RUBRICA: A MAKING OF.

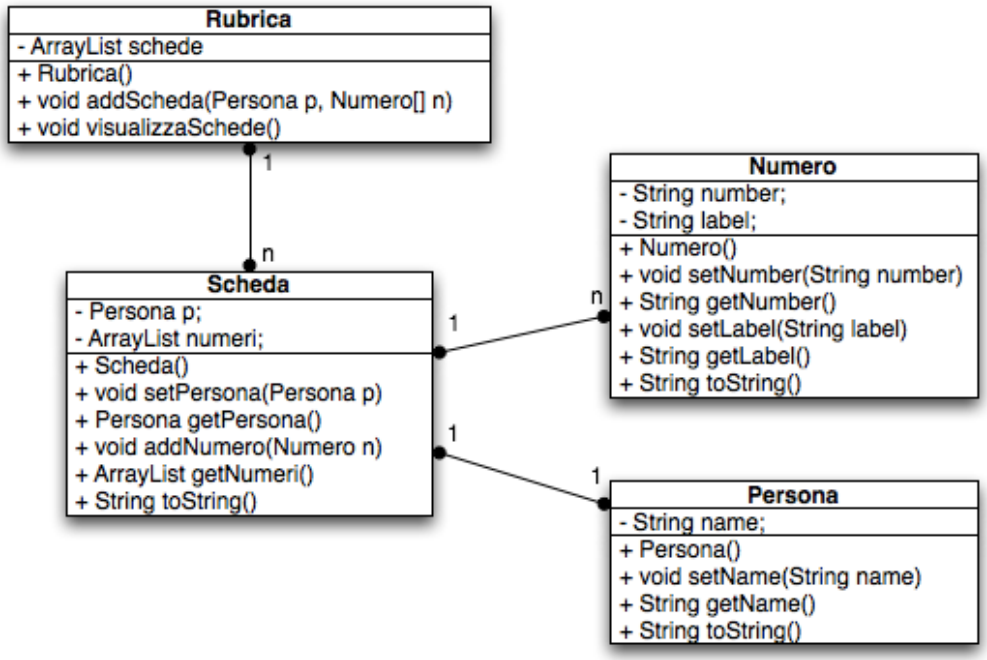
Un primo esempio di modello ad oggetti

Finora abbiamo modellato classi semplici (Persona, Cerchio), e non abbiamo fornito molta interazione tra di esse: ci siamo limitati ad utilizzare i loro metodi in maniera isolata, e a stampare a terminale le loro proprietà. È ora il momento di pensare ad una struttura reale, e al modo di implementarla in qualche linguaggio informatico (Java, nel nostro caso).

Per quanto poco stimolante, tra i primi esempi didatticamente validi salta in mente quello di una **rubrica telefonica**, cioè di un elenco di schede che contengono informazioni su una singola persona; le entità in gioco sono quindi le seguenti:

- una rubrica;
- varie schede;
- varie persone (sola una per scheda);
- vari numeri (uno o più per singola persona),

Tali informazioni si rappresentano ordinatamente nel seguente diagramma:



Le classi sono legate fra esse da vincoli di quantità.

A differenza dei diagrammi di flusso logico tradizionali, che pongono la propria enfasi sul tempo verticale, il diagramma delle classi va interpretato sul piano, rompendo il nesso *prima-dopo*: le entità definite si relazionano durante tutto il ciclo di vita del programma, e per ognuna di esse sono definiti i **campi** ed i **metodi**.

In dettaglio:

- campi e metodi segnati con il **segno meno** (-) sono **privati**;
- campi e metodi segnati con il **segno più** (+) sono **pubblici**;
- i numeri a margine delle linee di connessioni indicano la **cardinalità** del vincolo: ad esempio, ad una rubrica (1) possono corrispondere più schede(n), così come ad ogni scheda (1) è associata una sola persona (1); di conseguenza si implementa il modello delle classi, usanto *vettori* o *ArrayList* se necessario.

L'implementazione concreta

All'interno dell'archivio sono presenti questo medesimo file, e varie classi in file separati:

- **`Rubrica.java;`**
- **`Scheda.java;`**
- **`Persona.java;`**
- **`Numero.java;`**

Per eseguire il programma, digitate come al solito:

```
javac Rubrica.java
java Rubrica
```

e vedete cosa accade su terminale:

```
La rubrica contiene 2 voci.

Mario
Casa: 666666

Gino
Casa: 666666
Mobile: 333333
```

Molto bene, il programma funziona e stampa il contenuto della rubrica. Il codice è abbondantemente commentato ed il diagramma precedente è sufficientemente esplicativo, quindi avete mezzi a sufficienza per capire bene la realizzazione del programma; seguono però consigli ed osservazioni circa questioni meno salienti, ma che potrebbero esservi utili in sede di realizzazione ex novo del modello.

- Il programma è separato in file differenti [uno per classe] per comodità di lettura, ma è come se fosse un file unico. Quando scrivete **`new NomeClasse()`**, Java cerca all'interno di tutti i file della cartella corrente una dichiarazione di tale classe, ed in caso di ricerca fortunata provvede all'esecuzione di tale costruttore;
- Se abbiamo interesse a costruire oggetti le cui classi *non* sono all'interno della cartella, dobbiamo importarli esplicitamente, ad esempio:

  ```
  import java.util.ArrayList;
  ```

  Vedremo presto il significato di tale scrittura, per ora prendetelo per buono;
- La classe Rubrica presenta due metodi **`addScheda()`**, con parametri di ingresso diversi: a seconda dell'esigenza chiamate l'uno o l'altro, sarà Java a capire quale dei due deve utilizzare, grazie al binding dinamico;
- La classe Scheda presenta un metodo **`addNum()`** per aggiungere numeri all'istanza corrente: tale approccio è stato preferito al prevedibile **`setNum()`** per evidenti motivi - i numeri vengono inseritiuno alla volta! Non fa male però aggiungere anche il metodo precedente, ergo fatelo;
- Il campo **`number`** della classe Numero è di tipo **`String`**: questo è controintuitivo, però è l'unico modo per risolvere il problema degli zeri iniziali;
- La classe Persona ha il solo campo **`name`** - dotatela almeno di quello **`lastName`** ed **`email`**!

Avete materiale a sufficienza per realizzare la vostra rubrica da zero, presto implementeremo funzioni di ordinamento e di ricerca; per ora vi auguro buon lavoro e vi invito a chiedere lumi sia in aula che per email.

D.