Corso di Laurea in Ingegneria Informatica

Prova scritta di *Fondamenti di informatica e laboratorio* – Modulo 2

9 settembre 2019

**Tempo a disposizione: 2 ore**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Cognome e nome:*** |  | ***Matricola:*** |  | ***Corso:*** | ***◊ A  ◊ B*** |

Si vuole implementare in Java un sistema che supporti il Parlamento di una certa nazione. Il sistema gestisce informazioni sui parlamentari (istanze della classe *Parlamentare*), sui disegni di legge (istanze della classe *DisegnoDiLegge*) e sulle votazioni (istanze della classe *Votazione*).

Ogni *Parlamentare* è identificato da un *codice* *fiscale* ed è caratterizzato dalla regione di appartenenza del collegio in cui è stato eletto e dal suo partito politico, rappresentato da una stringa.

Un *DisegnoDiLegge* già sottoposto a votazione da parte del Parlamento è identificato da un *codice* ed è caratterizzato dal nome, dalla lista dei codici fiscali dei parlamentari proponenti, dalla *data* in cui è stato votato (rappresentabile per semplicità con un intero) e dall’esito della votazione (approvazione o non approvazione), rappresentato con un booleano.

Ciascun oggetto della classe *Votazione* è identificato da un *codice* ed è caratterizzato dal codice fiscale del parlamentare, dal codice del *DisegnoDiLegge* e dal voto espresso (favorevole o contrario), rappresentato con un booleano.

Si implementino in Java le classi *Parlamentare*, *DisegnoDiLegge* e *Votazione*. Si implementi inoltre la classe *Sistema,* scegliendo opportunamente le classi Java più adeguate a gestire tutte le informazioni su parlamentari, disegni di legge e votazioni.

Oltre a scrivere eventuali metodi che si ritengono necessari per implementare l’applicazione, occorre fornire almeno i seguenti metodi nella classe *Sistema*:

1. *public ArrayList<DisegnoDiLegge>* ***m1****(String r).* Il metodo restituisce i disegni di legge approvati soltanto con voti favorevoli espressi da parlamentari eletti in collegi della regione *r*.
2. *public ArrayList<Parlamentare>* ***m2****(String p).* Il metodo restituisce l’elenco dei parlamentari del partito politico *p* che hanno espresso voto contrario su qualche disegno di legge proposto da almeno un parlamentare dello stesso partito *p*.
3. *public ArrayList<String>* ***m3****(DisegnoDiLegge d).* Il metodo restituisce l’elenco dei partiti politici i cui parlamentari hanno votato a favore del disegno di legge *d,* ordinatoin modo decrescente in base al numero di parlamentari di quel partito che hanno espresso tali voti favorevoli*.*
4. *public HashMap<DisegnoDiLegge, int[]>* ***m4****(int d1, int d2)*. Il metodo restituisce una *HashMap* in cui, per ogni disegno di legge votato nell’intervallo di date [*d1,d2*], è riportato un array di due interi che riportano il numero di voti favorevoli e di voti contrari espressi dai parlamentari per quel disegno di legge*.*

ESEMPIO:

Si supponga che nel sistema siano stati inseriti i seguenti parlamentari:

* p1 = {codiceFiscale=”RSS”, regione=”Calabria”, partito=”partito1”}
* p2 = {codiceFiscale=”BNC”, regione=”Lombardia”, partito=”partito2”}
* p3 = {codiceFiscale=”GIA”, regione=”Lombardia”, partito=”partito2”}
* p4 = {codiceFiscale=”VRD”, regione=”Calabria”, partito=”partito3”}
* p5 = {codiceFiscale=”MAR”, regione=”Calabria”, partito=”partito1”}

i seguenti disegni di legge:

* ddl1 = {codice = 1, nome = “DDL1”, listaParlamentari= {"RSS, “BNC", “MAR”}, data = 1, esito = True}
* ddl2 = {codice = 2, nome = “DDL2”, listaParlamentari= {"VRD", “MAR”}, data = 2, esito = False}
* ddl3 = {codice = 3, nome = “DDL3”, listaParlamentari= {“BNC”, "GIA”, VRD"}, data = 3, esito = True}

le seguenti votazioni:

* {codice= 1, parlamentare=”RSS”, disegnoDiLegge=1, Voto=True}
* {codice= 2, parlamentare=”BNC”, disegnoDiLegge=1, Voto=False}
* {codice= 3, parlamentare=”MAR”, disegnoDiLegge=1, Voto=True}
* {codice= 4, parlamentare= “VRD”, disegnoDiLegge=1, Voto=True}
* {codice= 5, parlamentare=”GIA”, disegnoDiLegge=1, Voto=False}
* {codice= 6, parlamentare= “RSS”, disegnoDiLegge=2, Voto=False}
* {codice= 7, parlamentare= “BNC”, disegnoDiLegge=2, Voto=False}
* {codice= 8, parlamentare= “MAR”, disegnoDiLegge=2, Voto=True}
* {codice= 9, parlamentare=” VRD”, disegnoDiLegge=2, Voto=True}
* {codice= 10, parlamentare= ”GIA”, disegnoDiLegge=2, Voto=False}
* {codice= 11, parlamentare= “RSS”, disegnoDiLegge=3, Voto=False}
* {codice= 12, parlamentare= “BNC”, disegnoDiLegge=3, Voto=True}
* {codice= 13, parlamentare= “MAR”, disegnoDiLegge=3, Voto=False}
* {codice= 14, parlamentare= ”VRD”, disegnoDiLegge=3, Voto=True}
* {codice= 15, parlamentare=”GIA”, disegnoDiLegge=3, Voto=True}

Allora:

* Se invocato con il parametro *r=”Calabria”,* il metodo ***m1*** restituisce la lista *[ddl1]*, infatti il disegno di legge *ddl1* è stato votato solo dai parlamentari *RSS*, *MAR e VRD*, tutti eletti in collegi della regione *Calabria* .
* Se invocato con il parametro *p=”partito1”,* il metodo ***m2*** restituisce la lista *[p1]*, infatti il parlamentare *p1* ha votato contro il disegno di legge *ddl2* proposto dal collega *p2*, ed entrambi i parlamentari appartengono al partito *partito1*.
* Se invocato con il parametro *d=ddl3,* il metodo ***m3*** restituisce la lista *[“partito2”, “partito3”]*, infatti il disegno di legge *ddl3* è stato votato dai parlamentari *BNC* e *GIA* del partito *partito2*, e dal parlamentare *VRD* del partito *partito3*.
* Se invocato con i parametri *d1=2* e *d2=3, il metodo* ***m4***restituisce la seguente mappa:

|  |  |
| --- | --- |
| *ddl2* | *[2,3]* |
| *ddl3* | *[3,2]* |

Infatti, i disegni di legge *ddl2* e *ddl3* sono stati proposti nelle date 2 e 3, rispettivamente, ed hanno ricevuto 2 e 3 voti a favore, da parte dei parlamentari *MAR* e *VRD* per *ddl2*, e *BNC*, *VRD* e *GIA* per *ddl3*, e 3 e 2 voti contrari, da parte dei parlamentari *RSS*, *BNC* e *GIA* per *ddl2*, e *RSS* e *MAR* per *ddl3*.