

# ESAME DI FONDAMENTI DI INFORMATICA T-2 del 9/9/2020

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'Università di Dentinia, **UniDent**, ha richiesto lo sviluppo di un'applicazione che permetta ai propri studenti di seguire la propria carriera, calcolando anche la media pesata in base ai crediti degli esami sostenuti.

## DESCRIZIONE DEL DOMINIO DEL PROBLEMA

Il *piano didattico* del corso di studi è composto da una serie di *attività formative*, caratterizzate ciascuna da nome (che può contenere spazi), *settore scientifico-disciplinare* (una sigla standardizzata), *tipologia* (una lettera da A a F, a cui corrispondono rispettivamente le attività *di base, caratterizzanti, affini o integrative, a scelta libera, per la prova finale e la lingua inglese, altre*) e naturalmente dal *numero di crediti* (CFU) ad essa assegnato.

Un *esame* associa a una attività formativa una *data* di effettuazione e un *voto* (o giudizio) ottenuto.

In UniDent non esistono *idoneità*: pertanto, i voti possibili sono soltanto quelli da 18 a 30 e Lode, oltre ai giudizi *Assente, Ritirato, Respinto*. Le regole universitarie prevedono inoltre che:

- non si possa ripetere un esame già sostenuto con esito positivo
- non si possa sostenere più volte lo stesso esame (per la stessa attività formativa) nello stesso giorno

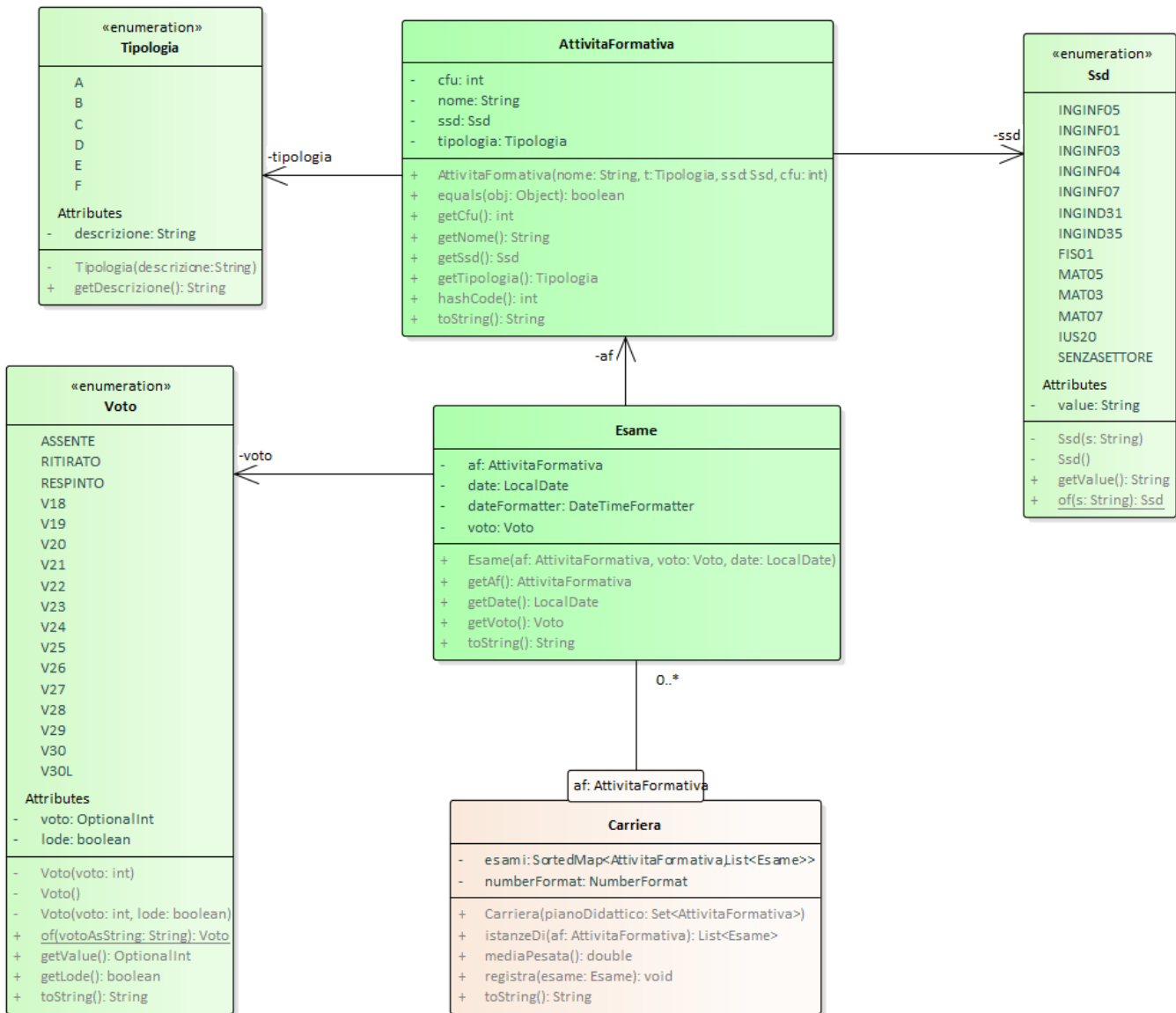
La *carriera* di uno studente comprende l'elenco di tutte le attività formative da sostenere, a ciascuna delle quali sono via via associati gli esami corrispondenti (anche più di uno per ogni attività formativa, nel caso sfortunato in cui l'esito non sia stato inizialmente positivo). Ovviamente, all'inizio la carriera contiene tutte le attività formative ma nessun esame associato.

La *media pesata* degli esami sostenuti è definita come il rapporto fra la somma dei prodotti *voto\*cfu* e la somma dei cfu corrispondenti: ovviamente, i giudizi *Assente, Ritirato, Respinto* non fanno media. Nel caso siano presenti solo esami con giudizio, il risultato è per convenzione NaN. La media deve avere *esattamente due cifre decimali*.

**ESEMPIO:** se uno studente ha sostenuto tre esami, rispettivamente da 12, 9 e 6 cfu, con voti 28, 24 e 30, la media pesata si calcola con la formula  $M = (12*28 + 9*24 + 6*30)/(12+9+6) = 27,11/30$ .

Il piano didattico del corso di studio *Ingegneria Informatica* di **UniDent** è descritto nel file **PianoDidattico.txt**, il cui formato è riportato più oltre.

**TEMPO STIMATO PER SVOLGERE L'INTERO COMPITO: 1h50 – 2h20**



SEMANTICA:

- l'enumerativo **Tipologia** (fornito) elenca le possibili tipologie da A ad F, con le corrispondenti descrizioni.
- l'enumerativo **Ssd** (fornito) elenca i possibili settori scientifico-disciplinari. Oltre ai metodi tipici degli enumerativi, è presente il metodo *factory of* che, data la stringa corrispondente (es. "Ing-Inf/05"), ottiene il corrispondente valore dell'enumerativo (che per ovvie ragioni di sintassi Java non può contenere trattini o barrette). Dualmente, il metodo *getValue* restituisce il valore (stringa) corrispondente. Fra le costanti enumerative è inclusa lo speciale valore SENZASETTORE (a cui corrisponde la stringa "\t\t"), usato per rappresentare in modo uniforme le attività formative di tipologia E ed F, che non hanno settore.
- l'enumerativo **Voto** (fornito) elenca i possibili voti, inclusi i tre giudizi *Assente*, *Ritirato* e *Respinto*. Anche in questo caso, oltre ai metodi tipici degli enumerativi è presente il metodo *factory of* che, data la stringa corrispondente (es. "22", "30L", "RESPINTO", etc.), ottiene il corrispondente valore dell'enumerativo (che per ovvie ragioni di sintassi Java non può contenere solo cifre). Dualmente, il metodo *getValue* restituisce il valore (intero), se esiste, corrispondente al voto: per questo si utilizza un *OptionalInt*. Il metodo *getLode* restituisce *true* per i soli 30 a cui è associata la lode. Un'opportuna implementazione di *toString* consente di ottenere una rappresentazione stringa coerente al valore enumerativo.

- d) la classe **AttivitaFormativa** (fornita) rappresenta l'attività formativa con le proprietà discusse nel *Dominio del Problema*: sono presenti i classici metodi per recuperare gli elementi, produrre un'opportuna stringa descrittiva, nonché *equals* e *hashCode*.
- e) la classe **Esame** (fornita) rappresenta un esame con le proprietà discusse nel *Dominio del Problema*: anche in questo caso sono presenti i classici metodi per recuperare gli elementi e *toString*.
- f) la classe **Carriera (da realizzare)** rappresenta la carriera dello studente. Internamente mantiene una mappa ordinata per nome dell'attività formativa che associa a ogni attività formativa la lista (inizialmente vuota, poi ordinata in senso crescente per data) degli esami sostenuti per tale attività formativa. Metodi:
- il costruttore riceve il piano didattico, inteso come insieme (set) non ordinato delle attività formative: provvede a inizializzare tutte le strutture dati e le entità accessorie necessarie
  - il metodo *registra(Esame)* consente di registrare l'esame ricevuto come argomento, a condizione che quest'ultimo non abbia data precedente a quelli già registrati per la stessa attività formativa e rispetti le regole UniDent (ossia, non sia già stato superato con esito positivo e non sia già stato sostenuto, con qualunque esito, nello stesso giorno). Se l'esame si riferisce a un'attività formativa non presente in carriera, o non rispetta tali regole, dev'essere lanciata apposita **IllegalArgumentException** con messaggio d'errore specifico e adeguatamente descrittivo.
  - Il metodo *istanzeDi(AttivitaFormativa)* restituisce la lista, eventualmente nulla, degli esami corrispondenti a una data attività formativa
  - Il metodo *mediaPesata* calcola la media pesata degli esami finora sostenuti
  - Il metodo *toString* deve produrre una stringa organizzata su più righe, **secondo il formato visibile nelle figure in fondo**. In particolare, dopo una prima riga di intestazione ("Esami sostenuti:"), dev'essere presente una riga per ogni esame, elencando gli esami sostenuti in ordine alfabetico e, in subordine, per data di effettuazione, secondo lo schema sotto illustrato. **NB: è cruciale che le attività formative per cui non sono stati ancora sostenuti esami siano omesse**. Terminato l'elenco degli esami, dopo una riga vuota, dev'essere riportata la media pesata, secondo lo schema sotto illustrato:

Esami sostenuti:			
Fondamenti di Informatica T-2	CFU: 6	Data: 10/07/20	Voto: RITIRATO
Fondamenti di Informatica T-2	CFU: 6	Data: 10/08/20	Voto: 30
Media pesata: 30/30			

#### REQUISITI:

- Le parole chiave ("CFU:", "Data:", "Voto:") devono essere scritte esattamente come indicato, coi ":" finali
- dopo ogni parola chiave deve sempre essere presente almeno uno spazio
- i dati dovrebbero essere opportunamente incolonnati, come negli esempi sopra.

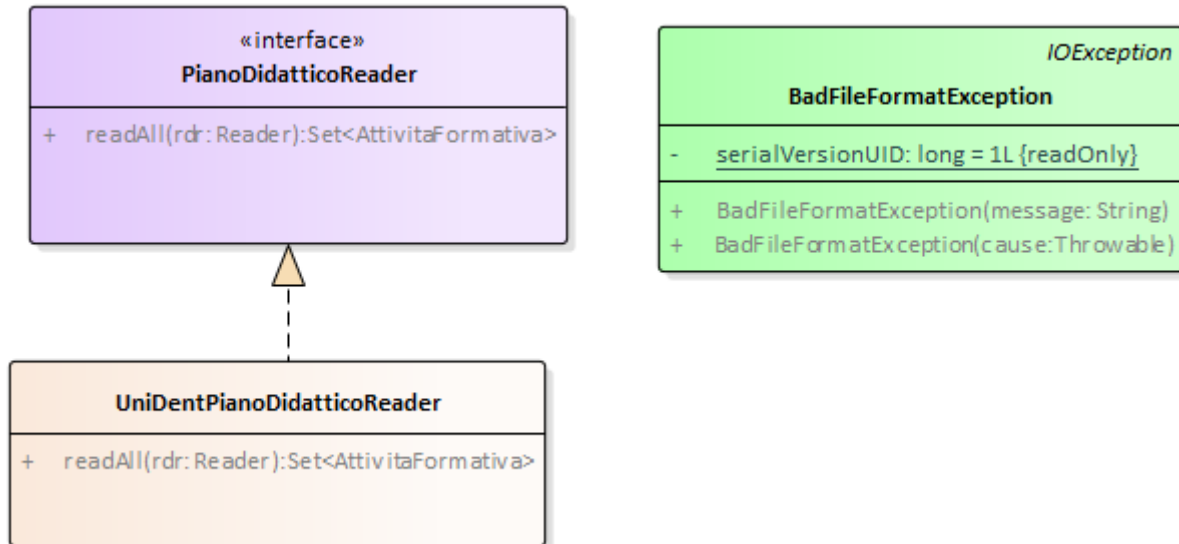
#### Persistenza (*unident.persistence*)

[TEMPO STIMATO: 40-50 minuti] (punti 9)

Il file di testo specifica una attività formativa per riga. Ogni riga contiene nell'ordine, separati da tabulazioni:

- il codice numerico univoco dell'attività formativa
- il nome dell'attività (può contenere spazi, apostrofi e ogni altro carattere utile)
- il periodo (semestre) di effettuazione (una sigla di uno o più caratteri senza spazi intermedi)
- la tipologia (una lettera fra A ed F)
- **solo per le tipologie fra A e D**: il settore scientifico-disciplinare (una sigla standardizzata)
- il numero intero di crediti corrispondenti

27991	ANALISI MATEMATICA T-1	1	A	Mat/05	9
...					
17268	PROVA FINALE	0	E	3	
28072	LABORATORIO DI AMMINISTRAZIONE DI SISTEMI T	2	F	6	
28074	TIROCINIO T	2	F	6	
...					
28014	FONDAMENTI DI TELECOMUNICAZIONI T	2	B	Ing-Inf/03	9
28020	SISTEMI OPERATIVI T	2	B	Ing-Inf/05	9
...					



SEMANTICA:

- a) L'eccezione **BadFileFormatException** (fornita) specializza **IOException** per esprimere l'idea di file formattato in modo scorretto: come tale è a controllo obbligatorio.
- b) L'interfaccia **PianoDidatticoReader** (fornita) dichiara il metodo **readAll**, che legge da un **Reader** (ricevuto come argomento) i dati di tutte le attività formative elencate nel piano didattico, restituendole sotto forma di **Set** (ovviamente, il piano didattico non può prevedere due attività identiche).
- c) La classe **UniDentPianoDidatticoReader** (da realizzare) implementa **PianoDidatticoReader**: non prevede costruttori, si limita a implementare il metodo **readAll** come sopra specificato. In caso di problemi di I/O deve essere propagata l'opportuna IOException, mentre in caso di Reader nullo o altri problemi di formato dei file deve essere lanciata una opportuna BadFileFormatException, il cui messaggio dettagli l'accaduto.

In particolare, il reader deve verificare: 1) che ogni riga contenga il giusto numero di elementi, tenendo conto in particolare dell'assenza del settore scientifico-disciplinare nelle attività di tipologia E ed F; 2) che la tipologia sia valida, ossia compresa fra A ed F; 3) che il settore scientifico-disciplinare sia valido, ossia sia correttamente accettato dal metodo factory di **Ssd**; 4) che il numero di crediti sia intero; 5) che non siano presenti attività formative duplicate.

Parte 2

[TEMPO STIMATO: 20-30 minuti]

(punti: 7)

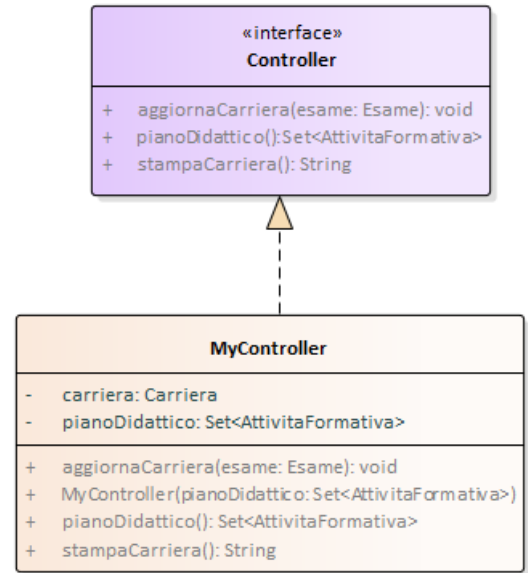
Controller (unident.controller)

(punti 0)

Il Controller (fornito) è organizzato secondo il diagramma UML nella figura alla pagina seguente.

SEMANTICA:

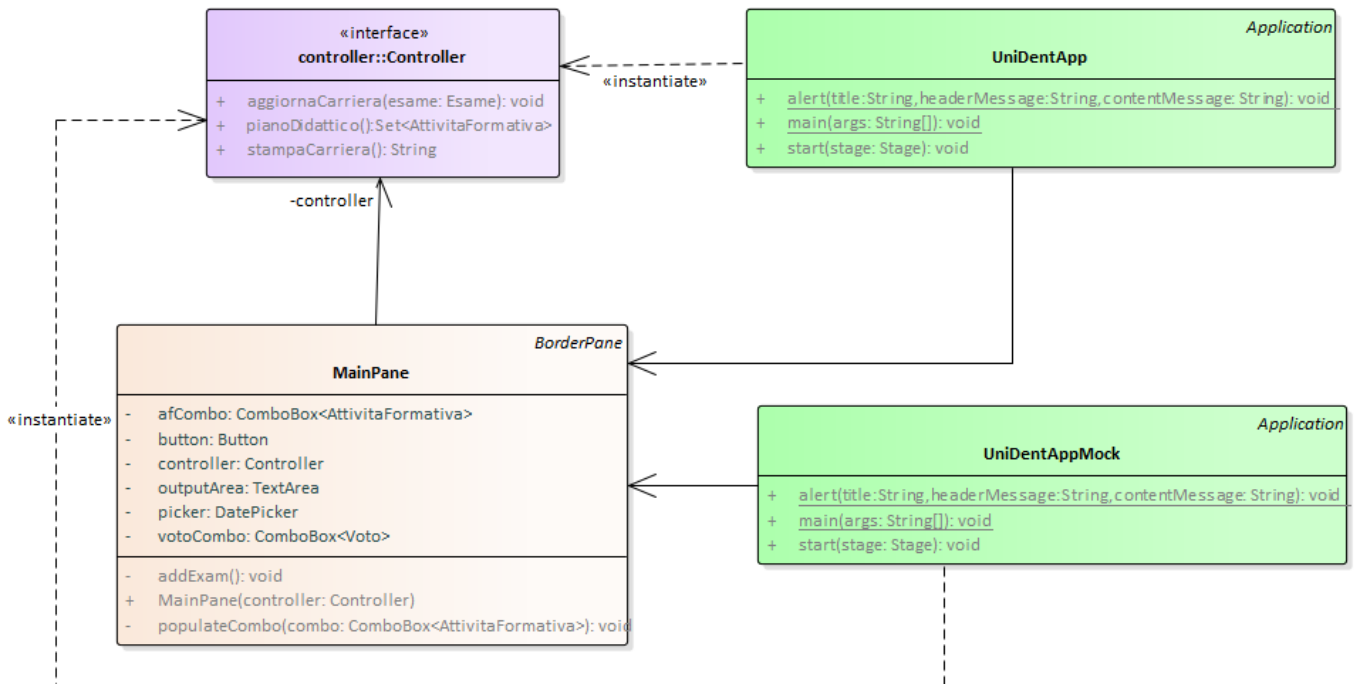
- a) L'interfaccia **Controller** (fornita) dichiara i metodi *pianoDidattico*, *aggiornaCarriera* e *stampaCarriera*.
- b) La classe **MyController** (fornita) implementa tale interfaccia, fornendo:
- il costruttore che, dall'insieme delle attività formative (piano didattico), costruisce e inizializza la **Carriera** dello studente;
  - implementa i tre metodi delegando il lavoro, per quanto necessario, all'istanza di **Carriera** interna.



**Interfaccia utente (unident.ui)**

[TEMPO STIMATO: 20-30 minuti] (punti 7)

L'interfaccia utente è illustrata nelle figure seguenti e segue il modello sotto illustrato:



La classe **UniDentApp** (fornita) costituisce l'applicazione JavaFX che si occupa di aprire i file, creare il controller e incorporare il **MainPane**. Per consentire di collaudare la GUI anche in assenza / in caso di malfunzionamento della parte di persistenza, è possibile avviare l'applicazione mediante la classe **UniDentAppMock**.

Entrambe le classi contengono anche il **metodo statico ausiliario alert**, utile per mostrare avvisi all'utente.

**Il MainPane è fornito parzialmente realizzato: è presente la parte strutturale, mentre manca la parte di popolamento combo e gestione degli eventi.**

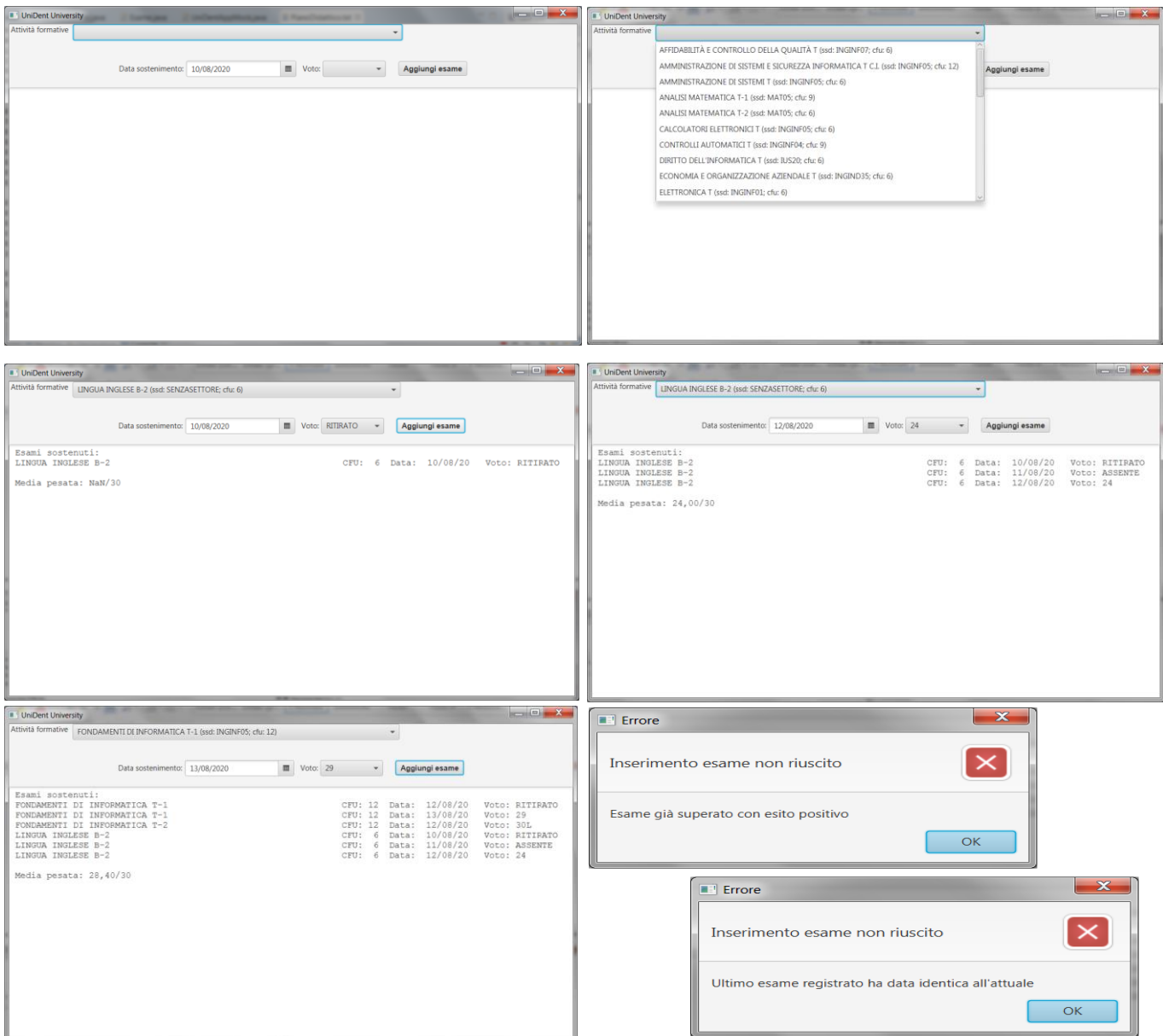
La classe **MainPane** (da completare) estende **BorderPane** e prevede:

- in alto, la **ComboBox** da popolare con l'elenco alfabetico **ordinato** delle attività formative: inizialmente, nessun elemento dev'essere preselezionato (Fig. 1)
- sotto, un **DatePicker** per scegliere la data dell'esame (inizializzato alla data corrente, Fig. 1), una **ComboBox** coi possibili voti (senza alcun elemento preselezionato, Fig. 1) e infine un **Button** "AGGIUNGI ESAME" per inserire l'esame in carriera

3) in basso, una **TextArea** che mostra gli esami finora sostenuti, in ordine alfabetico e in subordine per data crescente (Fig. 3, 4 e 5) seguiti dalla media pesata, su due cifre decimali; quando la media non è calcolabile, dev'essere mostrato NaN (Fig. 3). Al fine di ottenere una presentazione gradevolmente incolonnata, è richiesto l'uso di un font non proporzionale, come il Courier New (vedere figure).

La **parte da completare** comprende l'uso e/o l'implementazione dei seguenti metodi:

- 1) **populateCombo**, che popola la combo con *l'elenco alfabetico ordinato di tutte le attività formative* (Fig. 2)
- 2) **addExam**, da chiamare in risposta all'evento di pressione del **Button**, cerca di inserire l'esame in carriera; se ciò non risulta possibile (perché il controller emette eccezione), dev'essere emesso apposito messaggio di errore tramite **alert** (Fig. 6).



### 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?