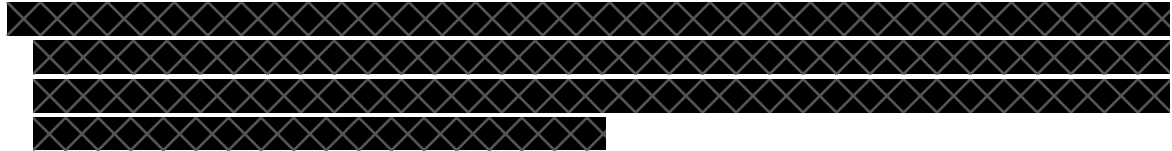


Piano di Studi

Note per la consegna



- Prima di caricare un file su upload, verificare che sia compilabile; cancellare o includere in commenti le parti che provocano errori in compilazione. Può essere utile usare opzione `-Xlint` di `javac`, che controlla l'uso corretto dei tipi generici.
- **Non** vanno definiti package.

Specifica del problema

Lo scopo è definire delle classi che permettano di gestire il piano di studi di uno studente. Il piano contiene gli esami che lo studente ha sostenuto o intende sostenere; è possibile inserire esami esterni, sostenuti cioè in una università diversa da quella a cui si è attualmente iscritti. Gli esami nel piano devono avere nomi distinti. Le classi da definire devono essere pubbliche, inoltre:

- i costruttori e i metodi descritti nella specifica devono essere pubblici;
- i campi di una classe devono essere privati;
- i prototipi sono a volte scritti in modo incompleto (mancano alcuni tipi);
- è possibile aggiungere a una classe ulteriori metodi (pubblici o privati);
- le classi elencate non devono compiere operazioni di stampa (`System.out.println`, ecc.) o di lettura da standard input; le uniche classi che possono farlo sono le classi che implementano una applicazione (classi con metodo `main`);
- è ammesso che i messaggi stampati dal proprio programma abbiano un formato diverso da quelli riportati nel testo, purché le informazioni siano equivalenti e chiaramente identificabili.

Si raccomanda di dare ai campi e ai nuovi metodi dei nomi significativi, che esplicitino il loro significato, eventualmente scrivere commenti chiarificatori.

Classe Voto

Classe che descrive un voto in trentesimi. La classe definisce un costruttore che costruisce una voto di cui si specifica il valore (un intero, ad esempio il valore 28 definisce il voto "28/30"). La classe definisce il seguente metodo pubblico:

- `valoreVoto()`

Restituisce il valore del voto (un intero, ad esempio il valore del voto "28/30" è 28).

Classe VotoConLode

Sottoclasse della classe Voto che descrive il voto “30/30 e lode”. La classe definisce un solo costruttore, privo di argomenti. Di regola il valore di questo voto è 30; per modificare il valore, la classe definisce il seguente metodo pubblico:

- `setValoreLode(val)`

Stabilisce che il valore del voto “30/30 e lode” è `val` (un intero).

Classe Esame

Classe che descrive un esame. La classe definisce i seguenti costruttori:

- Un costruttore che costruisce un esame di cui si specificano nome (stringa), cfu (intero), voto (tipo Voto).
- Un costruttore che costruisce un esame di cui si specificano nome e cfu.

La classe definisce i seguenti metodi pubblici

- `valoreVoto()`

Restituisce il valore del voto di questo esame (un intero); se il voto non è definito, va restituito -1.

- `setVoto(voto)`

Assegna a questo esame il voto specificato (tipo Voto).

- `sostenuto()`

Restituisce `true` se e solo se questo esame è stato sostenuto (ossia, ha un voto).

Classe EsameEsterno

Sottoclasse della classe Esame che descrive un esame esterno. La classe possiede un solo costruttore che costruisce un esame esterno di cui si specificano nome, cfu, voto (tipo Voto) e una descrizione (stringa).

Classe PianoDiStudi

Classe che rappresenta il piano di studi di uno studente. Gli esami nel piano vanno rappresentati mediante un campo privato di tipo `List<Esame>`. L’inserimento di un esame nel piano è possibile se e solo se il piano non contiene un esame il cui nome è uguale al nome dell’esame da inserire (gli esami nel piano devono avere nomi distinti).

La classe definisce un solo costruttore che crea un piano di studi vuoto e i seguenti metodi pubblici.

- `add(esame)`

Inserisce in questo piano l'esame specificato (tipo `Esame`), se questo è possibile.

- `add(nomeEsame, cfu, voto)`

Inserisce in questo piano l'esame di cui si specificano nome, cfu e voto (tipo `Voto`), se questo è possibile.

- `add(nomeEsame, cfu)`

Inserisce in questo piano l'esame di cui si specificano nome e cfu, se questo è possibile.

- `add(nomeEsame, cfu, voto, descrizione)`

Inserisce in questo piano l'esame esterno di cui si specificano nome, cfu, voto (tipo `Voto`) e descrizione (stringa), se questo è possibile.

- `setVoto(nomeEsame, voto)`

Se l'esame con il nome specificato è in questo piano, assegna all'esame il voto specificato (tipo `Voto`). Per implementare il metodo, usare i metodi `get` e `indexOf` di `List<E>`.

- `numeroEsami()`

Calcola il numero degli esami in questo piano.

- `numeroEsamiInterni()`

Calcola il numero degli esami interni (ossia, che non provengono da altri atenei) in questo piano.

- `numeroEsamiSostenuti()`

Calcola il numero degli esami già sostenuti in questo piano.

- `numeroLodi()`

Calcola il numero di esami con votazione "30/30 e lode" in questo piano.

- `totCfu()`

Calcola il numero complessivo di cfu (somma cfu di tutti gli esami in questo piano).

- `totCfuAcquisiti()`

Calcola il numero complessivo di cfu acquisiti (somma cfu degli esami già sostenuti).

Esercizio 1

Compilare il file `Esercizio1.java` in Fig. 1. Il comando `java Esercizio1` deve stampare le linee in Fig. 2; l'ordine in cui gli esami sono stampati non è rilevante.

```

public class Esercizio1 {
    public static void main(String[] args) {
        PianoDiStudi mioPiano = new PianoDiStudi();
        mioPiano.add( "Algebra 1", 9, new Voto(28) );
        mioPiano.add( "Geometria 1", 6, new Voto(24) );
        mioPiano.setVoto( "Programmazione 1", new Voto(25) ); // non fa nulla (esame non inserito)
        mioPiano.add( "Programmazione 1", 6, new Voto(30) );
        mioPiano.add( "Geometria 2", 9);
        mioPiano.add( "Programmazione 2", 6 );
        mioPiano.add( "Analisi Matematica 1", 9 );
        mioPiano.add( "Analisi Matematica 2", 6 );
        mioPiano.add( "Geometria 2", 6); // non inserito (duplicazione)
        mioPiano.add( "Geometria 2", 9, new VotoConLode() ); // non inserito (duplicazione)
        mioPiano.add( "Calcolo Numerico 1", 9, new VotoConLode() );
        mioPiano.setVoto( "Geometria 2", new Voto(27) );
        mioPiano.setVoto( "Analisi Matematica 1", new Voto(20) );
        mioPiano.setVoto( "Analisi Matematica 1", new Voto(24) ); // sovrascrive voto precedente
        mioPiano.setVoto( "Analisi Matematica 2", new VotoConLode() );
        mioPiano.add( new EsameEsterno( "Analisi Matematica 3", 6, new Voto(27), "Politecnico MI" ) );
        mioPiano.add( new EsameEsterno( "Algebra 1", 6, new Voto(28), "Politecnico MI" ) ); // non inserito (duplicazione)
        mioPiano.add( new EsameEsterno( "Fisica Generale 1", 9, new VotoConLode(), "Politecnico MI" ) );
        mioPiano.add( new Esame( "Probabilita'", 9 ) );
        mioPiano.add( new Esame( "Fisica Matematica 1", 6 ) );
        mioPiano.add( new Esame( "Fisica Matematica 2", 6 ) );
        mioPiano.setVoto( "Fisica Matematica 1", new Voto(26) );
        /*** FINE INSERIMENTO ESAMI ***/
        System.out.println( mioPiano );
        System.out.println( "Numero esami inseriti ----> " + mioPiano.numeroEsami() );
        System.out.println( "Numero esami interni ----> " + mioPiano.numeroEsamiInterni() );
        System.out.println( "Numero esami sostenuti ----> " + mioPiano.numeroEsamiSostenuti() );
        System.out.println( "Numero lodi -----> " + mioPiano.numeroLodi() );
        System.out.println( "Totale CFU inseriti -----> " + mioPiano.totCfu() );
        System.out.println( "Totale CFU acquisiti -----> " + mioPiano.totCfuAcquisiti() );
    } // end main
} //end class

```

Figura 1: File Esercizio1.java.

Esercizio 2

Aggiungere alla classe `PianoDiStudi` il metodo pubblico:

- `media()`

Calcola la media degli esami sostenuti (media pesata); quindi, se gli esami sostenuti sono e_1, \dots, e_n , la media è:

$$\frac{\sum_{k=1}^n \text{cfu}(e_k) \cdot \text{valoreVoto}(e_k)}{\sum_{k=1}^n \text{cfu}(e_k)}$$

Definire la applicazione `Esercizio2` che costruisce un piano di studi come in `Esercizio1`. Dopo che tutti gli esami sono stati inseriti, va stampata la media dei voti e successivamente la media dei voti assumendo che il valore del voto “30/30 e lode” sia 32. Si deve ottenere:

```

media -----> 27.64
media (lode = 32) ---> 28.28

```

```

Algebra 1.....9 CFU - 28/30
Geometria 1.....6 CFU - 24/30
Programmazione 1.....6 CFU - 30/30
Geometria 2.....9 CFU - 27/30
Programmazione 2.....6 CFU
Analisi Matematica 1....9 CFU - 24/30
Analisi Matematica 2....6 CFU - 30/30 e lode
Calcolo Numerico 1.....9 CFU - 30/30 e lode
Analisi Matematica 3....6 CFU - 27/30 *ESAME ESTERNO* Politecnico MI
Fisica Generale 1.....9 CFU - 30/30 e lode *ESAME ESTERNO* Politecnico MI
Probabilita'.....9 CFU
Fisica Matematica 1.....6 CFU - 26/30
Fisica Matematica 2.....6 CFU

Numero esami inseriti -----> 13
Numero esami interni -----> 11
Numero esami sostenuti ----> 10
Numero lodi -----> 3
Totale CFU inseriti -----> 96
Totale CFU acquisiti -----> 75

```

Figura 2: Linee stampate dalla applicazione `Esercizio1`.

Esercizio 3

Aggiungere alla classe `PianoDiStudi` i seguenti metodi pubblici.

- `sort()`

Ordina gli esami in questo piano in ordine alfabetico. Tale ordinamento va definito come l'*ordinamento naturale* (*natural order*) fra gli esami.

- `sort(cmp)`

Ordina gli esami in questo piano in base all'ordinamento definito dal comparatore `cmp` (oggetto di tipo `Comparator`).

Definire la applicazione `Esercizio3` che costruisce un piano di studi come in `Esercizio1`. Dopo che tutti gli esami sono stati inseriti, compiere le seguenti operazioni:

- (a) Ordinare il piano di studi in ordine alfabetico e stampare il piano.
- (b) Ordinare il piano di studi in ordine decrescente rispetto al voto, partendo dal voto massimo "30/30 e lode". A parità di voto, gli esami vanno ordinati in ordine alfabetico; gli esami non sostenuti vanno posti per ultimi, ordinati in ordine alfabetico (vedi Fig. 3).

Il comparatore da usare per il punto (b) va definito nel file `Esercizio3.java`.

```

Analisi Matematica 2....6 CFU - 30/30 e lode
Calcolo Numerico 1.....9 CFU - 30/30 e lode
Fisica Generale 1.....9 CFU - 30/30 e lode *ESAME ESTERNO* Politecnico MI
Programmazione 1.....6 CFU - 30/30
Algebra 1.....9 CFU - 28/30
Analisi Matematica 3....6 CFU - 27/30 *ESAME ESTERNO* Politecnico MI
Geometria 2.....9 CFU - 27/30
Fisica Matematica 1.....6 CFU - 26/30
Analisi Matematica 1....9 CFU - 24/30
Geometria 1.....6 CFU - 24/30
Fisica Matematica 2.....6 CFU
Probabilita'.....9 CFU
Programmazione 2.....6 CFU

```

Figura 3: Linee stampate dalla applicazione `Esercizio3` per il punto (b).

Esercizio 4

Definire l'eccezione `VotoException` per rappresentare un errore nella specifica di un voto e l'eccezione `CfuException` per rappresentare un errore nella specifica dei crediti. Più precisamente, vincoli su voti e cfu sono:

- (V1) un voto nel piano di studi deve essere compreso fra 18 e 30 (estremi inclusi);
- (V2) il valore della lode deve essere compreso fra 30 e 33 (estremi inclusi);
- (V3) cfu deve essere compreso fra 1 e 9 (estremi inclusi).

Modificare la classi definite sollevando le eccezioni `VotoException` e `CfuException` in tutti i punti in cui è possibile violare uno dei vincoli (V1), (V2) e (V3). Si suggerisce di definire le due eccezioni come eccezioni *non controllate* (*unchecked*); successivamente si può migliorare l'implementazione definendole come eccezioni *controllate* (*checked*).

Applicazione `AppPianoDiStudi`

L'obiettivo è definire una applicazione `AppPianoDiStudi` che permette di gestire un piano di studi. La applicazione legge da standard input una sequenza di linee, ciascuna delle quali descrive una azione da compiere; subito dopo che la linea è letta, va eseguita l'operazione corrispondente. Nel descrivere il formato delle linee di input assumiamo che:

- `cfu` è un intero.
- `specificaVoto` è un intero oppure la lettera L.

Una linea letta da standard input può avere uno dei seguenti formati.

- **E#nomeEsame#cfu**

Inserisce nel piano di studi l'esame con nome e cfu specificati. Se il piano contiene già un esame con il nome specificato, non viene compiuta alcuna operazione. Se cfu non è nell'intervallo corretto, va stampato su standard error un opportuno messaggio; per gestire questo caso, intercettare l'eccezione `CfuException`.

- **E#nomeEsame#cfu#specificaVoto**

Inserisce nel piano di studi l'esame con nome, cfu e voto specificati. Se il piano contiene già un esame con il nome specificato, non viene compiuta alcuna operazione. Se cfu oppure il voto non sono nell'intervallo corretto, va stampato su standard error un opportuno messaggio; per gestire questi casi, intercettare le eccezioni `CfuException` e `VotoException`.

- **EE#nomeEsame#cfu#specificaVoto#descrizione**

Inserisce nel piano di studi l'esame esterno con nome, cfu, e descrizione specificati. Se il piano contiene già un esame con il nome specificato, non viene compiuta alcuna operazione. Se cfu oppure il voto non sono nell'intervallo corretto, va stampato su standard error un opportuno messaggio; per gestire questi casi, intercettare le eccezioni `CfuException` e `VotoException`.

- **S#nomeEsame#specificaVoto**

Assegna all'esame di cui si specifica il nome il voto descritto da `specificaVoto`. Se il piano non contiene alcun esame con il nome specificato, non viene compiuta alcuna operazione. Se il voto non è nell'intervallo corretto, va stampato su standard error un opportuno messaggio; per gestire questo caso, intercettare l'eccezione `VotoException`.

- **V#valoreLode**

Assegna al voto "30/30 e lode" il valore specificato. Si assume che `valoreLode` sia un intero. Se il valore della lode non è nell'intervallo corretto, va stampato su standard error un opportuno messaggio; per gestire questo caso, intercettare l'eccezione `VotoException`.

- **PA**

Stampa il piano di studi in ordine alfabetico, comprese le informazioni aggiuntive stampate negli esercizi precedenti (esami interni/esterni, media, conteggio cfu, ecc.).

- **PV**

Come PA, con la differenza che gli esami sono elencati come descritto nel punto (b) dell'Esercizio 3.

La applicazione `AppPianoDiStudi` deve definire i seguenti metodi statici:

- `buildVoto(String specificaVoto)`

Si assume che `specificaVoto` abbia il formato descritto sopra (intero o “L”). Il metodo restituisce il voto corrispondente.

- `pianoToString(PianoDiStudi piano)`

Restituisce una stringa che descrive il piano, comprese le informazioni aggiuntive. La media va arrotondata a due cifre decimali.

- `runLineaInput(String lineaInput, PianoDiStudi piano).`

Si assume che `lineaInput` sia una linea di input, nel formato descritto sopra, e che `piano` sia il piano di studi che si sta gestendo. Il metodo esegue la linea di input, come specificato sopra.

Nel file `AppPianoDiStudi.java` va anche definito un comparatore per gestire l’operazione PV (si può copiare quello già definito per l’Esercizio 3). Si consiglia di **non** catturare le eccezioni all’interno del metodo `runLineaInput`, ma direttamente nel metodo `main`.

Esempio di esecuzione

Supponiamo che le linee di input siano:

```
E#Algebra 1#9#24
E#Algebra 1#9#30
E#Algebra 2#10#30
E#Algebra 2#0#30
E#Algebra 2#6#30
E#Geometria 1#6#L
E#Analisi Matematica 1#9
E#Algebra 3#6#17
E#Algebra 3#6#31
E#Algebra 3#9#30
E#Programmazione 2#9
EE#Fisica 1#9#30#Politecnico T0
EE#Fisica 2#28#28#Politecnico T0
EE#Fisica 2#10#28#Politecnico T0
EE#Fisica 2#6#10#Politecnico T0
EE#Fisica 2#6#40#Politecnico T0
EE#Fisica 2#6#29#Politecnico T0
EE#Algebra 1#6#28#Politecnico T0
EE#Programmazione 1#9#L#Politecnico T0
E#Probabilita' #9#29
E#Logica 1#6
```


E#Logica 2#6
 E#Fisica 1#6
 E#Geometria 2#9
 EE#Sistemi operativi#9#25##Politecnico TO
 E#Algoritmi e strutture dati#9#25
 S#Programmazione 3#25
 E#Algebra lineare numerica#6
 E#Metodi e modelli matematici per le applicazioni#6
 S#Programmazione 2#25
 PA
 PV
 V#29
 V#34
 V#33
 PA

La applicazione AppPianoDiStudi produce il seguente output:

```

===== PIANO DI STUDI (ORDINE ALFABETICO) =====
Algebra 1.....9 CFU - 24/30
Algebra 2.....6 CFU - 30/30
Algebra 3.....9 CFU - 30/30
Algebra lineare numerica.....6 CFU
Algoritmi e strutture dati.....9 CFU - 25/30
Analisi Matematica 1.....9 CFU
Fisica 1.....9 CFU - 30/30 *ESAME ESTERNO* Politecnico TO
Fisica 2.....6 CFU - 29/30 *ESAME ESTERNO* Politecnico TO
Geometria 1.....6 CFU - 30/30 e lode
Geometria 2.....9 CFU
Logica 1.....6 CFU
Logica 2.....6 CFU
Metodi e modelli matematici per le applicazioni....6 CFU
Probabilita'.....9 CFU - 29/30
Programmazione 1.....9 CFU - 30/30 e lode *ESAME ESTERNO* Politecnico TO
Programmazione 2.....9 CFU - 25/30
Sistemi operativi.....9 CFU - 25/30 *ESAME ESTERNO* Politecnico TO
Numero esami inseriti ----> 17
Numero esami interni ----> 13
Numero esami sostenuti ----> 11
Numero lodi -----> 2
Totale CFU inseriti -----> 132
Totale CFU acquisiti -----> 90
media -----> 27.73 (valore lode: 30)
===== PIANO DI STUDI (ORDINE PER VOTO) =====
Geometria 1.....6 CFU - 30/30 e lode
Programmazione 1.....9 CFU - 30/30 e lode *ESAME ESTERNO* Politecnico TO
Algebra 2.....6 CFU - 30/30
Algebra 3.....9 CFU - 30/30
Fisica 1.....9 CFU - 30/30 *ESAME ESTERNO* Politecnico TO
Fisica 2.....6 CFU - 29/30 *ESAME ESTERNO* Politecnico TO
Probabilita'.....9 CFU - 29/30
Algoritmi e strutture dati.....9 CFU - 25/30
Programmazione 2.....9 CFU - 25/30
Sistemi operativi.....9 CFU - 25/30 *ESAME ESTERNO* Politecnico TO
Algebra 1.....9 CFU - 24/30
Algebra lineare numerica.....6 CFU
  
```

```

Analisi Matematica 1.....9 CFU
Geometria 2.....9 CFU
Logica 1.....6 CFU
Logica 2.....6 CFU
Metodi e modelli matematici per le applicazioni....6 CFU
Numero esami inseriti ----> 17
Numero esami interni ----> 13
Numero esami sostenuti ----> 11
Numero lodi -----> 2
Totale CFU inseriti -----> 132
Totale CFU acquisiti -----> 90
media -----> 27.73 (valore lode: 30)
===== PIANO DI STUDI (ORDINE ALFABETICO) =====
Algebra 1.....9 CFU - 24/30
Algebra 2.....6 CFU - 30/30
Algebra 3.....9 CFU - 30/30
Algebra lineare numerica.....6 CFU
Algoritmi e strutture dati.....9 CFU - 25/30
Analisi Matematica 1.....9 CFU
Fisica 1.....9 CFU - 30/30 *ESAME ESTERNO* Politecnico TO
Fisica 2.....6 CFU - 29/30 *ESAME ESTERNO* Politecnico TO
Geometria 1.....6 CFU - 30/30 e lode
Geometria 2.....9 CFU
Logica 1.....6 CFU
Logica 2.....6 CFU
Metodi e modelli matematici per le applicazioni....6 CFU
Probabilita'.....9 CFU - 29/30
Programmazione 1.....9 CFU - 30/30 e lode *ESAME ESTERNO* Politecnico TO
Programmazione 2.....9 CFU - 25/30
Sistemi operativi.....9 CFU - 25/30 *ESAME ESTERNO* Politecnico TO
Numero esami inseriti ----> 17
Numero esami interni ----> 13
Numero esami sostenuti ----> 11
Numero lodi -----> 2
Totale CFU inseriti -----> 132
Totale CFU acquisiti -----> 90
media -----> 28.23 (valore lode: 33)

```

Vengono inoltre stampati su standard output i seguenti messaggi d'errore:

```

CfuException: cfu non validi: 10 (devono essere compresi fra 1 e 9)
CfuException: cfu non validi: 0 (devono essere compresi fra 1 e 9)
VotoException: voto non valido: 17 - (deve essere compreso fra 18 e 30)
VotoException: voto non valido: 31 - (deve essere compreso fra 18 e 30)
CfuException: cfu non validi: 28 (devono essere compresi fra 1 e 9)
CfuException: cfu non validi: 10 (devono essere compresi fra 1 e 9)
VotoException: voto non valido: 10 - (deve essere compreso fra 18 e 30)
VotoException: voto non valido: 40 - (deve essere compreso fra 18 e 30)
VotoException: Valore della lode non valido: 29 (deve essere compreso fra 30 e 33)
VotoException: Valore della lode non valido: 34 (deve essere compreso fra 30 e 33)

```

Se standard output e standard error non sono rediretti, i messaggi su standard input e standard error sono fra lore intercalati.