

Esercizio 1

Si implementi in Java una classe *Sistema* che fornisca metodi per l'analisi di dati relativi all'installazione di impianti di energia rinnovabile all'interno di un condominio. Si supponga che le classi *Appartamento* e *Impianto* siano già disponibili e forniscano i seguenti metodi:

Classe *Appartamento*:

- `public String getCodice()`, che restituisce il codice che identifica l'appartamento.
- `public String getScala()`, che restituisce il codice della scala in cui si trova l'appartamento.
- `public boolean equals(Object o)`.
- `public String toString()`.

Classe *Impianto*:

- `public String getAppartamento()`, che restituisce il codice dell'appartamento a cui sarà collegato l'impianto.
- `public int getPrezzoInstallazione()`, che restituisce il prezzo di installazione dell'impianto.
- `public int getDataInstallazione()`, che restituisce la data di installazione dell'impianto (espressa per semplicità come intero).
- `public String getTipo()`, che restituisce il tipo ("fotovoltaico" o "solare termico") dell'impianto.
- `public LinkedList<String> getTecnici()`, che restituisce la lista dei nomi dei tecnici che parteciperanno all'installazione dell'impianto.
- `public boolean equals(Object o)`.
- `public String toString()`.

La classe *Sistema* contiene le liste degli appartamenti e degli impianti. Oltre ad eventuali metodi che si ritengano utili, si includano nella classe almeno i seguenti metodi:

- `public LinkedList<String> scaleUnicoTipo()`. Il metodo restituisce la lista dei codici delle scale che soddisfano le seguenti condizioni:
 - è prevista l'installazione di almeno un impianto nella scala;
 - tutti gli impianti che saranno installati nella scala sono dello stesso tipo.
- `public String tecnicoMax()`. Il metodo restituisce il nome del tecnico che produrrà il massimo incasso. L'incasso prodotto da un tecnico è la somma dei prezzi delle installazioni a cui il tecnico parteciperà. Se più tecnici soddisfano la condizione, il metodo restituisce uno qualsiasi di essi.
- `public String scalaSimile(String cod, int d1, int d2)`. Il metodo restituisce il codice della scala in cui è previsto, tra la data *d1* e la data *d2* (incluse), un numero di installazioni uguale a quello previsto, nello stesso intervallo di date, nella scala con codice *cod*. Se più scale soddisfano la condizione, il metodo restituisce una qualsiasi di esse.

Esempio. Si assuma che i dati a disposizione siano i seguenti:

- Appartamenti:
 - {Codice "A1", scala "S1"}
 - {Codice "A2", scala "S1"}
 - {Codice "A3", scala "S2"}
 - {Codice "A4", scala "S2"}
 - {Codice "A5", scala "S3"}
 - {Codice "A6", scala "S3"}
- Impianti:
 - {Appartamento "A1", prezzo 120, data 10, tipo "fotovoltaico", tecnici ["Verdi", "Bianchi"]}
 - {Appartamento "A1", prezzo 100, data 20, tipo "solare termico", tecnici ["Verdi", "Rossi"]}

- {Appartamento "A2", prezzo 120, data 30, tipo "solare termico", tecnici ["Rossi", "Bianchi"]}
- {Appartamento "A3", prezzo 100, data 15, tipo "fotovoltaico", tecnici ["Verdi", "Rossi"]}
- {Appartamento "A4", prezzo 120, data 20, tipo "fotovoltaico", tecnici ["Verdi"]}
- {Appartamento "A5", prezzo 100, data 25, tipo "fotovoltaico", tecnici ["Bianchi"]}
- {Appartamento "A6", prezzo 110, data 35, tipo "solare termico", tecnici ["Rossi", "Bianchi"]}

Allora:

- *public LinkedList<String> scaleUnicoTipo()* restituisce ["S2"] perché è prevista l'installazione di almeno un impianto nella scala "S2" e tutti gli impianti che saranno installati nella scala "S2" sono dello stesso tipo ("fotovoltaico").
- *tecnicoMax()* restituisce "Bianchi" perché tale tecnico produrrà il massimo incasso (450).
- *scalaSimile("S1", 10, 20)* restituisce "S2" perché, tra la data del 10 e la data del 20 (incluse), nella scala "S2" è previsto un numero di installazioni (2) uguale a quello previsto, nello stesso periodo, nella scala "S1".

Esercizio 2

Si arricchisca la classe *ListaConcatenataInt* sviluppata durante il corso con un metodo *int contaCoppie(int k)* che restituisce il numero di coppie di elementi consecutivi tali che il primo è minore di k ed il secondo è maggiore o uguale a k . Ad esempio, se la lista contiene i valori [8, 5, 5, 3, 6, 2, 7, -4, 4, 5] e $k = 5$, allora il metodo restituisce 3 perché la lista contiene 3 coppie che soddisfano le condizioni (3 e 6, 2 e 7, 4 e 5). Il metodo *contaCoppie* dovrà essere ricorsivo o invocare un opportuno metodo ricorsivo sulla classe *NodoInt*.