Esercitazione 5 – RTTI, Serializzazione

- *Definire la classe concreta **ContinuousItem** che estende la classe **Item** e modella una coppia <Attributo continuo - valore numerico> (e.g., Temperature=30.5)*

  Metodi

  ContinuousItem(Attribute attribute, Double value)

Output:

Comportamento: richiama il costruttore della super classe

***double** distance(Object a)*

Comportamento: Determina la distanza (in valore assoluto) tra il valore scalato memorizzato nello item corrente (this.getValue()) e quello scalato associato al parametro a. Per ottenere valori scalati fare uso di getScaledValue(...)

- *Modificare la classe **Data** come segue:*

Metodi

public Data()

- *Comportamento: popolare l'insieme di transazioni (Object [][]data) iniziali considerando **Temperature** come attributo continuo e non più come discreto. Valori di esempio sono qui riportati:*

  data[0][1]=new Double (30.3);

  data[1][1]=new Double (30.3);

  data[2][1]=new Double (30);

  data[3][1]=new Double (13);

```
data[4][1]=new Double (0);
data[5][1]=new Double (0);
data[6][1]=new Double (0.1);
data[7][1]=new Double (13);
data[8][1]=new Double (0.1);
data[9][1]=new Double (12);
data[10][1]=new Double (12.5);
data[11][1]=new Double (12.5);
data[12][1]=new Double(29.21);
data[13][1]=new Double (12.5);
```

Modificare la definizione dell'attributo Temperature in explanatorySet come segue:

```
explanatorySet.add(new ContinuousAttribute("Temperature",1, 3.2, 38.7));
```

*Modificare Tuple getItemSet(**int** index)*

*Comportamento: Crea e un istanza di Tuple che modelli la transazione con indice di riga **index** in data. Restituisce il riferimento a tale istanza. Usare lo RTTI per distinguere tra **ContinuousAttribute** e **DiscreteAttribute** (e quindi creare nella tupla un **ContinuousItem** o un **DiscreteItem**)*

- *Modificare la classe QTMiner con l'aggiunta dei metodi per la serializzazione e de-serializzazione di C.*

  Aggiungere il costruttore:

public QTMiner(String fileName) throws FileNotFoundException, IOException, ClassNotFoundException

Input: percorso+ nome file

*Comportamento: Apre il file identificato da fileName, legge l'oggetto ivi memorizzato e lo assegna a **C**.*

public void salva(String fileName) throws FileNotFoundException, IOException

Input: percorso+ nome file

Comportamento: Apre il file identificato da ***fileName*** *e salva l'oggetto riferito da C in tale file.*

- *Implementare dove serve l'interfaccia* ***Serializable***
- *Testare usando la classe MainTest (fornita dal docente). Un esempio di output è riportato in Output.txt*