

Esercitazione 1 – Classi, Aggregazione, Ereditarietà, Classi Astratte

Si definiscano le classi Java che modellano attributi (classi *Attribute*, *ContinuousAttribute*, *DiscreteAttribute*) e insieme di transazioni da elaborare per la scoperta di pattern emergenti (classe *Data*)

Lo studente deve decidere i modificatori di visibilità

- Definire la classe *astratta Attribute* che modella un generico attributo discreto o continuo.

Membri Attributi

String *name*; // nome simbolico dell'attributo

int index; // identificativo numerico dell'attributo (indica la posizione della colonna (0,1,2,...) che rappresenta l'attributo nella tabella di dati)

Membri Metodi

Attribute(String *name*, *int index*)

Input: valori per nome simbolico dell'attributo e identificativo numerico dell'attributo

Output: //

Comportamento: inizializza i valori dei membri *name*, *index*

String getName()

Input:

Output : nome dell'attributo

Comportamento: restituisce il valore nel membro *name*;

int getIndex()

Input:

Output : identificativo numerico dell'attributo

*Comportamento: restituisce il valore nel membro **index**;*

public String toString()

Input:

Output : identificativo

*Comportamento: restituisce il valore del membro **name**;*

- Definire la classe **ContinuousAttribute** che estende la classe **Attribute** e modella un attributo continuo (numerico) rappresentandone il dominio [min,max]

Membri Attributi

float **max**;

float **min** ;// rappresentano gli estremi di un intervallo

Membri Metodi

ContinuousAttribute(String name, **int** index, float min, float max)

Input: nome dell'attributo, identificativo numerico dell'attributo e min e max dell'intervallo dominio

Output : //

Comportamento: Invoca il costruttore della classe madre e avvalora i membri

float getMin()

Input: //

Output : estremo inferiore dell'intervallo

*Comportamento: Restituisce il valore del membro **min***

float getMax()

Input: //

Output : estremo superiore dell'intervallo

*Comportamento: Restituisce il valore del membro **max***

- Definire la classe **DiscreteAttribute** che estende la classe **Attribute** e modella un attributo discreto rappresentando l'insieme di valori distinti del relativo dominio.

Membri Attributi

*String **values**[]; // array di stringhe, una per ciascun valore discreto , che rappresenta il dominio dell'attributo*

Membri Metodi

*DiscreteAttribute(String name, **int** index, String values[])*

Input: valori per nome simbolico dell'attributo, identificativo numerico dell'attributo e valori discreti che ne costituiscono il dominio

Output : //

*Comportamento: Invoca il costruttore della classe madre e avvalora l'array **values**[] con i valori discreti in input.*

int getNumberOfDistinctValues()

Input: //

Output : numero di valori discreti dell'attributo

*Comportamento: Restituisce la cardinalità del membro **values***

String getValue(int index)

Input: // : indice di tipo intero

Output : restituisce un valore nel dominio dell'attributo

*Comportamento: Restituisce il valore in posizione i del membro **values***

- Definire la classe **Data** (parzialmente fornita dal docente) per modellare un insieme di transazioni (vettori attributo-valore)

Membri Attributi

Object **data** [][]; // una matrice di Object che ha numero di righe pari al numero di transazioni da memorizzare e numero di colonne pari al numero di attributi in ciascuna transazione

int **numberOfExamples**; // cardinalità dell'insieme di transazioni

Attribute **attributeSet**[]; // un array di attributi, che sono avvalorati in ciascuna transazione

Membri Metodi

Data()

Input:

Output :

Comportamento:

(1) Popola la matrice **data** [][] con le transazioni (14 transazioni per 5 attributi, come nella tabella riportata di seguito);

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

(2) Avvalora l'array **attributeSet** [] con cinque oggetti di **DiscreteAttribute**, uno per ciascun attributo.

int getNumberOfExamples()

Input: //

Output: cardinalità dell'insieme di transazioni

Comportamento: restituisce il valore del membro [numberOfExamples](#)

int getNumberOfAttributes()

Input: //

Output: cardinalità dell'insieme degli attributi

Comportamento: restituisce la cardinalità del membro [attributeSet](#)

Object getAttributeValue(**int** exampleIndex, **int** attributeIndex)

Input: indice di riga per la matrice [data](#) che corrisponde ad una specifica transazione, indice di colonna per un attributo

Output: valore assunto dall'attributo identificato da [attributeIndex](#) nella transazione identificata da [exampleIndex](#) nel membro [data](#).

Comportamento: restituisce il valore dell' attributo [attributeIndex](#) per la transazione [exampleIndex](#) memorizzata in [data](#)

Attribute getAttribute(**int** attributeIndex)

Input: indice di colonna per un attributo

Output: attributo con indice [attributeIndex](#).

Comportamento: restituisce l'attributo in posizione [attributeIndex](#) di [attributeSet](#)

public String toString()

Input: //

Comportamento: per ogni transazione memorizzata in [data](#), concatena i valori assunti dagli attributi nella transazione separati da virgole in una stringa. Le stringhe che rappresentano ogni transazione sono poi concatenate in un'unica stringa da restituire in output.

- Si utilizzi il metodo *main* in *Data* per verificare il corretto funzionamento delle classi definite.

Esempio di output:

1:sunny,hot,high,weak,no,
2:sunny,hot,high,strong,no,
3:overcast,hot,high,weak,yes,
4:rain,mild,high,weak,yes,
5:rain,cool,normal,weak,yes,
6:rain,cool,normal,strong,no,
7:overcast,cool,normal,strong,yes,
8:sunny,mild,high,weak,no,
9:sunny,cool,normal,weak,yes,
10:rain,mild,normal,weak,yes,
11:sunny,mild,normal,strong,yes,
12:overcast,mild,high,strong,yes,
13:overcast,hot,normal,weak,yes,
14:rain,mild,high,strong,no,