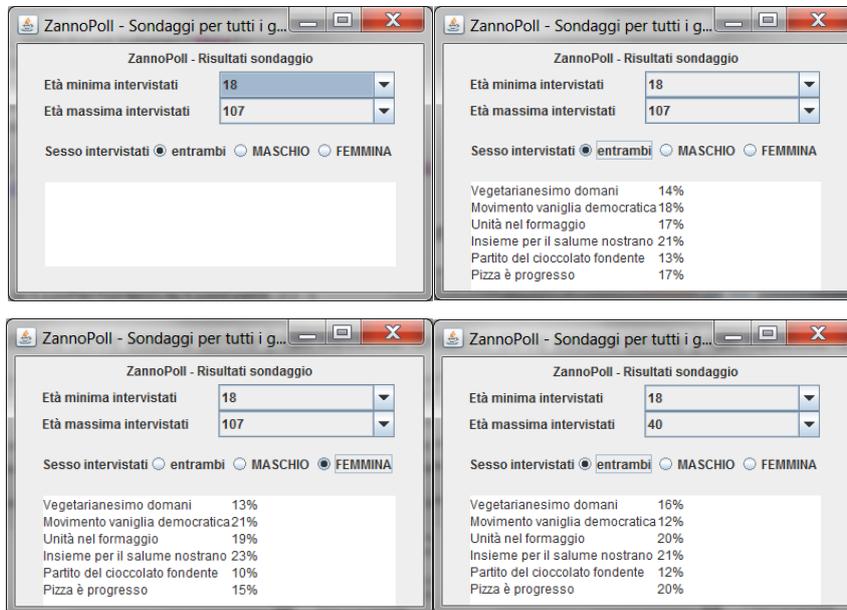
- 4) In basso, il pulsante *Calcola* scatena l'aggiornamento del grafico: i nuovi dati sono ottenuti invocando il metodo *getSondaggioPercentuale* del controller (naturalmente il grafico va aggiornato solo se tale metodo restituisce un risultato non vuoto).

Interfaccia Utente Swing (zannopoll.ui.swing) per studenti A.A. 2015/16 e precedenti

(punti 10)



L'interfaccia utente deve essere simile (non necessariamente identica) all'esempio mostrato nella figura seguente:



La classe **Program** (fornita) contiene il *main* di partenza dell'intera applicazione, che si occupa di aprire il file e creare tutto il necessario. Per consentire di collaudare la GUI anche in assenza della parte di persistenza, è possibile avviare l'applicazione mediante la classe **GUITest**.

La classe **JMainFrame** (*da realizzare*) deve organizzare l'interfaccia come sopra illustrato, ovvero:

- 1) in alto, una label che funge da titolo
- 2) più sotto, precedute da apposite label, due **combo** contenenti tutti i valori dall'età minima all'età massima degli intervistati (recuperabili dal controller): la prima dev'essere pre-settata al valore di default minimo, la seconda al valore di default massimo.
- 3) Nella riga successiva, tre radiobutton *Entrambi*, *Maschio*, *Femmina* consentono di filtrare le interviste per sesso dell'intervistato.
- 4) Infine, in basso, una textarea 5x30, non editabile, costituisce il dispositivo di output.
- 5) A ogni selezione di un'età nella combo, o a ogni scelta di uno dei radiobutton, l'applicazione deve reagire ricalcolando le percentuali ed eventualmente mostrandole nella textarea sottostante; i nuovi dati sono

ottenuti invocando il metodo *getSondaggioPercentuale* del controller (naturalmente il grafico va aggiornato solo se tale metodo restituisce un risultato non vuoto).